

**POLITECHNIKA CZĘSTOCHOWSKA  
WYDZIAŁ INŻYNIERII MECHANICZNEJ I INFORMATYKI**



**PRACA DOKTORSKA**

# **Algorytmy identyfikacji sensorów obrazujących**

**Jarosław Bernacki**

Promotor: prof. dr hab. inż. Rafał Scherer

**Częstochowa, 2023**



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>6</b>
1.1	Aktualny stan badań . . . . .	10
1.1.1	Identyfikacja indywidualnego źródła – ISCI . . . . .	10
1.1.2	Identyfikacja modelu – SCMI . . . . .	13
1.2	Motywacja pracy . . . . .	15
1.3	Układ pracy . . . . .	16
1.4	Publikacje . . . . .	16
1.5	Konwencje . . . . .	18
<b>2</b>	<b>Zbiór zdjęć IMAGINE</b>	<b>19</b>
2.1	Istniejące zbiory danych . . . . .	20
2.2	Zbiór zdjęć IMAGINE . . . . .	20
2.3	Witryna internetowa zbioru IMAGINE oraz pobieranie zdjęć . . . . .	24
2.4	Podsumowanie . . . . .	26
<b>3</b>	<b>Algorytmy identyfikacji sensorów obrazujących</b>	<b>27</b>
3.1	Podstawowe definicje . . . . .	27
3.2	Istniejące algorytmy identyfikacji sensorów obrazujących . . . . .	29
3.3	Proponowane algorytmy . . . . .	33
3.3.1	Algorytm MSE-DSI . . . . .	33
3.3.2	Algorytm CompaRe . . . . .	37
3.3.3	Algorytm Vignetting-CT . . . . .	40
3.3.4	Algorytm Distortion-CT . . . . .	44
3.3.5	Identyfikacja z wykorzystaniem konwolucyjnych sieci neuronowych . . . . .	49
3.4	Podsumowanie . . . . .	51

<b>4</b>	<b>Weryfikacja eksperymentalna</b>	<b>53</b>
4.1	Zbiory zdjęć . . . . .	53
4.2	Klasyfikacja . . . . .	55
4.2.1	Miary oceny . . . . .	55
4.2.2	Algorytmy z literatury . . . . .	56
4.2.3	Algorytm Lukás – procedura klasyfikacji . . . . .	56
4.3	Wyniki klasyfikacji . . . . .	57
4.3.1	Algorytm MSE-DSI . . . . .	57
4.3.2	Algorytm CompaRe . . . . .	60
4.3.3	Algorytm Vignetting-CT . . . . .	64
4.3.4	Algorytm Distortion-CT . . . . .	67
4.4	Czas przetwarzania obrazów . . . . .	69
4.4.1	Algorytm MSE-DSI . . . . .	69
4.4.2	Algorytm CompaRe . . . . .	72
4.4.3	Algorytm Vignetting-CT . . . . .	79
4.4.4	Algorytm Distortion-CT . . . . .	80
4.4.5	Identyfikacja w oparciu o konwolucyjne sieci neuronowe . . . . .	81
4.5	Podsumowanie . . . . .	83
<b>5</b>	<b>Analiza statystyczna uzyskanych wyników</b>	<b>85</b>
5.1	Wstęp teoretyczny do analizy statystycznej . . . . .	85
5.1.1	Statystyki opisowe . . . . .	85
5.1.2	Testowanie hipotez . . . . .	86
5.1.3	Opis wykorzystanych testów . . . . .	88
5.2	Statystyki opisowe . . . . .	94
5.2.1	Algorytm MSE-DSI . . . . .	94
5.2.2	Algorytm CompaRe . . . . .	95
5.2.3	Algorytm Vignetting-CT . . . . .	97
5.3	Weryfikacja hipotez statystycznych . . . . .	98
5.3.1	Zbiór zdjęć IMAGINE . . . . .	98
5.3.2	Algorytm MSE-DSI . . . . .	108
5.3.3	Algorytm CompaRe . . . . .	114
5.3.4	Algorytm Vignetting-CT . . . . .	121
5.4	Podsumowanie . . . . .	125

<b>6</b>	<b>Odporność algorytmów identyfikacji</b>	<b>127</b>
6.1	Operacje na pikselach – techniki zaszumiania zdjęć . . . . .	128
6.1.1	Losowe punkty na zdjęciach (szum „pieprz i sól”) . . . . .	128
6.1.2	Rozmywanie zdjęć metodą Gaussa . . . . .	129
6.1.3	Szum Poissona . . . . .	131
6.1.4	Usuwanie najmniej znaczącego bitu . . . . .	133
6.1.5	Numeryczna analiza deformacji . . . . .	135
6.1.6	Podsumowanie . . . . .	141
6.2	Algorytm przeciwdziałania identyfikacji . . . . .	141
6.2.1	Metoda przeciwdziałania identyfikacji w rozumieniu algorytmu Lukás . . . . .	141
6.2.2	Wyniki eksperymentów . . . . .	142
6.3	Podsumowanie . . . . .	145
<b>7</b>	<b>Podsumowanie</b>	<b>146</b>
<b>A</b>	<b>Algorytm Lukás – wartości współczynnika korelacji <math>\rho</math></b>	<b>149</b>
<b>B</b>	<b>Wybrane macierze konfuzji</b>	<b>152</b>
B.1	Algorytm MSE-DSI . . . . .	152
B.2	Algorytm CompaRe . . . . .	156
B.3	Algorytm Vignetting-CT . . . . .	163
B.4	Odporność algorytmów identyfikacji – wyniki dla zbioru IMAGINE . . . . .	165
B.4.1	Szum „pieprz i sól” . . . . .	165
B.4.2	Rozmycie Gaussa . . . . .	166
B.4.3	Szum Poissona . . . . .	166
B.4.4	Usuwanie najmniej znaczącego bitu (LSB) . . . . .	167
<b>C</b>	<b>Algorytm Lanczosa próbkowania obrazu</b>	<b>168</b>
	<b>Spis rysunków</b>	<b>170</b>
	<b>Spis tabel</b>	<b>173</b>
	<b>Spis algorytmów</b>	<b>179</b>
	<b>Bibliografia</b>	<b>180</b>

# Rozdział 1

## Wstęp

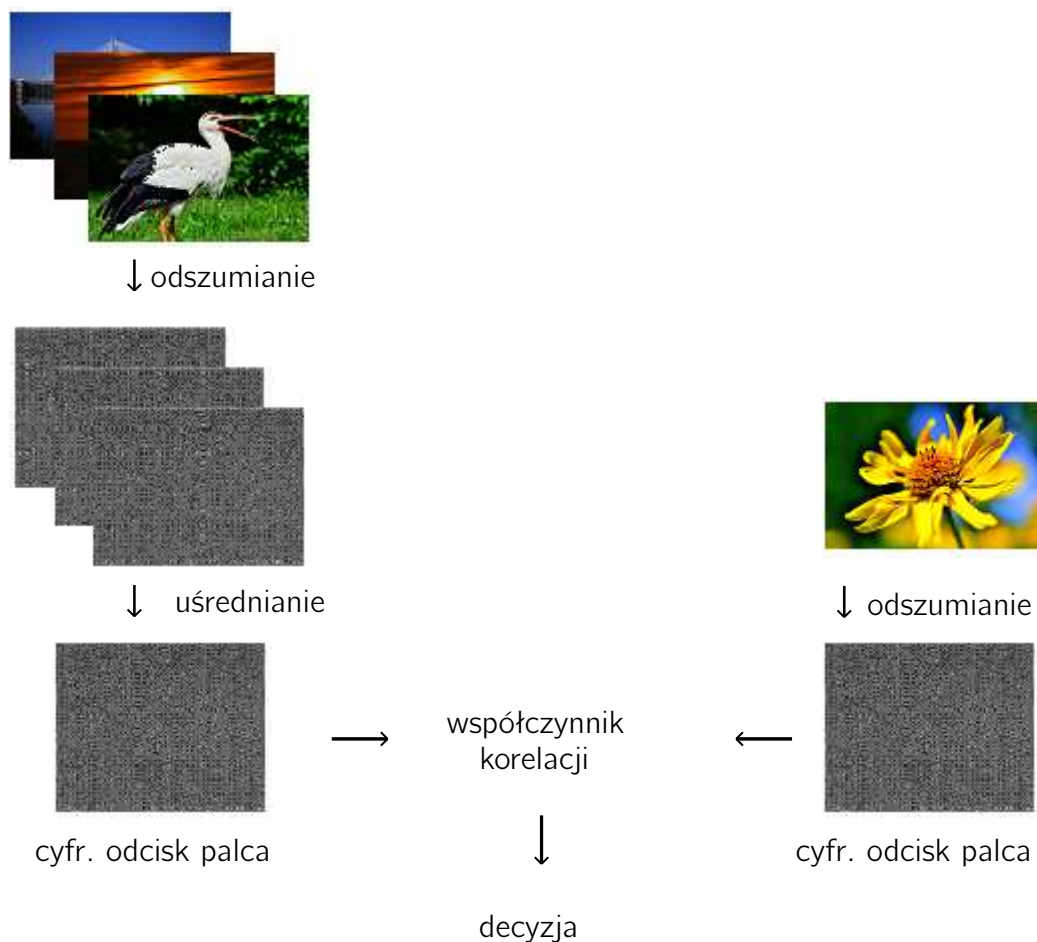
Kryminalistyka cyfrowa (ang. *digital forensics*) jest dziedziną, która zajmuje się badaniem materiałów pochodzących z urządzeń cyfrowych. Materiały te mogą być wykorzystane jako materiał dowodowy np. w postępowaniu sądowym. Przykładowymi urządzeniami, z których takie materiały mogą pochodzić, są: komputer (serwer), telefon komórkowy (smartfon, tablet), urządzenia posiadające sensory obrazujące (aparat cyfrowy, skaner, drukarka, itd). Sensorem obrazującym jest urządzenie elektroniczne zawierające materiały światłoczułe zdolne do rejestracji obrazu i zapisu go w formie cyfrowej. Przykładem urządzenia wyposażonego w sensor obrazujący jest aparat cyfrowy, skaner płaski bądź urządzenie mobilne wyposażone w wbudowany aparat cyfrowy (np. смартфон czy tablet).

Identyfikacja aparatów cyfrowych na podstawie zdjęć (ang. *digital camera identification*) jest popularnym kierunkiem badań od wielu lat. Problem ten jest rozpatrywany w dwóch aspektach: identyfikacja indywidualnego źródła (ang. *individual source camera identification – ISCI*) oraz identyfikacja modelu (ang. *source camera model identification – SCMI*). Identyfikacja w aspekcie ISCI rozróżnia wszystkie rozpatrywane aparaty osobno – przykładowo, dla  $n$  egzemplarzy modelu Nikon D750 oraz  $m$  egzemplarzy modelu Sony A7R III, indeksujemy każdy z egzemplarzy: Nikon D750 (0), Nikon D750 (1), ..., Nikon D750 ( $n$ ); Sony A7R III (0), Sony A7R (1), ... Sony A7R III ( $m$ ). Identyfikacja modelu (SCMI) sprowadza się do rozróżniania określonego modelu aparatu cyfrowego od innych modeli, ale nie odróżnia określonego aparatu od innych egzemplarzy tego samego typu [67, 68]. Stąd dla  $n$  aparatów Nikon D750 i  $m$  aparatów Sony A7R III, metodologia SCMI wyróżni jedynie ogólny model danego aparatu (D750 lub A7R III). W literaturze największym zainteresowaniem cieszą się algorytmy i metody rozróżniające aparaty cyfrowe w oparciu o metodologię ISCI. Jednym z najistotniejszych algorytmów jest metoda opublikowana przez Lukás w 2006 roku, która wykorzystuje tzw. cyfrowy odcisk palca (ang. *camera's fingerprint*) [104]. W pracy tej pokazano, że cyfrowy odcisk palca, który można ob-

liczyć na podstawie zdjęć generowanych przez aparat, może być wykorzystany do identyfikacji aparatów cyfrowych. Cyfrowy odcisk palca  $\mathbf{N}$  można obliczyć w następujący sposób:

$$\mathbf{N} = \mathbf{I} - F(\mathbf{I}) \quad (1.1)$$

gdzie  $\mathbf{I}$  jest zdjęciem wejściowym, a  $F$  to filtr odszumiający. Schemat działania algorytmu przedstawiony jest na rys. 1.1.



Rysunek 1.1: Schemat odszumiania obrazów i klasyfikacja nowego obrazu w oparciu o algorytm Lukás [104]

Podejście zaproponowane w pracy [104] jest bardzo efektywne w kontekście identyfikacji urządzeń w aspekcie ISCI, a dokładność klasyfikacji w większości przypadków sięga nawet 99% [58, 66, 82, 104]. Algorytm wymaga odszumiania wszystkich kanałów kolorystycznych

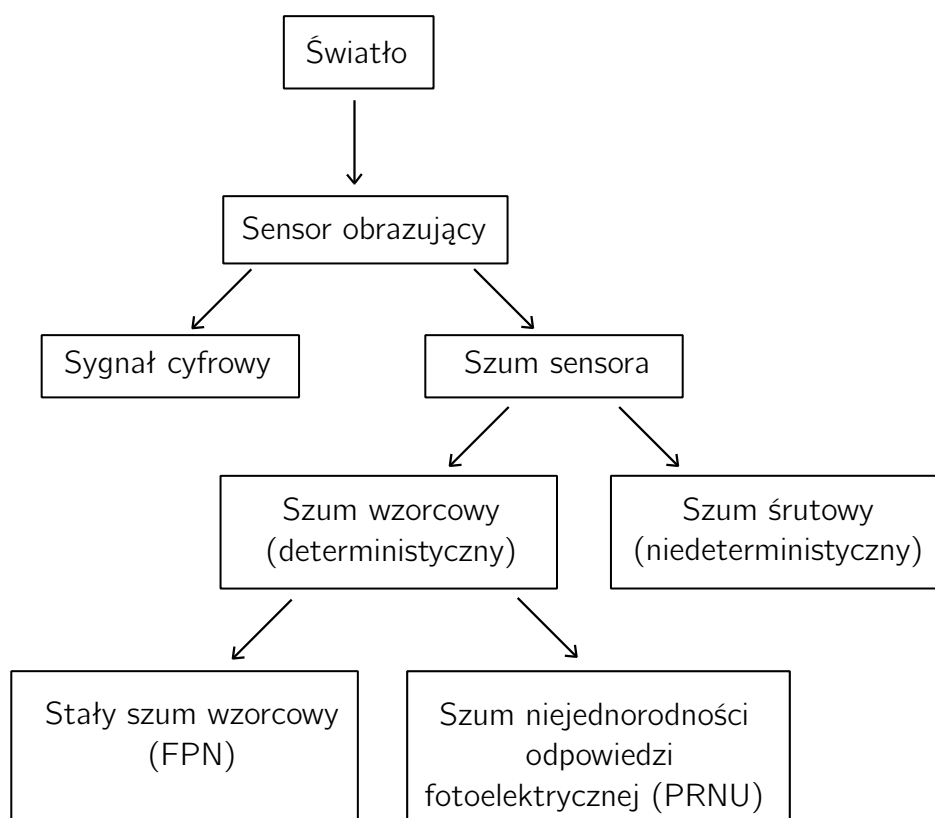
zdjęć wejściowych w ich pełnej rozdzielczości. Jednak z uwagi na duże rozmiary zdjęć współczesnych aparatów bądź smartfonów (np.  $6016 \times 4016$ ,  $6960 \times 4640$ ,  $8000 \times 6000$  czy  $8256 \times 5504$  pikseli), obliczenie cyfrowych odcisków palca może okazać się bardzo czasochłonne. Na przykład, jeśli obraz wejściowy  $I$  ma rozmiar 24 milionów pikseli ( $6000 \times 4000$  pikseli), odpowiadający mu cyfrowy odcisk palca  $I'$  ma dokładnie te same wymiary. Warto wspomnieć, że duża liczba nowoczesnych algorytmów identyfikacji [26, 59, 60, 83, 145] oparta jest na formule odsumowania zdjęć zaproponowanej właśnie w algorytmie Lukás. Stąd motywacja, aby opracować nowe algorytmy, które będą oferować porównywalną skuteczność identyfikacji w krótszym czasie.

Istnieje wiele źródeł niedoskonałości i szumów, które wchodzą w różne etapy procesu tworzenia obrazu (ang. *image acquisition process*). Nawet jeśli sensor obrazujący wykona zdjęcie absolutnie równomiernie oświetlonej sceny, uzyskany obraz cyfrowy będzie wykazywał niewielkie zmiany intensywności między poszczególnymi pikselami. Wynika to częściowo z szumu fotonicznego (zwanego także szumem śrutowym) [75, 78] (który jest składnikiem losowym), a częściowo z powodu szumu wzorcowego – deterministycznego składnika, który pozostaje w przybliżeniu taki sam dla wielu zdjęć tej samej sceny. Dzięki tej właściwości szum wzorcowy jest obecny w każdym wykonanym przez sensor obrazujący zdjęciu, dzięki czemu można go wykorzystać do identyfikacji aparatu cyfrowego.

Głównymi składnikami szumu wzorcowego są stały szum wzorcowy (ang. *fixed pattern noise* – FPN) oraz szum niejednorodności odpowiedzi fotoelektrycznej (ang. *photo-response nonuniformity noise* – PRNU). FPN jest powodowany przez ciemne prądy (ang. *dark currents*). Odnosi się to przede wszystkim do różnic między pikselami, gdy sensor obrazujący nie jest wystawiony na działanie światła. Ponieważ FPN jest szumem addytywnym, niektóre średniej i wysokiej klasy cyfrowe aparaty konsumenckie automatycznie tłumią ten szum, odejmując ciemną ramkę [40] od każdego zrobionego zdjęcia. FPN zależny jest również od ekspozycji i temperatury. W obrazach naturalnych dominującą częścią szumu wzorcowego jest PRNU. Szum PRNU jest spowodowany niejednorodnością pikseli, która jest definiowana jako różna wrażliwość pikseli na światło. Ta wrażliwość spowodowana jest niejednorodnością płytek krzemowych i niedoskonałościami podczas procesu produkcji sensora obrazującego [15, 104]. Przykładowo, kilka sensorów obrazujących, których zadaniem jest sfotografowanie tej samej, identycznie oświetlonej sceny, może w finalnym zdjęciu wygenerować różne wartości intensywności pikseli, np. 214, 212, 215. W takim przypadku oko ludzkie nie zdoła rozpoznać tak nieznaczących różnic pomiędzy wartościami pikseli, jednak algorytm identyfikacji, który dokładnie analizuje wartości pikseli, będzie w stanie rozróżnić takie obrazy, a co za tym idzie, również sensory. Identyfikacja sensorów na podstawie PRNU cieszy się dużym zainteresowaniem w literaturze na przestrzeni ostatnich lat [65, 73, 104, 114, 115, 140]. Klasyfikację szumów występujących w sensorach obrazujących



można zobaczyć na rys. 1.2.



Rysunek 1.2: Klasyfikacja szumów występujących w sensorach obrazujących. Szum generowany przez sensor składa się z deterministycznego szumu wzorcowego (ang. *pattern noise*) oraz niedeterministycznego szumu śrutowego (ang. *shot noise*). Szum wzorcowy natomiast tworzą stały szum wzorcowy (ang. *fixed pattern noise* – FPN) oraz szum niejednorodności odpowiedzi fotoelektrycznej (ang. *photo-response nonuniformity noise* – PRNU) [15]

Niniejsza rozprawa porusza temat identyfikacji sensorów obrazujących w oparciu o analizę materiałów z nich pochodzących, czyli obrazów (zdjęć). Rozpatrywanymi urządzeniami będą aparaty cyfrowe (lustrzanki, bezlusterkowce, aparaty kompaktowe, a także urządzenia mobilne, takie jak smartfony lub tablety) oraz skanery płaskie. Analiza dotyczy przede wszystkim identyfikacji w aspekcie ISCI. Identyfikacja będzie przeprowadzana na podstawie naturalnych zdjęć przedstawiających rzeczywiste sceny (tj. pochodzących „wprost” z aparatu cyfrowego, a więc niepoddawanych zewnętrznym modyfikacjom w oprogramowaniu służącym do obróbki zdjęć, typu Adobe Lightroom/Photoshop itd.).

## 1.1 Aktualny stan badań

Jak wspomniano we Wprowadzeniu, identyfikację sensorów obrazujących można przeprowadzać w oparciu o aspekt identyfikacji indywidualnego źródła (ang. *individual source camera identification* – ISCI) lub identyfikacji modelu (ang. *source camera model identification* – SCMI). W niniejszym podrozdziale przedstawiono aktualny stan badań w obu aspektach.

### 1.1.1 Identyfikacja indywidualnego źródła – ISCI

**Metody oparte na PRNU** Jak wspomniano w poprzednim podrozdziale, identyfikacja sensorów obrazujących na podstawie szumu PRNU cieszy się dużą popularnością w literaturze [26, 44, 55, 59, 67, 79, 82, 111, 112]. Algorytmy wykorzystujące takie podejście generalnie zapewniają wysoką dokładność identyfikacji aparatów cyfrowych, która zwykle nie jest niższa niż 90% w przypadku rozpoznawania konkretnego egzemplarza aparatu cyfrowego. Wadą opisywanego podejścia jest czas obliczania PRNU, które może okazać się czasochłonne. W pracy [23] pokazano, że typowy czas potrzebny do obliczenia PRNU waha się od 2 minut (dla małego obrazu o rozdzielczości  $3200 \times 2400$ , czyli około 8 milionów pikseli) do nawet 5 minut dla obrazu o rozmiarze 24 milionów pikseli ( $6000 \times 4000$  pikseli) w przypadku typowego komputera osobistego. Dlatego fakt ten może wykluczyć algorytm z zastosowania na skalę masową. W pracy [35] pokazano, że takie podejście sprawdza się także w przypadku plików wideo pochodzących z kamer cyfrowych.

W pracy [64] przedstawiono metodę analizy korelacji krzyżowej i stosunku energii szczytowej do korelacji (PCE) w celu identyfikacji aparatu. Wzorzec PRNU jest obliczany tak, jak pokazano w równaniu 1.1, a detektor korelacji ze stosunkiem PCE jest używany do pomiaru podobieństw między szumami resztowymi. Podobnie jak w pracy [104], algorytm zapewnia wysoką dokładność identyfikacji. Jednak wydajność czasowa algorytmu nie jest badana, niemniej jednak można się spodziewać, że podejście to będzie czasochłonne.

W pracy [82] opisano metodę, która poprawia wyniki uzyskane w pracy [104]. Szum wzorca sensora jest reprezentowany jako biały szum wpływający na obrazy. Zaproponowano użycie korelacji z normą korelacji kołowej jako statystyki testowej. Identyfikacja poszczególnych egzemplarzy aparatów wynosi nie mniej niż 95%. Jednak podobnie jak w przypadku wcześniej wspomnianych algorytmów, wydajność czasowa nie jest zadowalająca dla przetwarzania obrazu na skalę masową ze względu na długi czas przetwarzania pojedynczego obrazu.

W pracy [14] omówiono wstępne badanie użycia algorytmu  $k$ -średnich do zarządzania wzorcami PRNU. Wzorce są porównywane ze sobą za pomocą korelacji w celu klasteryzacji za pomocą algorytmu  $k$ -średnich. Podobne wzorce zgrupowane w klastrze są uważane za należące

do tego samego aparatu. Podobne podejście zostało także zaprezentowane w pracy [142].

W pracy [79] wykorzystano wyniki z publikacji [104] do analizy, czy obrazy udostępnione w sieciach społecznościowych na różnych kontach zostały wykonane przez tego samego użytkownika. Wyniki eksperymentów potwierdziły przydatność algorytmu Lukás w tak zdefiniowanym problemie.

W pracy [148] omówiono wpływ losowych projekcji (ang. *random projections*) w celu kompresji odcisków palców aparatów cyfrowych. Ideą losowych projekcji jest zmniejszenie rozmiaru obrazu wejściowego  $\mathbf{I}$ . Autorzy zbadali możliwość powiązania zdjęć z aparatem w przypadku skompresowanych odcisków palca aparatów. Wyniki wykazały, że taka operacja nie zmniejsza dokładności klasyfikacji. Szczegółowa idea losowych projekcji jest opisana w pracy [6]. Praca [99] także przedstawia ideę kompaktowej reprezentacji odcisku palca. Proponowana metoda wykorzystuje wartości własne macierzy (ang. *eigenvalues*) do obliczenia kompaktowej formy PRNU na podstawie obrazów wejściowych  $\mathbf{I}$ . Eksperymenty wykazały wysoką skuteczność klasyfikacji aparatów w oparciu o proponowaną metodę. Problematyczne może jednak okazać się obliczanie wartości własnych macierzy, które jest wymagające obliczeniowo.

W pracy [59] zastosowano analizę PRNU dla metody umożliwiającej uwierzytelnienie za pomocą biometrii i urządzenia mobilnego. Zaproponowano protokół uwierzytelnienia, który łączy rozpoznawanie sensora aparatu wraz z rozpoznawaniem tęczówki oka. Takie rozwiązanie może znaleźć zastosowanie na przykład w systemie logowania. Działanie protokołu sprowadza się do następujących czynności: użytkownik wykonuje zdjęcie swojego oka za pomocą aparatu w swoim smartfonie, a system sprawdza zarówno skan tęczówki (uprzednio posiadanej w bazie danych), jak i PRNU z wykorzystanego aparatu. W protokole, PRNU jest obliczane w taki sam sposób, jak w pracy [104] (równanie 1.1). Takie rozwiązanie powinno zwiększyć bezpieczeństwo systemu uwierzytelniania.

Odcisk palca aparatu został uwzględniony również w innych aspektach. W pracy [30, 68, 135, 158] przebadano odporność PRNU pod kątem „oszukania” klasyfikatora. Okazuje się, że możliwe jest podrobienie PRNU określonego aparatu cyfrowego w celu wygenerowania fałszywego obrazu „udającego”, że został wygenerowany przez inny aparat. W pracy [68] wprowadzono metodę *PRNU fingerprint-copy*. Celem było zaciemnienie PRNU konkretnego aparatu poprzez wstawienie do niego PRNU innego aparatu. Można to zrealizować wykonując proste operacje algebraiczne. Załóżmy, że  $\hat{\mathbf{K}}_N$  jest odciskiem palca aparatu obliczonym z  $N$  obrazów, a  $\mathbf{J}$  jest odciskiem palca innego aparatu, który chcemy „umieścić” w  $\hat{\mathbf{K}}_N$ . Obliczamy

$$\mathbf{J}' = \mathbf{J}(1 + \alpha\hat{\mathbf{K}}_N) \quad (1.2)$$

gdzie  $\alpha > 0$  jest współczynnikiem odpowiadającym za intensywność odcisku palca. Eksperymentalna ocena wykazała, że taka wymiana odcisków palców z aparatu była skuteczna, tzn.

mając  $\hat{K}_N$  można było wytworzyć fałszywy obraz  $J$  „udający” taki aparat. Nadmienić jednak należy, że taka strategia może nie być praktyczna, ponieważ wymagany jest reprezentatywny zestaw zdjęć z aparatu, na który chcemy „wymienić” odcisk palca.

**Uczenie głębokie** Niezwykle popularnym kierunkiem badań w ostatnich latach jest wykorzystanie uczenia głębokiego (ang. *deep learning*), które obejmuje konwolucyjne sieci neuronowe (ang. *convolutional neural networks* – CNN). Sieci konwolucyjne znajdują szeroki zakres zastosowań w różnych problemach badawczych, obejmujących między innymi analizę tekstów, detekcja anomalii, rozpoznawanie mowy czy przetwarzanie obrazów. CNN z powodzeniem można także wykorzystać w problemie identyfikacji sensorów obrazujących, czego dowodzą liczne publikacje w literaturze [26, 34, 37, 95, 97, 110, 126, 144, 145, 150, 154, 155, 156]. Słabą jednak stroną modeli opartych na uczeniu głębokim są duże wymagania sprzętowe, konieczne do przeprowadzenia procesu uczenia sieci, które w zdecydowanej większości przypadków przekraczają możliwości typowego komputera klasy PC.

Podjęcie uczenia głębokiego do identyfikacji sensorów obrazujących zostało zbadane w pracy [54]. Warstwy spłotowe składały się z zestawu filtrów górnoprzepustowych, które zostały zastosowane do obrazu wejściowego  $\mathbf{I}$  w celu wyodrębnienia i poznania charakterystycznych jego cech. Proponowana metoda nie została jednak gruntownie przebadana, gdyż do eksperymentów klasyfikacyjnych wykorzystano jedynie trzy urządzenia mobilne: Apple iPhone 5, Samsung Galaxy S4 oraz Samsung Galaxy Tab II.

Wykorzystanie konwolucyjnych sieci neuronowych zostało przedstawione w pracy [26]. Zaproponowano uczenie sieci składającej się z czterech spłotowych warstw bezpośrednio ze zdjęć JPG.

W pracy [113] zaproponowano platformę podobną do [59], w której wykorzystano uczenie głębokie do identyfikacji tęczówki oraz aparatu cyfrowego. Zaproponowane rozwiązanie może być szczególnie przydatne w zadaniu uwierzytelnienia użytkownika, np. przy logowaniu do systemu za pomocą urządzeń mobilnych. Temat ten był również badany w pracach [44, 81, 112, 147].

Konwolucyjna sieć neuronowa z w pełni połączonymi warstwami (ang. *fully connected*) została przedstawiona w pracy [145]. Warstwy konwolucyjne, które obejmują spłot (ang. *convolution*), aktywację (ang. *activation*) i łączenie (ang. *pooling*) są odpowiedzialne za reprezentację cech. Do klasyfikacji wykorzystano w pełni połączone warstwy. Wzór szumu PRNU został wyodrębniony z obrazów ze znanej formuły  $\mathbf{N} = \mathbf{I} - F(\mathbf{I})$  [68, 82, 104], gdzie  $\mathbf{I}$  oznacza obraz wejściowy, a  $F$  jest filtrem odszumiającym. Uczenie sieci przeprowadzono z wykorzystaniem karty graficznej nVidia GeForce Titan X (12 GB). Czas na naukę sieci wyniósł 5,5 godziny, jedynie dla 12 egzemplarzy aparatów.

W pracy [110] zbadano odporność modeli uczenia głębokiego. Pokazano, jak „oszukać” klasyfikator oparty na CNN, aby identyfikacja aparatu była utrudniona lub niemożliwa. Zbadano kilka ataków, w których obraz przechodzi bezstratną lub stratną kompresję. Zaatakowanie klasyfikatora przeprowadzono następującymi metodami: metodą szybkiego znaku gradientu (FGSM) [70], DeepFool [117] oraz za pomocą procedury JSMA [121]. Idea FGSM jest dość prosta i polega na atakowaniu zdjęć wejściowych poprzez dodanie do nich dodatkowego szumu. Metoda może być skuteczna, jednak w niektórych przypadkach wizualna jakość zdjęcia może zostać wyraźnie pogorszona, dlatego wprowadzono metodę DeepFool, która minimalizuje ten problem. DeepFool opiera się na lokalnej linearyzacji atakowanego klasyfikatora. JSMA to zachłanna procedura iteracyjna, która oblicza mapę istotności, a następnie oznacza piksele, które najbardziej przyczyniają się do prawidłowej klasyfikacji obrazu. Eksperymenty wykazały, że zastosowanie wymienionych ataków może w niektórych przypadkach skutecznie „oszukać” klasyfikator oparty na CNN.

W pracy [55] omówiono w pełni połączoną sieć. Mapy cech zostały obliczone jako splot obrazu **I** i proponowanego jądra (filtra) **K**. Sieć wykorzystywała dwie warstwy splotowe (ang. *convolutional layers*), warstwę *max-pooling* i trzy w pełni połączone warstwy. Przeprowadzono dwa eksperymenty na zestawie danych MICHE [42], który obejmuje trzy urządzenia mobilne: Apple iPhone 5S, Samsung Galaxy S5 i Samsung Galaxy Tab 2. Celem pierwszego eksperymentu było sklasyfikowanie każdego egzemplarza na podstawie aparatu z tyłu obudowy. W drugim eksperymencie sklasyfikowano pięć sensorów: oba aparaty (przód/tył) z Apple iPhone 5S i Samsung Galaxy S4 oraz aparat w Samsung Galaxy Tab 2. Eksperymenty potwierdziły wysoką skuteczność klasyfikacji rozważanych urządzeń. Uczenie sieci przeprowadzono na karcie graficznej nVidia Tesla K80 (24GB). Nie zbadano jednak czasu potrzebnego na nauczenie sieci.

**Pozostałe metody** Mniej popularne metody identyfikacji sensorów obrazujących w aspekcie ISCI obejmują: analizę zdjęć HDR [77], identyfikację na podstawie kurzu [47, 48], analizę zimnych i gorących pikseli (ang. *hot and cold pixels*) [31, 61, 80, 96, 101], czy wykorzystanie rozkładu Poissona [138].

### 1.1.2 Identyfikacja modelu – SCMI

W przypadku podejścia opartego na identyfikacji modelu (SCMI), najczęściej analizuje się wady optyczne aparatu cyfrowego, które mają wpływ na końcową jakość generowanych obrazów. Najczęściej wykorzystywanymi wadami optycznymi są winietowanie obiektywu [89], dystorsja obiektywu [66] czy aberracja chromatyczna [93]. Winietowanie można również rozumieć jako zmniejszenie jasności obrazu na jego krawędziach w stosunku do centrum obrazu [89]. Kolejną

wadą optyczną jest dystorsja obiektywu. Wada ta powoduje, że powiększenie obrazu zmienia się wraz z odległością od osi optycznej. Wyróżniamy trzy rodzaje dystorsji obiektywu: beczkowa (ang. *barrel*), poduszkowa (ang. *pincushion*) oraz falista (ang. *mustache*). W dystorsji beczkowej powiększenie obrazu zmniejsza się wraz z odległością od osi optycznej. W dystorsji poduszkowej powiększenie obrazu wzrasta wraz z odległością od osi optycznej. Wreszcie zniekształcenie faliste jest połączeniem zniekształcenia beczkowego i poduszkowego [139].

Próbie identyfikacji sensorów obrazujących na podstawie parametrów dystorsji obiektywu podjęto w pracy [38]. Ponadto, wykorzystano także wartości intensywności pikseli. Także w pracy [66] rozważono metodę identyfikacji aparatów cyfrowych na podstawie parametrów dystorsji obiektywu. Istnieje wiele modeli, którymi można modelować dystorsję, np. model Browna [43], niemniej autorzy zaproponowali własny model:

$$r' = r(1 + a_2r^2 + a_4r^4) \quad (1.3)$$

gdzie  $r'$  jest środkiem obrazu. Parametry  $a_2$  i  $a_4$  określają rodzaj oraz wielkość zniekształcenia i są uważane za unikalne dla każdego aparatu. Wyniki eksperymentów wskazały jednak losową skuteczność identyfikacji aparatów na podstawie parametrów  $a_2$  oraz  $a_4$ , a w niektórych przypadkach poprawność klasyfikacji wynosiła zaledwie 40%.

W pracy [45] przeanalizowano identyfikację aparatów cyfrowych w oparciu o algorytm automatycznego balansu bieli (ang. *Auto White Balance* – AWB). Autorzy próbowali przybliżyć parametry algorytmów AWB i na ich podstawie zidentyfikować sensor obrazujący. Wyniki wskazały, że proponowane podejście może być bardzo skuteczne w rozważanym problemie.

Kolejnym z obszarów badawczych jest analiza histogramu. Histogram to wykres, na którym pokazany jest procentowy rozkład pikseli w każdym kolorze. Umożliwia on wykrycie defektów pikseli, takich jak gorące/martwe piksele [31, 32, 86, 93, 101]. Problem gorących/martwych pikseli był szczególnie charakterystyczny dla urządzeń z sensorem typu *charge coupled device* (CCD). Obecnie większość aparatów/urządzeń mobilnych wyposażonych jest w sensor typu *Complementary Metal-Oxide-Semiconductor* (CMOS), który generuje obrazy z mniejszą liczbą artefaktów pikseli.

Praca [8] przedstawia statystyczny algorytm identyfikacji tęczy za pomocą sensorów obrazujących. Wybrane cechy obrazu zebrano za pomocą pomiaru statystycznego bloku obrazu (ang. *Block Image Statistical Measure* – BISM), entropii falek wysokiego rzędu (ang. *High Order Wavelet Entropy* – HOWE), pomiaru tekstury (ang. *Texture Measure* – TM), jednopoziomowej wieloorientacyjnej tekstury falkowej (ang. *Single-level Multi-orientation Wavelet Texture* – SIMoWT) i pomiarów jakości obrazu. Obrazy podzielono na bloki, w których zastosowano poszczególne metody. W pracy zaproponowane zostały również dwa zbiory danych zawierające 6000 obrazów tęczy. Analiza miar jakości obrazu była również rozważana w pracy [143].

W pracy [33] przedstawiono metodę identyfikacji aparatów cyfrowych za pomocą demozaikowania. Błędy demozaikowania obliczono jako różnicę między obrazem wejściowym a jego ponowną interpolacją przy użyciu kilku algorytmów demozaikowania. Taka operacja generuje informacje o parametrach algorytmu demozaikowania zastosowanym w aparacie. Parametry te można uznać za unikalne i wykorzystać do identyfikacji określonego aparatu. Eksperymenty potwierdziły wysoką skuteczność klasyfikacji aparatów cyfrowych.

W pracy [9] klasyfikacja sensorów została oparta na pomiarach entropii i jakości obrazu. Satta [130] zaproponował mapę niezawodności jako parametr do szacowania szumu wartości piksela. Wyczerpujący przegląd defektów sensorów, które można wykorzystać do identyfikacji aparatu, opisano w pracach [25, 56]. Przedstawione metody dotyczą głównie modeli algebraicznych do określania pochodzenia obrazu, omówiono również metodę na znalezienie gorących/martwych pikseli w oparciu o filtry nieliniowe (np. filtr medianowy).

## 1.2 Motywacja pracy

Analiza literatury pozwoliła zauważyć, że główny nacisk kładziony jest na skuteczną identyfikację sensorów obrazujących z jak najwyższym prawdopodobieństwem. Niewielka jest natomiast liczba prac, których uwaga (poza skutecznością klasyfikacji) jest również skoncentrowana na szybkości działania proponowanych algorytmów. Jest to więc motywacja do prowadzenia badań w takim właśnie kierunku. Wobec powyższego, **celem rozprawy** jest opracowanie algorytmów identyfikacji sensorów obrazujących w aspekcie ISCI, które działają szybciej niż algorytmy istniejące w literaturze przy jednoczesnym zachowaniu porównywalnej skuteczności identyfikacji.

**Zakres rozprawy** obejmuje:

- (1) Opracowanie zbioru danych IMAGINE, służącego do testowania algorytmów identyfikacji sensorów obrazujących zawartych w aparatach cyfrowych;
- (2) Opracowanie algorytmu MSE-DSI, algorytmów identyfikacji wykorzystujących analizę wad optycznych (winietowanie, dystorsja obiektywu) oraz algorytmu identyfikacji na podstawie kompaktowej formy cyfrowego odciska palca sensorów obrazujących;
- (3) Zbadanie odporności algorytmu Lukás oraz konwolucyjnych sieci neuronowych na identyfikację aparatów cyfrowych w oparciu o zdjęcia poddane operacjom zaszumienia;
- (4) Weryfikacja eksperymentalna zaproponowanych metod, mająca na celu wykazanie przydatności zaproponowanych algorytmów w problemie identyfikacji sensorów obrazujących.

## 1.3 Układ pracy

Praca składa się z 7 rozdziałów.

W **rozdziale 2** przedstawiono opis zbioru zdjęć IMAGINE przeznaczonego do testowania algorytmów identyfikacji sensorów obrazujących. Scharakteryzowano wykorzystane urządzenia, opisano witrynę internetową, na której zbiór jest zamieszczony, przedstawiono metodę pobierania zdjęć.

W **rozdziale 3** opisano algorytmy identyfikacji sensorów obrazujących: zarówno znane z literatury, jak i proponowane. Przedstawiono autorskie algorytmy identyfikacji sensorów w aspekcie ISCI: MSE-DSI, CompaRe, a także omówiono algorytmy Vignetting-CT oraz Distortion-CT. Przeanalizowano także identyfikację z wykorzystaniem konwolucyjnych sieci neuronowych.

**Rozdział 4** przedstawia wyniki klasyfikacji źródeł przez zaproponowane algorytmy w odniesieniu do algorytmów znanych z literatury. Poruszony jest zarówno temat skuteczności klasyfikacji, jak i porównanie szybkości działania poszczególnych algorytmów. Przedstawione wyniki analizy wykorzystują reprezentatywną liczbę zdjęć pochodzących z nowoczesnych urządzeń.

W **rozdziale 5** dokonano analizy statystycznej uzyskanych wyników, która wykazała użyteczność proponowanych algorytmów.

W **rozdziale 6** przeanalizowano odporność algorytmu Lukás oraz konwolucyjnych sieci neuronowych na identyfikację sensorów obrazujących w oparciu o zdjęcia poddane różnorodnym zewnętrznym szumom; zaproponowano także metodę przeciwdziałania identyfikacji sensorów.

**Rozdział 7** zamyka rozprawę podsumowaniem i wskazaniem kierunków dalszych prac badawczych.

## 1.4 Publikacje

Wyniki dotychczasowych badań zostały opublikowane w dziewięciu pracach. Przegląd literatury został opisany w pracy [18]<sup>1</sup> (**A1**).

**A1** J. Bernacki: A survey on digital camera identification methods, *Forensic Science International: Digital Investigation* (Elsevier), vol. 34, 300983 (2020);

Zbiór zdjęć testowych IMAGINE został opisany w pracy [2] **B1**:

---

<sup>1</sup>Praca została zgłoszona do recenzji do czasopisma *Digital Investigation* (100 punktów na liście MEiN, współczynnik Impact Factor (IF) = 2.192). Po recenzji czasopismo przekształciło się w *Forensic Science International: Digital Investigation*; na liście MEiN czasopismo posiada aktualnie 20 punktów, współczynnik IF = 1.805.



- B1** J. Bernacki, R. Scherer: IMAGINE Dataset: Digital Camera Identification Image Benchmarking Dataset – praca zgłoszona na konferencję International Conference on Computational Science 2023

Algorytm MSE-DSI został opublikowany i przedstawiony na konferencji Computational Collective Intelligence – 13th International Conference [24] (**C1**). Ponadto, w pracy [16] (**C2**) przedstawiono rozszerzoną wersję artykułu **C1**.

- C1** J. Bernacki, R. Scherer: Fast Imaging Sensor Identification, Computational Collective Intelligence – 13th International Conference, Lecture Notes in Computer Science (Springer), vol. 12876, str. 572-584 (2021);

- C2** J. Bernacki, R. Scherer: Digital forensics: a fast algorithm for a digital sensor identification, Journal of Information and Telecommunication (Taylor & Francis), vol. 0, nr 0, str. 1-21 (2022);

Algorytm CompaRe został przedstawiony w pracy [21] (**C3**).

- C3** J. Bernacki: Digital camera identification by fingerprint's compact representation, Multimedia Tools and Applications (Springer), vol. 81, nr 15, str. 21641-21674 (2022);

Algorytmy Vignetting-CT oraz Distortion-CT zostały opublikowane w pracy [17] (**C4**):

- C4** J. Bernacki: Digital camera identification based on analysis of optical defects, Multimedia Tools and Applications (Springer), vol. 79, nr 3-4, str. 2945-2963 (2020).

Analiza identyfikacji sensorów w oparciu o konwolucyjne sieci neuronowe została przedstawiona w artykule [22] (**C5**):

- C5** J. Bernacki, K. Costa, R. Scherer: Individual Source Camera Identification with Convolutional Neural Networks, 14th Asian Conference on Intelligent Information and Database Systems, Recent Challenges in Intelligent Information and Database Systems (Springer), str. 45-55 (2022)

Analiza odporności algorytmów identyfikacji została opisana w dwóch pracach [19] (**D1**) oraz [20] (**D2**):

- D1** J. Bernacki: On robustness of camera identification algorithms, Multimedia Tools and Applications (Springer), vol. 80, nr 1, str. 921-942 (2021);

**D2** J. Bernacki: Robustness of digital camera identification with convolutional neural networks, *Multimedia Tools and Applications* (Springer), vol. 80, nr 19, str. 29657-29673 (2021)

Ponadto, wyniki uzyskane w pracy **C1** są bezpośrednią kontynuacją wcześniejszych badań autora rozprawy przedstawionymi w poniższej pracy (nie wchodzi ona jednak w zakres niniejszej rozprawy):

**C0** J. Bernacki, M. Klonowski, P. Syga: Some Remarks about Tracing Digital Cameras – Faster Method and Usable Countermeasure, *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017)* – vol. 4: SECRIPT, Madryt – 24-26 lipca 2017, str. 343-350

## 1.5 Konwencje

W opisach formalnych czcionką pogrubioną oznaczone są wektory bądź macierze. W niektórych przypadkach, aby uniknąć niejasności, przytoczone są oryginalne (angielskie) nazwy pojęć, których tłumaczenie nie jest jednoznaczne lub mogłoby się okazać kontrowersyjne (dotyczy to w szczególności nazw warstw sieci konwolucyjnych – np. warstwa *max-pooling*, itp.).

Wyniki klasyfikacji algorytmów będziemy prezentować w oparciu o macierze konfuzji. Z uwagi na rozległą weryfikację eksperymentalną, znaczną liczbę macierzy konfuzji zamieszczamy w dodatku B do niniejszej pracy.

Z uwagi, iż cytowane w rozprawie algorytmy z literatury na ogół nie posiadają swoich nazw, będziemy je identyfikować w oparciu o nazwisko pierwszego autora (w przypadku, gdy praca posiada wielu autorów) lub obu autorów (jeśli nie występuje ich więcej). Dla przejrzystości, pomijamy wykorzystanie frazy „et al.”, gdyż ograniczałaby czytelność. Przykładowo, algorytm zaprezentowany w pracy [104] będziemy identyfikować jako „algorytm Lukás” zamiast „algorytm Lukás et al.” itd.

We względu na wymogi projektu „Politechnika Częstochowska uczelnią dostępną” oraz likwidację barier cyfrowych (konkurs POWR.03.05.00-IP.08-00-DOS/20), niniejsza rozprawa została zredagowana z wykorzystaniem czcionki bezszeryfowej.

## Rozdział 2

# Zbiór zdjęć IMAGINE

W niniejszym rozdziale przedstawiony zostanie zbiór zdjęć o nazwie IMAGINE, który może zostać wykorzystany do testowania algorytmów identyfikacji aparatów. Zbiór wykorzystuje szereg nowoczesnych rodzajów urządzeń wyposażonych w sensory obrazujące wykonane w technologii CMOS (ang. *Complementary Metal-Oxide-Semiconductor*), takich jak: aparaty cyfrowe (kompaktowe, bezlusterkowe (ang. *mirrorless interchangeable lens camera* – MILC), lustrzanki cyfrowe (ang. *digital single lens reflex* – DSLR), urządzenia mobilne z wbudowanym aparatem cyfrowym (smartfony, tablety), a także drony. Zbiór wykorzystuje łącznie ponad trzy tysiące zdjęć z 66 egzemplarzy urządzeń. Motywacją do opracowania zbioru jest stosunkowo niewielka liczba dostępnych zbiorów zdjęć w literaturze. Zbiór wykorzystuje zdjęcia, których część pochodzi z internetowego serwisu o nazwie *Optyczne.pl*<sup>1</sup>. Portal ten zajmuje się głównie testowaniem nowoczesnych urządzeń fotograficznych. Część zdjęć pochodzi ze zbiorów prywatnych autora rozprawy<sup>2</sup>.

W pierwszej części przypomnimy istniejące zbiory zdjęć, a następnie szczegółowo opiszemy proponowany zbiór wraz z listą dostępnych urządzeń. W kolejnej części przedstawimy witrynę internetową, na której zbiór jest dostępny, a także opiszemy sposób pobrania zdjęć.

Wyniki prezentowane w niniejszym rozdziale zostały opublikowane w pracy **B1**.

---

<sup>1</sup>Autor rozprawy uzyskał pisemną zgodę redaktora naczelnego serwisu *Optyczne.pl* na wykorzystanie zdjęć do celów badawczych.

<sup>2</sup>[https://www.flickr.com/photos/jarek\\_b/](https://www.flickr.com/photos/jarek_b/)

## 2.1 Istniejące zbiory danych

Jednym z najbardziej znanych i powszechnie używanych jest zbiór Dresden Image Database [63], który zawiera kilkadziesiąt tysięcy zdjęć z kilkuset różnych urządzeń (głównie aparatów kompaktowych). Z racji opublikowania zbioru w 2011 roku, zawiera on dość przestarzałe jak na dzisiejsze standardy urządzenia – większość wykorzystywanych aparatów wyposażona jest w niemal nieużywane dziś sensory CCD (ang. *charge-coupled device*).

Zbiór VISION [134] został zaprezentowany w 2017 roku i wykorzystuje zdjęcia z 35 nowoczesnych smartfonów następujących producentów: Apple (iPad/iPhone), Huawei, Lenovo, LG, Microsoft, OnePlus, Samsung, Sony, Wiko i Xiaomi. Zbiór ten jednak nie obejmuje nowoczesnych aparatów cyfrowych.

Opublikowany w 2018 roku zbiór UNIFI [133] zawiera zdjęcia wykonane w technologii HDR (ang. *High Dynamic Range*). Wykorzystuje urządzenia następujących producentów: Asus, Huawei, iPad/iPhone, OnePlus, Xiaomi i Samsung. Podobnie, jak w przypadku zbioru VISION, słabą stroną jest niewielka liczba wykorzystanych urządzeń.

Zbiór zdjęć MICHE [1, 42] z 2015 roku zawiera zdjęcia tęczówki oka wykonane trzema modelami smartfonów: Apple iPhone 5, Samsung Galaxy S4 oraz Samsung Galaxy Tab 2. Zbiór ten jednak jest przeznaczony do testowania algorytmów rozpoznawania tęczówki, więc jego przydatność z punktu widzenia identyfikacji sensorów może być dyskusyjna.

Zbiór danych Fochrenheim został opublikowany w 2020 roku [74]. Zawiera ponad 23 000 zdjęć wykonanych 27 smartfonami. Jest to zaawansowana baza, w której zdjęcia przedstawiają 143 rzeczywiste sceny. Autorzy zweryfikowali przydatność swojego zbioru, testując z jego użyciem kilka wybranych z literatury metod identyfikacji sensorów obrazujących. Ideowo, zbiór jest podobny do proponowanego zbioru IMAGINE, jednak został opublikowany nieco wcześniej, tj. wtedy, gdy trwały prace nad proponowanym zbiorem. Ograniczeniem zbioru jest ograniczenie się jedynie do urządzeń mobilnych i brak przetestowania nowoczesnych aparatów cyfrowych.

## 2.2 Zbiór zdjęć IMAGINE

Jak wspomniano na początku rozdziału, proponowany zbiór zdjęć IMAGINE wykorzystuje obrazy z nowoczesnych urządzeń wyposażonych w sensory CMOS. Dostępne są zdjęcia z 17 aparatów bezlusterkowych, 5 aparatów kompaktowych (w tym jedna kamera sportowa), 2 dronów, 17 lustrzanek, 12 urządzeń mobilnych (11 smartfonów oraz jeden tablet). Niektóre urządzenia występują w dwóch egzemplarzach w obrębie tego samego modelu, co umożliwi wykorzystanie zbioru do testowania algorytmów identyfikacji w aspekcie ISCI (omówionym we Wprowadzeniu).

Tabela 2.1: Rozmiary sensorów (niektóre sensory oznaczone symbolami FF oraz FE mają wymiary nieznacznie inne niż wymienione w tabeli)

Sensor	Rozmiar [mm]	Przekątna [mm]
FF	36,0 × 24,0	43,27
FX	35,9 × 24,0	43,18
FE	35,6 × 23,8	42,74
APS-C <sup>1</sup>	22,3 × 14,8	26,76
APS-C <sup>2</sup>	23,5 × 15,7	28,26
APS-C <sup>3</sup>	23,6 × 15,8	28,40
1"	13,2 × 8,8	15,86
1/2,55"	6,17 × 4,55	7,67
1/2,3"	6,16 × 4,62	7,70
1/3"	4,80 × 3,60	6,00
1/3,1"	4,40 × 3,30	5,50
1/4,0"	3,20 × 2,40	4,00

Urządzenia wyposażone są w sensory obrazujące o różnych fizycznych rozmiarach. Obejmują one sensory rozmiaru zbliżonego do klatki filmu małoobrazkowego (w języku potocznym zwane „pełną klatką”), sensory o wymiarach około 1,5 razy mniejszych od klatki filmu małoobrazkowego (potocznie znane jako APS-C – ang. *Advanced Photo System type-Classic*), sensory typu 1" oraz małe sensory stosowane w aparatach kompaktowych, dronach, kamerach sportowych bądź urządzeniach mobilnych. Fizyczne rozmiary sensorów małoobrazkowych są następujące: w przypadku aparatów marki Canon (oznaczenie FF) – 36 × 24 mm; w przypadku aparatów marki Nikon (oznaczenie FX) – 35,9 × 24,0 mm; aparaty Sony (oznaczenie FE) – 35,6 × 23,8 mm. Sensory klasy APS-C są zbliżone do rozmiarów 23,0 × 16,0 mm. Ściślej, dla aparatów marki Canon fizyczny rozmiar tego typu sensora wynosi 22,3 × 14,8 mm, w przypadku marki Fujifilm jest to 23,5 × 15,7 mm, natomiast dla marki Nikon – 23,6 × 15,8 mm. Sensory typu 1" mają fizyczne wymiary 13,2 × 8,8 mm. Pozostałe sensory, które są stosowane w aparatach kompaktowych, dronach oraz kamerach sportowych posiadają fizyczny rozmiar około 6,0 × 4,0 mm, natomiast w przypadku urządzeń mobilnych wykorzystywanych w niniejszym zbiorze, w przeważającej liczbie przypadków jest to około 4,0 × 3,0 mm. W tab. 2.1 przedstawiono fizyczne rozmiary typów sensorów wykorzystywanych urządzeń, natomiast tab. 2.2 prezentuje pełną listę urządzeń, które są użyte w zbiorze.

Tabela 2.2: Wykorzystane urządzenia w zbiorze IMAGINE (MILC – aparat bezlusterkowy, DSLR – lustrzanka cyfrowa). Symbol \* oznacza brak oficjalnej informacji od producenta o fizycznym rozmiarze sensora (można podejrzewać, że jest on podobny do rozmiaru 1/4,0"). Rozdzielczość zdjęć zdefiniowana jest w pikselach i oznacza maksymalny rozmiar zdjęcia, jaki można uzyskać z określonego aparatu

Nazwa urządzenia	Typ	Rozdzielczość	Sensor	Egzemplarze
Acer Liquid Jade S	smartfon	4160 × 3120	*	1
Apple iPhone 5S	smartfon	3624 × 2448	1/3"	1
Canon EOS 1D X Mark II	DSLR	5472 × 3648	FF	1
Canon EOS 5D Mark IV	DSLR	6720 × 4480	FF	1
Canon EOS 6D Mark II	DSLR	6240 × 4160	FF	1
Canon EOS 750D	DSLR	6000 × 4000	APS-C <sup>1</sup>	2
Canon EOS 760D	DSLR	6000 × 4000	APS-C <sup>1</sup>	2
Canon EOS M3	MILC	6000 × 4000	APS-C <sup>1</sup>	2
Canon EOS M5	MILC	6000 × 4000	APS-C <sup>1</sup>	2
Canon EOS M50	MILC	6000 × 4000	APS-C <sup>1</sup>	2
Canon EOS 90D	DSLR	6940 × 4640	APS-C <sup>1</sup>	1
Canon EOS M100	MILC	6000 × 4000	APS-C <sup>1</sup>	1
Canon EOS M200	MILC	6000 × 4000	APS-C <sup>1</sup>	1
Canon EOS R	MILC	6720 × 4480	FF	1
Canon EOS R5	MILC	8192 × 5464	FF	1
Canon EOS R6	MILC	5472 × 3648	FF	1
Canon EOS RP	MILC	6240 × 4160	FF	1
Canon PowerShot G9 X Mark II	kompaktowy	5472 × 3648	1"	2
Canon PowerShot SX270 HS	kompaktowy	4000 × 3000	1/2,3	1
DJI Spark	dron	3968 × 2976	1/2,3	1
Fujifilm X-T200	MILC	6000 × 4000	APS-C <sup>2</sup>	1
Lenovo K5 Plus	smartfon	4096 × 2304	*	1
LG K10	smartfon	4160 × 2336	*	1
Microsoft Lumia 640	smartfon	3264 × 1840	1/4,0"	1
Nikon D5	DSLR	5568 × 3712	FX	1
Nikon D6	DSLR	5568 × 3712	FX	1
Nikon D500	DSLR	5568 × 3712	APS-C <sup>3</sup>	1

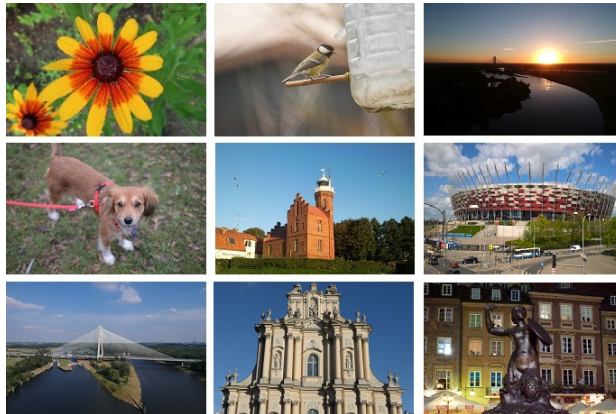
Kontynuacja na następnej stronie

Tabela 2.2 – Kontynuacja z poprzedniej strony

Nikon D610	DSLR	6016 × 4016	FX	1
Nikon D750	DSLR	6016 × 4016	FX	2
Nikon D780	DSLR	6048 × 4024	FX	1
Nikon D810	DSLR	7360 × 4912	FX	1
Nikon D850	DSLR	8256 × 5504	FX	1
Nikon D3100	DSLR	4608 × 3072	APS-C <sup>3</sup>	2
Nikon D5600	DSLR	6000 × 4000	APS-C <sup>3</sup>	2
Nikon D7200	DSLR	6000 × 4000	APS-C <sup>3</sup>	2
Nikon P100	kompaktowy	3648 × 2736	1/2,3"	1
Nikon Z6	MILC	6048 × 4024	FX	2
Nikon Z6 II	MILC	6048 × 4024	FX	1
Nikon Z7	MILC	8256 × 5504	FX	1
Nikon Z7 II	MILC	8256 × 5504	FX	1
Nokia 2.2	smartfon	4160 × 3120	1/3,1"	1
Samsung Galaxy A40	smartfon	4608 × 3456	1/2,8"	1
Samsung Galaxy Ace 3	smartfon	2560 × 1920	*	1
Samsung Galaxy S7	smartfon	4032 × 3024	1/2,55"	2
Samsung Galaxy SIII mini	smartfon	2560 × 1920	*	1
Samsung Galaxy Tab A 10.1	tablet	3264 × 1836	*	1
Samsung Galaxy Trend 2 Lite	smartfon	2048 × 1232	*	1
Samsung Omnia II	smartfon	2560 × 1920	*	1
Sony A1	MILC	8640 × 5760	FE	1
Sony A7R III	MILC	7952 × 5304	FE	1
Sony A7S	MILC	4240 × 2832	FE	1
Sony A9	MILC	6000 × 4000	FE	1
Sony ActionCam AS200V	kompaktowy	3104 × 1744	1/2,3"	1
Sony RX100 VI	kompaktowy	5472 × 3648	1"	1
Yuneec Breeze 4K	dron	4160 × 3120	1/3"	1

Wykorzystane zdjęcia są plikami JPG pochodzącymi bezpośrednio z aparatów i nie są w żaden sposób edytowane w żadnym oprogramowaniu. Wszystkie aparaty były ustawione w domyślnym trybie fotografowania (auto) z domyślnym balansem bieli. Niektóre urządzenia są dostępne jako dwa egzemplarze tego samego modelu – dotyczy to następujących aparatów: Canon EOS 750D, Canon EOS 760D, Canon EOS M3, Canon EOS M5, Canon EOS M50,

Canon PowerShot G9X Mark II, Nikon D750, Nikon D3100, Nikon D5600, Nikon D7200, Nikon Z6 oraz Samsung Galaxy S7. Ponadto, można wskazać, że zdjęcia z urządzeń Samsung Galaxy Tab A 10.1 oraz Samsung Galaxy Trend 2 Lite przedstawiają te same kadry. Umożliwia to wykorzystanie zdjęć do testowania algorytmów analizujących wady optyczne aparatu (np. winietowanie obiektywu czy dystorsja). Przykładowe zdjęcia z proponowanego zbioru można zobaczyć na rys. 2.1.



Rysunek 2.1: Przykładowe zdjęcia pochodzące ze zbioru IMAGINE

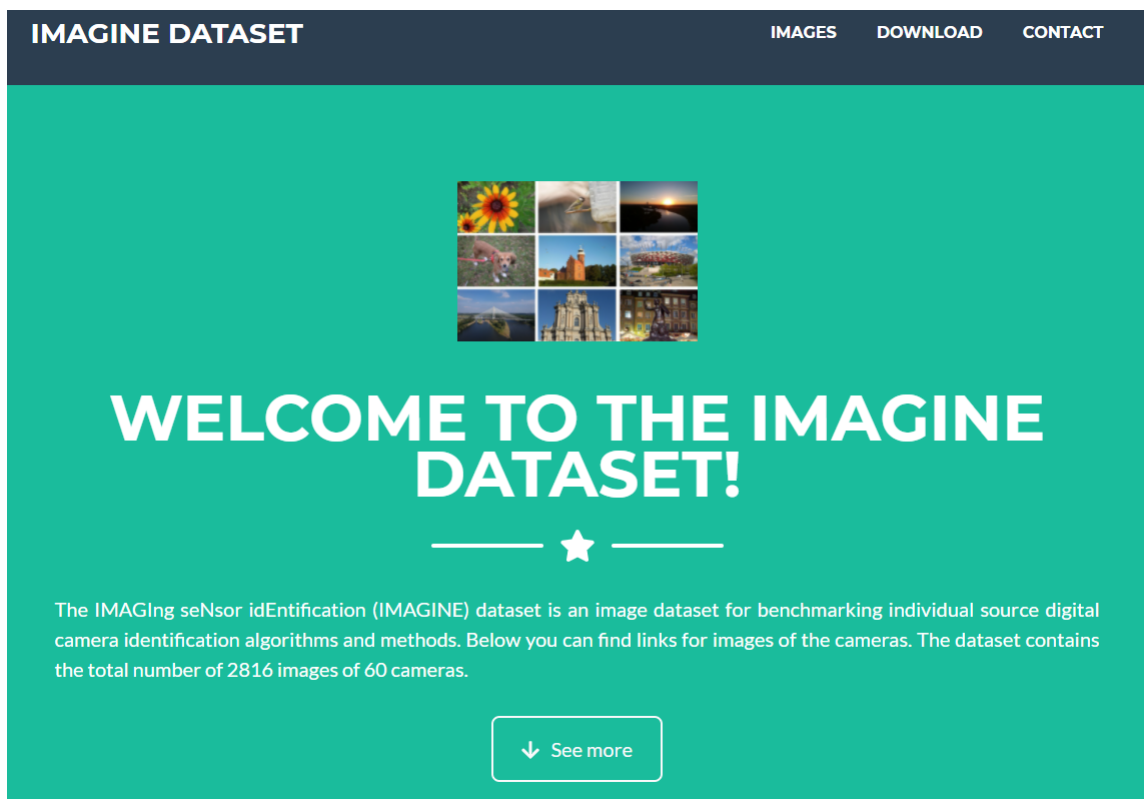
## 2.3 Witryna internetowa zbioru IMAGINE oraz pobieranie zdjęć

Zawartość zbioru IMAGINE jest umieszczona na stronie internetowej, na której zdjęcia można zarówno przeglądać, jak i pobierać. Witryna dostępna jest pod adresem:

<https://kisi.pcz.pl/imagen/>.

Strona internetowa jest wykonana w oparciu o technologie HTML5/CSS3 z uwzględnieniem nowoczesnego standardu responsywności (ang. *responsive web design* – RWD). Dzięki RWD wyświetlanie strony jest wygodne nie tylko na stacjonarnych monitorach o różnych długościach przekątnej ekranu, ale również na urządzeniach mobilnych, która łatwo dostosowuje się do rozmiaru ekranu urządzenia. Na rys. 2.2 przedstawiony jest wygląd strony internetowej omawianego zbioru.





Rysunek 2.2: Strona główna zbioru danych IMAGINE

Zdjęcia wchodzące w skład zbioru IMAGINE można pobrać za pomocą skryptu, który jest dostępny do pobrania na stronie internetowej. Skrypt został napisany w skryptowym języku BASH. Precyzyjna instrukcja obsługi skryptu przedstawiona jest na stronie internetowej. Skrypt jest intuicyjny i łatwy w obsłudze, współpracuje z systemami operacyjnymi zarówno z serii Microsoft Windows, jak i z systemami z rodziny Linux. Na Listingu 2.1 przedstawiono fragment skryptu umożliwiającego pobranie zdjęć zbioru IMAGINE. Domyślnie umożliwia on pobranie wszystkich zdjęć z całego zbioru, jednak użytkownik może w łatwy sposób edytować jego zawartość, by pobrać jedynie interesujące go obrazy.

Listing 2.1: Fragment skryptu BASH umożliwiającego pobieranie zdjęć

```
mkdir Apple_iPhone_5S
for ((n=1;n<=50;n++))
do
  wget -O Apple_iPhone_5S/$n.JPG
  http://www.kisi.pcz.pl/IMAGINE/img/Apple_iPhone_5S/$n.JPG
```

**done**

```
mkdir Canon_EOS_1D_X_Mark_II
for ((n=1;n<=35;n++))
do
  wget -O Canon_EOS_1D_X_Mark_II/$n.jpg
  http://www.kisi.pcz.pl/IMAGINE/img/Canon_EOS_1D_X_Mark_II/$n.jpg
done
```

## 2.4 Podsumowanie

W niniejszym rozdziale zaprezentowano zbiór danych IMAGINE do testowania algorytmów identyfikacji aparatów cyfrowych w aspekcie ISCI. Przedstawiono dotychczas istniejące zbiory i wykazano ich słabe strony, uzasadniając celowość opracowania nowego zbioru. Szczegółowo omówiono proponowany zbiór, a także wykorzystywane urządzenia. Scharakteryzowano witrynę internetową, na której zbiór jest udostępniony, opisano sposób pobrania zdjęć.

## Rozdział 3

# Algorytmy identyfikacji sensorów obrazujących

W niniejszym rozdziale omówione zostaną algorytmy identyfikacji sensorów obrazujących na podstawie obrazów. W pierwszej części przypomnimy wybrane algorytmy z literatury, natomiast w drugiej części przedstawimy algorytmy autorskie. Warto wskazać, że w przypadku algorytmów z literatury przypominamy jedynie ich najważniejsze założenia – szczegółowe parametry wywołania algorytmów i metod (w szczególności konwolucyjnych sieci neuronowych) są przedstawione w publikacjach cytowanych autorów.

Zakładamy wykorzystanie kolorowych zdjęć w formacie RGB, tj. zawierających trzy kanały kolorystyczne, które reprezentowane są jako macierze. Zakładamy także występowanie w zdjęciach artefaktów pikseli (opisanych w podrozdziale 1), dzięki którym identyfikacja sensorów jest możliwa. Wykorzystamy zdjęcia, które nie są modyfikowane z wykorzystaniem zewnętrznego oprogramowania (np. Adobe Lightroom/Photoshop itd.).

### 3.1 Podstawowe definicje

#### Piksel i obraz

**Piksel** (ang. *pixel* – w skrócie *px*) jest najmniejszym elementem obrazu, który jest wypełniony jednolitym kolorem. **Obraz** (zdjęcie) składa się z określonej liczby pikseli. Obraz **I** w odcieniach szarości o wymiarach  $M \times N$  jest macierzą, w której każdy z pikseli ma przypisaną wartość z przedziału  $[0, 255]$ :

$$\mathbf{I} \in \{0, \dots, 255\}^{M \times N} \quad (3.1)$$

Obraz kolorowy  $\mathbf{I}$  zgodny z modelem RGB o wymiarach  $M \times N$  składa się z trzech macierzy, które odpowiadają poszczególnym kanałom kolorystycznym: czerwonemu –  $\mathbf{R}$  (ang. *red*), zielonemu –  $\mathbf{G}$  (ang. *green*) oraz niebieskiemu –  $\mathbf{B}$  (ang. *blue*). Każda z wymienionych macierzy przyjmuje wartości z przedziału  $[0, 255]$ . Zatem pojedynczy piksel jest reprezentowany przez trzy intensywności składowe, z każdego z wymienionych kanałów kolorystycznych. Oznacza to, że pojedynczy piksel może reprezentować jeden z  $256 \cdot 256 \cdot 256 = 16\,777\,216$  kolorów. Formalnie, uzyskujemy:

$$\mathbf{R} \in \{0, \dots, 255\}^{M \times N}, \quad \mathbf{G} \in \{0, \dots, 255\}^{M \times N}, \quad \mathbf{B} \in \{0, \dots, 255\}^{M \times N} \quad (3.2)$$

Liniowe wartości RGB są surowymi danymi uzyskanymi z sensora obrazującego aparatu cyfrowego. Wartości  $\mathbf{R}$ ,  $\mathbf{G}$  i  $\mathbf{B}$  są wprost proporcjonalne do ilości światła padającego na sensor. Wstępne przetwarzanie danych obrazu, takich jak balans bieli, balans kolorów i kompensacja aberracji chromatycznej, odbywa się na liniowych wartościach RGB.

Współczesne aparaty cyfrowe najczęściej jednak zapisują zdjęcia w standardach sRGB [11] bądź Adobe RGB [7]. Wartości sRGB stosują na obrazach RGB nieliniową funkcję, zwaną korekcją gamma. Obrazy zapisane w przestrzeni sRGB są zwykle jaśniejsze, a kolory łatwiejsze do rozróżnienia w porównaniu do obrazów liniowego RGB. Krzywą parametryczną używaną do przekształcenia liniowych wartości kolorów RGB na sRGB definiujemy następująco:

$$\begin{aligned} f(u) &= -f(-u) && \text{dla } u < 0 \\ f(u) &= c \cdot u && \text{dla } 0 \leq u < d \\ f(u) &= a \cdot u^\gamma + b && \text{dla } u \geq d \end{aligned} \quad (3.3)$$

gdzie  $u$  reprezentuje jedną z wartości koloru  $\mathbf{R}$ ,  $\mathbf{G}$  lub  $\mathbf{B}$  z następującymi parametrami:  $a = 1,055$ ,  $b = -0,055$ ,  $c = 12,92$ ,  $d = 0,0031308$  oraz  $\gamma = 1/2,4$ .

W modelu Adobe RGB (zapropozowanym w roku 1998 przez firmę Adobe), wartości liniowego RGB także stosują korekcję gamma, jednakże model jest zdefiniowany za pomocą prostej funkcji potęgowej:

$$\begin{aligned} v &= u^\gamma && \text{dla } u \geq 0 \\ v &= -(-u)^\gamma && \text{dla } u < 0 \end{aligned} \quad (3.4)$$

gdzie  $\gamma = 1/2,19921875$ .

### Identyfikacja w oparciu o cyfrowy odcisk palca PRNU

Jeśli oznaczymy obraz z aparatu cyfrowego jako  $\mathbf{I}$ , a obraz, który zostałby zarejestrowany przy braku jakichkolwiek niedoskonałości jako  $\mathbf{I}_0$ , to  $\mathbf{I}$  można formalnie zdefiniować jako [57, 69]:

$$\mathbf{I} = \mathbf{I}_0 + \mathbf{I}_0\mathbf{K} + \Theta \quad (3.5)$$

gdzie  $\mathbf{K}$  jest współczynnikiem PRNU (cyfrowym odciskiem palca sensora), a  $\Theta$  obejmuje pozostałe składowe szumu, między innymi prądy ciemne (ang. *dark currents*), szum śrutowy (ang. *shot noise*), szum odczytu (ang. *readout noise*) oraz szum kwantyzacji (ang. *quantization noise*) [40, 75, 78]. Cyfrowy odcisk palca PRNU  $\mathbf{K}$  może być oszacowany na podstawie  $P$  obrazów  $\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \dots, \mathbf{I}^{(P)}$  z aparatu cyfrowego:

$$\mathbf{K} = \frac{1}{P} \sum_{i=1}^P \mathbf{I}^{(P)} \quad (3.6)$$

Niech  $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \dots, \mathbf{W}^{(P)}$  będą ich szumami resztkowymi (ang. *noise residuals*) uzyskanymi przy użyciu filtra odsumiającego  $F$ ,  $\mathbf{W}^{(i)} = \mathbf{I}^{(i)} - F(\mathbf{I}^{(i)})$ , gdzie  $i = 1, \dots, N$ . Możemy także zdefiniować estymator największej wiarygodności (ang. *maximum likelihood estimator*) współczynnika PRNU [69]:

$$\hat{\mathbf{K}} = \frac{\sum_{i=1}^N \mathbf{W}^{(i)} \mathbf{I}^{(i)}}{\sum_{i=1}^N (\mathbf{I}^{(i)})^2} \quad (3.7)$$

### Obliczanie współczynnika korelacji między cyfrowymi odciskami palca

Podobieństwo cyfrowych odcisków palca PRNU, wykorzystywane do klasyfikacji można ocenić analizując współczynnik korelacji  $\rho$ . Współczynnik ten określa zależność pomiędzy  $\mathbf{K}$ , a obrazem  $\mathbf{I}^{(x)}$ . Obliczamy go z następującego wzoru 3.8 [66]:

$$\rho(\mathbf{K}, \mathbf{I}^{(x)}) = \frac{(\mathbf{K} - \bar{\mathbf{K}}) \cdot (\mathbf{I}^{(x)} - \bar{\mathbf{I}}^{(x)})}{\|(\mathbf{K} - \bar{\mathbf{K}})\| \|(\mathbf{I}^{(x)} - \bar{\mathbf{I}}^{(x)})\|} \quad (3.8)$$

gdzie  $\|\mathbf{K}\| = \sqrt{\mathbf{K} \cdot \mathbf{K}}$ , natomiast  $\bar{\mathbf{K}}$  i  $\bar{\mathbf{I}}^{(x)}$  oznacza wartość średnią wszystkich elementów w macierzy, odpowiednio  $\mathbf{K}$  oraz  $\mathbf{I}^{(x)}$ . Współczynnik korelacji przyjmuje wartości z przedziału  $[0, 1]$  (dla każdego kanału kolorystycznego z osobna), gdzie 0 oznacza brak korelacji, a 1 maksymalną korelację.

## 3.2 Istniejące algorytmy identyfikacji sensorów obrazujących

W niniejszym podrozdziale opiszemy istniejące algorytmy identyfikacji sensorów obrazujących. Przedstawimy algorytmy Lukás, Bondi, Tuama, które opierają się na analizie cyfrowych odcisków palca obliczonych ze zdjęć w ich pełnym wymiarach. Przypomnimy także algorytmy

Valsesia oraz Li, które do identyfikacji aparatów wykorzystują cyfrowe odciski palca w kompaktowej formie.

### Algorytm Lukás

Jak wskazano w poprzednim podrozdziale, identyfikację sensora aparatu cyfrowego można zrealizować poprzez analizę wzorca PRNU. Czasami PRNU jest również często określany jako szum wzorca sensora (ang. *sensor pattern noise* – SPN) lub szum resztkowy (ang. *noise residual*). Jednym z najpopularniejszych algorytmów identyfikacji sensorów obrazujących w oparciu o tak zdefiniowany cyfrowy odcisk palca aparatu jest algorytm Lukás [104]. Algorytm ten jest zdefiniowany następująco:

---

#### Algorytm 1 Algorytm Lukás [104]

---

**Wejście:** Obraz (zdjęcie)  $\mathbf{I}$  w formacie RGB o wymiarach  $M \times N$  pikseli;

**Wyjście:** Cyfrowy odcisk palca (PRNU)  $\mathbf{N}$  o wymiarach  $M \times N$  pikseli.

1:  $\mathbf{N} = \mathbf{I} - \mathbf{F}(\mathbf{I})$

2: **return**  $\mathbf{N}$

---

W powyższej procedurze  $\mathbf{F}$  jest filtrem odszumiającym. Cyfrowy odcisk palca jest obliczony jedynie na podstawie jednego zdjęcia – w związku z tym powyższą procedurę należy powtórzyć dla większej liczby zdjęć (autorzy proponują wykorzystanie co najmniej  $n = 45$  zdjęć). Ostatecznie, cyfrowy odcisk palca dla określonego aparatu  $\mathbf{N}^{(c)}$  jest obliczony jako średnia cyfrowych odcisków palca z pojedynczych zdjęć:

$$\mathbf{N}^{(c)} = \frac{1}{n} \sum_{i=1}^n \mathbf{I}^{(i)} \quad (3.9)$$

Klasyfikacja nowego zdjęcia  $\mathbf{I}^{(x)}$  odbywa się przez obliczenie jego cyfrowego odcisku palca i porównanie go z obliczonym  $\mathbf{N}^{(c)}$  poprzez obliczenie współczynnika korelacji  $\rho$  między  $\mathbf{N}^{(c)}$  i  $\mathbf{I}^{(x)}$  (wzór 3.8). Współczynnik  $\rho$  i jego pewne własności omówimy szerzej w rozdz. 4. Przykładowe wartości współczynnika  $\rho$  można zobaczyć w dodatku A do niniejszej pracy.

Warto podkreślić, że cyfrowy odcisk palca aparatu rozpatrywany w kategoriach PRNU jest reprezentowany jako macierz o takich samych wymiarach, jak zdjęcie wejściowe  $\mathbf{I}$ .

### Konwolucyjne sieci neuronowe

**Bondi [26]** W pracy [26] przedstawiono konwolucyjną sieć neuronową (CNN) przeznaczoną do identyfikacji sensorów obrazujących na podstawie zdjęć. Struktura sieci przedstawia się

następująco:

1. Pierwsza warstwa konwolucyjna z 32 filtrami o wymiarach  $4 \times 4 \times 3$  z krokiem (ang. *stride*) 1 oraz warstwą *max-pooling* o wymiarach filtra 2 oraz krokiem 2;
2. Druga warstwa konwolucyjna z 48 filtrami o wymiarach  $5 \times 5 \times 32$  z krokiem 1 oraz warstwą *max-pooling* o wymiarach filtra 2 oraz krokiem 2;
3. Trzecia warstwa konwolucyjna z 64 filtrami o wymiarach  $5 \times 5 \times 48$  z krokiem 1 oraz warstwą *max-pooling* o wymiarach filtra 2 oraz krokiem 2;
4. Czwarta warstwa konwolucyjna z 128 filtrami o wymiarach  $5 \times 5 \times 64$  z krokiem 1 generująca jako wyjście wektor 128 elementów;
5. W pełni połączone warstwy (ang. *fully connected*) do końcowej klasyfikacji.

Pozostałe parametry są wyszczególnione w tekście pracy [26]. Sieć jest uczona obrazami JPG bezpośrednio z analizowanych aparatów cyfrowych, bez dodatkowych operacji takich jak odsumianie zdjęć (w przeciwieństwie do algorytmu Lukás).

**Tuama [145]** W pracy Tuama [145] zaproponowano konwolucyjną sieć neuronową z trzema konwolucyjnymi warstwami. Struktura sieci przedstawia się następująco:

1. Pierwsza warstwa konwolucyjna o wymiarach filtra  $3 \times 3$  z krokiem 2 generująca 64 mapy cech (ang. *feature maps*) o wymiarach  $126 \times 126$  pikseli z ReLU jako funkcją aktywacji;
2. Druga warstwa konwolucyjna o wymiarach filtra  $3 \times 3$  z krokiem 2 generująca 64 mapy cech o wymiarach  $63 \times 63$  pikseli z ReLU jako funkcją aktywacji;
3. Trzecia warstwa konwolucyjna o wymiarach filtra  $3 \times 3$  z krokiem 2 generująca 32 mapy cech o wymiarach  $63 \times 63$  pikseli z ReLU jako funkcją aktywacji;
4. Warstwa *max-pooling* o wymiarach filtra  $3 \times 3$  z krokiem 2 generująca 32 mapy cech o wymiarach  $31 \times 31$  pikseli;
5. Trzy warstwy w pełni połączone (ang. *fully connected layers*) z funkcją *soft-max* do końcowej klasyfikacji.

Sieć jest uczona w oparciu o wzorzec PRNU obliczony zgodnie z formułą zaproponowaną przez Lukás, a więc  $\mathbf{I} = \mathbf{F}(\mathbf{I})$ , gdzie  $\mathbf{I}$  jest obrazem wejściowym, a  $\mathbf{F}$  oznacza filtr odsumiający. Tak więc wymiary cyfrowych odcisków palca w rozumieniu metody Tuama mają te same wymiary, jak w przypadku algorytmu Lukás.

**Mandelli [108] i Kirchner & Johnson [85]** Struktury sieci zaproponowanych przez Mandelli [108] oraz Kirchner & Johnson [85] na ogół nie różnią się w sposób istotny od prac Bondi oraz Tuama, dlatego pomijamy prezentację struktur tych sieci. Niemniej wyniki proponowanych w rozprawie algorytmów będziemy również odnosić do tych metod.

### Kompaktowa reprezentacja cyfrowego odcisku palca

**Valsesia [148] i Li [99]** Algorytm Valsesia [148] wykorzystuje ideę losowych projekcji [12, 129, 149] (ang. *random projections*) do reprezentacji cyfrowego odcisku palca aparatu cyfrowego w kompaktowej (tj. „zmniejszonej”) postaci. Cyfrowy odcisk palca reprezentowany jest jako macierz, a losowe projekcje są na niego nakładane w celu skompresowania jego wymiarów. Ogólną ideą losowych projekcji jest rzutowanie oryginalnych  $n$ -wymiarowych danych na  $m$ -wymiarową podprzestrzeń (gdzie  $m < n$ ) przy użyciu macierzy losowej  $\Phi \in \mathbb{R}^{m \times n}$ . Algorytm można opisać następująco:

---

#### Algorytm 2 Algorytm Valsesia [148]

---

**Wejście:** Cyfrowy odcisk palca (PRNU)  $\mathbf{D}$  o wymiarach  $M \times N$  pikseli;

**Wyjście:** Cyfrowy odcisk palca (PRNU)  $\mathbf{A}$  o wymiarach  $m \times n$  pikseli.

- 1:  $\mathbf{A} = \Phi \mathbf{D}$
  - 2: **return**  $\mathbf{A}$
- 

Macierz  $\mathbf{A}$  reprezentuje nowy, skompresowany odcisk palca,  $\mathbf{D}$  to oryginalny odcisk palca aparatu,  $\Phi$  jest macierzą losowych projekcji,  $m < M$  oraz  $n < N$ . Tak więc celem algorytmu jest przemnożenie oryginalnego odcisku palca  $\mathbf{D}$  przez macierz losowych projekcji  $\Phi$  tak, aby w ten sposób uzyskana macierz  $\mathbf{A}$  miała znacznie mniejsze wymiary, niż macierz  $\mathbf{D}$ . Oczywiście, mnożenie macierzy nie zawsze jest możliwe, dlatego macierz losowych projekcji powinna mieć odpowiednio dobrane wymiary, aby możliwe było zrealizowanie algorytmu. Na przykład, jeśli  $\mathbf{D}$  posiada wymiary  $M \times N$ , pomnożenie przez  $\Phi$  macierzy o wymiarach  $N \times P$  wygeneruje w rezultacie macierz  $\mathbf{A}$  o wymiarach  $M \times P$ . Stąd, odpowiedni dobór  $P$  może skutkować wygenerowaniem macierzy  $\mathbf{A}$  o znacznie mniejszych wymiarach w porównaniu do macierzy  $\mathbf{D}$ .

W pracy Li [99] także zaproponowano algorytm kompaktowej reprezentacji odcisku palca aparatu cyfrowego. Idea algorytmu znacznie jednak różni się od podejścia zaproponowanego w pracy Valsesia. Celem jest wygenerowanie wektora cech (ang. *SPN feature extractor*), który przechowuje największe wartości własne macierzy wejściowego odcisku palca aparatu cyfrowego. Na podstawie takiego wektora wyodrębnia się nowy odcisk palca. Pełna definicja algorytmu znajduje się w publikacji autorów [99], a z uwagi na dużą ilość formalnych opisów metody, pomijamy jej prezentację w niniejszej rozprawie.



## 3.3 Proponowane algorytmy

W niniejszym podrozdziale przedstawimy i omówimy autorskie algorytmy identyfikacji sensorów obrazujących. Przedstawione zostaną algorytmy wykorzystywane do identyfikacji sensorów w aspekcie ISCI: MSE-DSI oraz CompaRe. Algorytm MSE-DSI zostanie porównany z algorytmami: Lukás [104], Bondi [26] oraz Tuama [145], natomiast algorytm CompaRe porównamy ponadto z [85, 108] oraz z algorytmami Valsesia [148] oraz Li [99]. Zaproponujemy także pewne usprawnienie pracy konwolucyjnych sieci neuronowych, które pozwoli na przyspieszenie czasu nauki sieci.

Omówimy także algorytmy: Vignetting-CT, który do identyfikacji wykorzystuje wadę winietowania obiektywu, a także algorytm Distortion-CT, który jest metodą identyfikacji opartą na analizie dystorsji obiektywu.

Prezentowane algorytmy zostały opublikowane w pracach **C1**, **C2**, **C3**, **C4** oraz **C5**.

### 3.3.1 Algorytm MSE-DSI

Algorytm identyfikacji sensorów obrazujących Mean Square Error-Digital Sensor Identification (MSE-DSI)<sup>1</sup> ideowo jest zbliżony zasadą działania do algorytmu Lukás. Wymaga on odszumienia zdjęcia wejściowego  $\mathbf{I}$ , na bazie którego odbywa się identyfikacja sensora. Proponowany algorytm wykorzystuje miarę oceny jakości, jaką jest błąd średniokwadratowy (ang. *mean square error* – MSE). Miary tej używa się najczęściej do oceny jakości obrazów czy plików wideo [36, 41, 46, 51, 124]. Oznaczmy przez  $\mathbf{I}$  zdjęcie wejściowe, przez  $\mathbf{I}^{(\mathcal{R})}$  czerwony kanał kolorystyczny zdjęcia  $\mathbf{I}$ , a przez  $F$  filtr odsumiający. Obliczenie wartości MSE algorytmu MSE-DSI jest przedstawione jako algorytm 3.

Zmienne  $M$ ,  $N$  określają wymiary obrazu w pikselach. Możliwe jest wykorzystanie dowolnego filtra odsumiającego (np. medianowego), choć najczęściej w literaturze wykorzystuje się falkowy filtr odsumiający (ang. *wavelet-based denoising filter*) [58, 59, 60, 83, 84, 104]. Szczegóły dotyczące falkowego filtra odsumiającego można poznać w pracy [57].

Algorytm opiera się na obliczeniu wartości MSE na podstawie różnicy intensywności pikseli tylko jednego kanału kolorów obrazu wejściowego  $\mathbf{I}^{(\mathcal{R})}$  i jego odsumionej postaci  $F(\mathbf{I}^{(\mathcal{R})})$ . W przeciwieństwie do algorytmu Lukás, proponujemy przetwarzanie tylko jednego kanału kolorystycznego zamiast wszystkich kanałów, co ma pozytywny wpływ na czas przetwarzania obrazu

---

<sup>1</sup>Algorytm MSE-DSI jest kontynuacją wcześniejszych badań autora rozprawy, które zostały przedstawione w pracy [23] (nie wchodzi ona jednak w zakres niniejszej rozprawy). We wspomnianej pracy został zaproponowany algorytm identyfikacji nazwany PSNR-CT, którego zasada działania jest podobna do omawianego algorytmu MSE-DSI, jednak skuteczność identyfikacji jest zauważalnie niższa.

**Algorytm 3** Algorytm MSE-DSI**Wejście:** Obraz  $\mathbf{I}$  w formacie RGB o wymiarach  $M \times N$  pikseli;**Wyjście:** Wartość MSE (liczba).

---

```

1: Przetwarzaj kanał  $\mathbf{I}^{(R)}$  zdjęcia  $\mathbf{I}$ 
2: for  $i = 0$  to  $m$  do
3:   for  $j = 0$  to  $n$  do
4:     
$$\text{MSE} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^m [\mathbf{I}_{ij}^{(R)} - F(\mathbf{I}_{ij}^{(R)})]^2$$

5:   end for
6: end for
7: return MSE

```

---

(linia 1). Największy koszt wykonania algorytmu reprezentuje operacja obliczenia wartości MSE (linia 4) wewnątrz zagnieżdżonej pętli (linie 2-3). Dlatego proponujemy, aby przyjąć  $m = 512$  oraz  $n = 512$  w celu redukcji liczby operacji wykonywanych przez algorytm [39]. Proponujemy wykorzystanie **dowolnego** wycinka czerwonego kanału kolorystycznego zdjęcia  $\mathbf{I}$  do analizy, co powinno zauważalnie przyspieszyć proces uczenia algorytmu [16]. W porównaniu do algorytmu Lukás, proponowany algorytm wykonuje znacznie mniejszą liczbę operacji dominujących. Jak już wspomnieliśmy, algorytm Lukás operuje na obrazach, których wymiary  $M$  oraz  $N$  są rzędu tysięcy pikseli. Algorytm MSE-DSI wymaga wartości  $m < M$  i  $n < N$ , które są znacznie mniejsze, rzędu setek pikseli. W związku z tym algorytm MSE-DSI w praktyce będzie działał znacznie szybciej. Przypomnijmy, że przykładowe wartości  $M$  i  $N$  dla algorytmu Lukás mogą wynosić, np.  $M = 6000$  oraz  $N = 4000$ . Oczywiście, redukcja danych wykorzystywanych do uczenia algorytmu niesie negatywne skutki w postaci niższej skuteczności końcowej klasyfikacji, jednak jak pokażemy w rozdz. 4, osiągi algorytmu MSE-DSI są akceptowalne. Można więc powiedzieć, że celem algorytmu MSE-DSI jest zapewnienie akceptowalnej skuteczności identyfikacji sensorów przy wyraźnie krótszym czasie działania względem algorytmu Lukás.

Oczywiście, wartość MSE-DSI obliczona za pomocą powyższej procedury reprezentuje pojedyncze zdjęcie z rozważanego aparatu, dlatego chcąc uzyskać reprezentatywny wynik dla danego aparatu, procedurę tę należy powtórzyć dla większej liczby zdjęć (na przykład co najmniej 45 zdjęć, jak zasugerowano w pracy [104]). Następnie proponujemy obliczyć wartość średnią dla rozważanego urządzenia jako średnią arytmetyczną wartości MSE-DSI z określonej liczby zdjęć. Zaobserwowano, że w ten sposób obliczona wartość MSE algorytmu MSE-DSI wydaje się charakterystyczna dla każdego z urządzeń, dlatego rozważamy te wartości w kontekście cyfrowych odcisków palca rozpatrywanych urządzeń. Przykładowe wartości MSE-DSI dla części urządzeń ze zbioru IMAGINE przedstawiono w tab. 3.1.

Analiza wartości MSE-DSI przedstawionych w tab. 3.1 pozwala zauważyć, że wspomniane wartości w znacznej liczbie przypadków wydają się unikalne dla rozważanych urządzeń. Można jednak zauważyć, że dla urządzeń określonego producenta, wartości MSE są zbliżone do siebie, jak widać w przypadku aparatów z rodziny Canon, Nikon czy Sony. Zwraca także uwagę, że wartości „układają” się grupowo dla konkretnych typów urządzeń, na przykład dla aparatów bezlusterkowych Canon, wartości MSE oscylują wokół 10-13. Dla aparatów Nikon, MSE jest na poziomie 18-22, dla aparatów Sony – około 9, maksymalnie 10. Najwyższe wartości obserwujemy dla smartfonów, urządzenia producenta Samsung osiągają wartości MSE na poziomie 28-32. Co istotne, wartości MSE-DSI wykazują również tendencje do unikalności dla kilku egzemplarzy tego samego modelu urządzenia. Przykładowo, dla aparatu Canon EOS 6D Mark II (1) rozpatrywana wartość wynosi 10,11, natomiast dla drugiego egzemplarza tego aparatu jest to 12,78; dla aparatu Canon EOS RP (1) otrzymujemy 11,88, natomiast dla EOS RP (2) jest to 11,54. Być może warto też podkreślić, że zachodzi podejrzenie, iż sensory obrazujące w urządzeniach EOS 6D Mark II oraz EOS RP firmy Canon mogą być dokładnie takie same<sup>2</sup> [4].

Należy jednak podkreślić, że w przypadku wielu egzemplarzy tego samego modelu urządzenia różnice pomiędzy wartościami MSE-DSI nie są już tak wyraźne, jak w przypadkach różnych modeli urządzeń, co może mieć wyraźny wpływ na efektywność identyfikacji. Niemniej analiza eksperymentalna algorytmu pozwala zauważyć, że algorytm można z powodzeniem stosować w aspekcie ISCI.

Klasyfikację można przeprowadzić z wykorzystaniem standardowego algorytmu uczenia maszynowego. Wykorzystując obliczone wartości MSE-DSI dla różnych aparatów, można stworzyć zbiór danych, który można wykorzystać do klasyfikacji „nowego” zdjęcia. Przyporządkowanie analizowanego zdjęcia (z uprzednio obliczoną wartością MSE-DSI) do określonego aparatu można przeprowadzić np. za pomocą algorytmu  $k$  najbliższych sąsiadów (ang. *k-nearest neighbors* – *k-NN*). Oczywiście, można także wykorzystać bardziej zaawansowane algorytmy, np. SVM (ang. *Support Vector Machine*). W pracy [14] zaproponowano użycie algorytmu  $k$ -średnich (ang. *k-means*).

Porównanie z algorytmem Lukás pozwoliło zauważyć, że proponowany algorytm wprawdzie osiąga niższą dokładność klasyfikacji, jednak jest znacznie szybszy w działaniu, dzięki czemu może zostać użyty na przykład do wstępnej selekcji aparatów w ramach przetwarzania wstępnego (ang. *preprocessing*). Należy także podkreślić, że dużą zaletą algorytmu jest identyfikacja jedynie na podstawie pojedynczej wartości liczbowej, a nie w postaci macierzy, jak ma to miej-

---

<sup>2</sup>Informacja ta może być trudna do zweryfikowania na drodze naukowej, jednakże firma Canon w materiałach marketingowych zapewnia, że sensory zawarte w aparatach cyfrowych Canon EOS 6D Mark II oraz Canon EOS RP są różne.

Tabela 3.1: Przykładowe wartości MSE-DSI (po obliczeniu średniej dla 45 urządzeń) dla niektórych urządzeń ze zbioru IMAGINE

Urządzenie	Wartość MSE-DSI
Apple iPhone 5S	26,01
Canon EOS 1D X Mark II	11,16
Canon EOS 6D Mark II (1)	10,11
Canon EOS 6D Mark II (2)	12,78
Canon EOS M200	13,82
Canon EOS R5	10,99
Canon EOS RP (1)	11,88
Canon EOS RP (2)	11,54
Canon PowerShot G9 X Mark II	14,75
Lenovo K5 Plus	17,93
Nikon D6	19,32
Nikon D500	18,96
Nikon D750 (1)	19,95
Nikon D750 (2)	20,20
Nikon D780	18,11
Nikon D850	18,56
Nikon D7200	17,44
Nikon Z7	21,21
Nikon Z7 II	22,87
Samsung Galaxy A40	31,02
Samsung Galaxy S7	28,04
Sony A1	9,98
Sony A7R III	10,01
Sony A9	9,76

sce w przypadku algorytmu Lukás, bądź większości sieci konwolucyjnych. Niesie to korzyści wynikające z braku konieczności przechowywania, plików o być może dużych rozmiarach, zawierających cyfrowe odciski palców. W przypadku rozważanego algorytmu, wartości MSE-DSI można przechowywać w pliku tekstowym w postaci dwuelementowych krotek, na przykład

[urządzenie; wartość MSE – DSI]

To z kolei ułatwia wykonanie kopii zapasowych obliczonych wartości, które ze względu na

niewielki rozmiar pliku, mogą być łatwo przechowywane w przenośnych nośnikach pamięci *flash*.

### 3.3.2 Algorytm CompaRe

Wiele algorytmów identyfikacji, w tym szczególnie często cytowany w niniejszej rozprawie algorytm Lukás, czy też metody oparte na konwolucyjnych sieciach neuronowych (np. Tuama, Bondi), dokonują identyfikacji sensorów w oparciu o cyfrowy odcisk palca mający te same wymiary, co obrazy wejściowe. Może to generować pewne problemy i trudności, na przykład algorytm Lukás, który do końcowej klasyfikacji wykorzystuje współczynnik korelacji, nie będzie mógł przeprowadzić klasyfikacji, gdy macierze reprezentujące cyfrowe odciski palca będą miały różne wymiary. Scenariusz taki jest niezwykle prawdopodobny, gdyż wiele modeli aparatów (także w przypadku tego samego producenta) oferuje różne wymiary zdjęć. Przykładowo, aparat Nikon D7200 generuje zdjęcia o wymiarach  $6000 \times 4000$  pikseli, podczas gdy dla aparatu Nikon D750 jest to  $6016 \times 4016$  pikseli. W takich przypadkach konieczne byłoby kadrowanie zdjęć celem dopasowania ich do określonych wymiarów, co dodatkowo zwiększyłoby czas przetwarzania zdjęć, a ponadto mogłoby mieć potencjalnie negatywny wpływ na efektywność klasyfikacji. Co więcej, obliczanie cyfrowych odcisków palca w oparciu o odsumowanie wszystkich kanałów kolorystycznych zdjęcia może być czasochłonne, szczególnie w kontekście współczesnych urządzeń generujących zdjęcia o dużych wymiarach. Wymienione problemy mogą sprawić, że wykorzystanie niektórych algorytmów identyfikacji może być bardzo utrudnione lub wręcz niemożliwe na masową skalę. Dla przykładu, jeśli obraz wejściowy  $I$  ma wymiary 24,3 milionów pikseli ( $6016 \times 4016$  pikseli), jego cyfrowy odcisk palca  $I'$  w rozumieniu algorytmu Lukás lub sieci konwolucyjnej Bondi ma dokładnie te same wymiary. Dlatego proponujemy algorytm, który wygeneruje cyfrowe odciski palca  $I'$  w tak zwanej zwartej reprezentacji, tj. o znacznie mniejszych wymiarach niż oryginalny obraz wejściowy  $I$  a ponadto, nie wymaga odsumowania obrazów. Przyspieszy to proces uczenia algorytmu identyfikacji, a w konsekwencji klasyfikacji aparatów. Jednocześnie oczekujemy, że dokładność identyfikacji aparatu pozostanie na podobnym poziomie, jak w przypadku algorytmów, które bazują na cyfrowych odciskach palców mających te same wymiary, jak zdjęcia wejściowe. Oczywiście można twierdzić, że odciski palców można obliczyć na oryginalnych obrazach o zmniejszonych wymiarach, jednak może to mieć negatywny wpływ na dokładność klasyfikacji [145], dlatego takiej operacji nie bierzemy pod uwagę.

Rozważmy algorytm CompaRe (Compact Representation), który przedstawiamy jako algorytm 4.

Proponowany algorytm ma na celu redukcję wymiarów cyfrowych odcisków palca w następujący sposób. Obraz wejściowy  $I$  o wymiarach  $M \times N$  powinien zostać podzielony na  $r$  kwadratowych i nienakładających się części, które będziemy nazywać podmacierzami. Następnie

**Algorytm 4** Algorytm CompaRe**Wejście:** Zdjęcie  $\mathbf{I}$  (macierz) o wymiarach  $M \times N$ ;**Wyjście:** Kompaktowa reprezentacja  $\mathbf{I}'$  (macierz) o wymiarach  $M' \times N'$ .

- 1: Weź  $p$  takie, że  $p|M$  oraz  $p|N$ ;
- 2: Podziel zdjęcie  $\mathbf{I}$  na  $r$  kwadratowych i nienakładających się części (podmacierzy)  $k$  o wymiarach  $p \times p$ ;
- 3: Oblicz ślad  $t_s$  każdej podmacierzy  $k_s$ , gdzie  $s \in \{1, \dots, r\}$ ;
- 4: Wygeneruj macierz  $\mathbf{I}'$  (kompaktowa reprezentacja) o wymiarach  $M' = \frac{M}{p}$  oraz  $N' = \frac{N}{p}$  i umieść w jej odpowiednich pozycjach wartości kolejnych śladów podmacierzy  $k_s$ .
- 5: **return**  $\mathbf{I}'$

obliczamy ślad (jest to suma elementów na głównej przekątnej) w każdej z takich podmacierzy i generujemy nową macierz  $\mathbf{I}'$ , w której umieszczamy w odpowiednich miejscach wartości obliczonych śladów. Aby podzielić obraz  $\mathbf{I}$  na  $r$  kwadratowych części, musimy znaleźć wartość  $p$ , która będzie dzielnikiem  $M$  i  $N$ . Gdyby wartość śladu macierzy osiągnęła wartość większą niż 255 (jest to maksymalna wartość intensywności piksela w modelu RGB), proponujemy użyć wartości 255 w celu uniknięcia potencjalnych konfliktów.

Ze względu na różne wymiary obrazów wejściowych, nie jest możliwe podanie konkretnej wartości  $p$ , ponieważ możliwość podzielenia obrazu wejściowego jest uzależniona od wymiarów  $\mathbf{I}$ . Proponujemy dobrać wartość  $p$  w taki sposób, aby uzyskać zwartą reprezentację  $\mathbf{I}'$  o wymiarach zbliżonych do  $640 \times 480$  pikseli lub, jeśli nie jest to możliwe, o podobnych wymiarach, np.  $608 \times 456$ ,  $568 \times 378$  pikseli itd. Takie wymiary dają najlepsze wyniki w identyfikacji aparatów, a także nie wymagają dużej liczby operacji obliczeniowych. W tab. 3.2 podajemy przykładowe rozdzielczości odcisków palców aparatów cyfrowych dla algorytmu Lukás oraz sieci konwolucyjnych oraz rozdzielczość cyfrowych odcisków palców po zastosowaniu procedury CompaRe z przykładowymi wartościami  $p$ . Więcej szczegółów odnośnie wyboru wartości  $p$  opisujemy w rozdz. 4, podrozdz. 4.3.2.

Procedurę CompaRe należy powtórzyć dla wszystkich obrazów z danego aparatu; następnie obliczyć wartość średnią uzyskanych w ten sposób kompaktowych reprezentacji, zgodnie ze wzorem 3.6 (można też skorzystać ze wzoru 3.7).

Jak pokazano w tab. 3.2, cyfrowe odciski palca  $\mathbf{I}'$  aparatów cyfrowych po zastosowaniu procedury CompaRe mają wymiary znacznie mniejsze niż oryginalne wymiary zdjęć wejściowych i odpowiadających im cyfrowych odcisków palca. Przykładowo, dla aparatu Nikon Z7 wymiary zdjęć wejściowych (a więc także odpowiadających im cyfrowym odciskom palca) to  $M = 8256$  oraz  $N = 5504$  (zatem ponad 45 milionów pikseli), podczas gdy po zastosowaniu algorytmu

Tabela 3.2: Przykładowe aparaty wraz z wymiarami obrazów (w pikselach)

Aparat cyfrowy	Wymiary $\mathbf{I}$ (Lukás i CNN)	$\mathbf{I}'$ po zastosowaniu CompaRe	Wartość $p$
Canon 5D Mark IV	$6720 \times 4480$	$560 \times 373$	$p = 12$
Canon RP	$6240 \times 4160$	$624 \times 416$	$p = 10$
Nikon D7200	$6000 \times 4000$	$750 \times 500$	$p = 8$
Nikon Z7	$8256 \times 5504$	$688 \times 458$	$p = 12$
Sony A7R III	$7952 \times 5304$	$568 \times 378$	$p = 14$
Praktica DCZ5	$2560 \times 1920$	$640 \times 480$	$p = 4$
Nikon D200	$3872 \times 2592$	$645 \times 432$	$p = 6$
Samsung NV 15	$3648 \times 2736$	$608 \times 456$	$p = 6$

CompaRe może to być  $M' = 688$  oraz  $N' = 458$ , a więc nieco ponad 315 tysięcy pikseli. Podobną redukcję wymiarów możemy też zaobserwować w przypadku pozostałych urządzeń. Widać więc wyraźnie, że podanie na wejście algorytmów identyfikacji cyfrowych odcisków palca  $\mathbf{I}'$  wygenerowanych proponowaną metodą zamiast oryginalnych obrazów  $\mathbf{I}$ , powinno znacząco przyspieszyć proces czas uczenia takich algorytmów.

Nie spodziewamy się problemu z niemożliwością wybrania  $p$  takiego, które jednocześnie dzieli  $M$  oraz  $N$ , ponieważ nowoczesne urządzenia wytwarzają obrazy o parzystych wartościach  $M$  i  $N$  (stąd wykonanie algorytmu zawsze będzie możliwe dla  $p = 2$ ). Gdyby jednak zdarzyło się, że  $M$  i  $N$  byłyby nieparzyste, proponujemy wykadrować oryginalny obraz wzdłuż obu krawędzi o jeden piksel w celu uzyskania parzystych wartości  $M$  i  $N$ ; w takim przypadku na pewno będzie możliwe wykonanie algorytmu dla  $p = 2$ .

Rozważmy prosty przykład, w którym przedstawimy istotę działania algorytmu. Załóżmy, że mamy obraz wejściowy  $\mathbf{I}$  o wymiarach  $9 \times 6$  pikseli ( $M = 9$  i  $N = 6$ ). Wtedy wymiary cyfrowego odcisku palca w rozumieniu algorytmu Lukás, czy sieci Bondi/Tuama również wynoszą  $9 \times 6$ .

**Przykład 3.1.** Niech  $\mathbf{I}$  będzie zdjęciem wejściowym, załóżmy  $p = 3$ :

$$\mathbf{I} = \begin{bmatrix} 5 & 3 & 1 & 6 & 0 & 9 & 3 & 4 & 9 \\ 4 & 1 & 2 & 3 & 8 & 7 & 8 & 1 & 2 \\ 6 & 3 & 6 & 7 & 4 & 0 & 1 & 3 & 7 \\ 2 & 9 & 6 & 3 & 3 & 1 & 4 & 4 & 7 \\ 1 & 3 & 2 & 1 & 6 & 8 & 0 & 6 & 2 \\ 9 & 9 & 0 & 4 & 2 & 1 & 5 & 3 & 1 \end{bmatrix}$$

Dzielimy zdjęcie  $\mathbf{I}$  na  $M' = \frac{M}{p} = 3$  oraz  $N' = \frac{N}{p} = 2$ , tj.  $r = 3 \cdot 2 = 6$  kwadratowe

i nienakładające się części, jak pokazano poniżej.

$$\mathbf{I} = \left[ \begin{array}{ccc|ccc|ccc} 5 & 3 & 1 & 6 & 0 & 9 & 3 & 4 & 9 \\ 4 & 1 & 2 & 3 & 8 & 7 & 8 & 1 & 2 \\ 6 & 3 & 6 & 7 & 4 & 0 & 1 & 3 & 7 \\ \hline 2 & 9 & 6 & 3 & 3 & 1 & 4 & 4 & 7 \\ 1 & 3 & 2 & 1 & 6 & 8 & 0 & 6 & 2 \\ 9 & 9 & 0 & 4 & 2 & 1 & 5 & 3 & 1 \end{array} \right]$$

Następnie generujemy macierz  $\mathbf{I}'$  o wymiarach  $M' \times N'$ , w której umieszczamy obliczone ślady z każdej części zdjęcia wejściowego  $\mathbf{I}$ :

$$\mathbf{I}' = \left[ \begin{array}{ccc|ccc} 5 + 1 + 6 = 12 & 6 + 8 + 0 = 14 & 3 + 1 + 7 = 11 \\ \hline 2 + 3 + 0 = 5 & 3 + 6 + 1 = 10 & 4 + 6 + 1 = 11 \end{array} \right]$$

Stąd, otrzymujemy:

$$\mathbf{I}' = \left[ \begin{array}{ccc|ccc} 12 & 14 & 11 \\ \hline 5 & 10 & 11 \end{array} \right] \Rightarrow \left[ \begin{array}{ccc} 12 & 14 & 11 \\ 5 & 10 & 11 \end{array} \right]$$

Obraz  $\mathbf{I}'$  jest kompaktową reprezentacją cyfrowego odcisku palca zdjęcia  $\mathbf{I}$ . Jak łatwo zauważyć, wykonanie opisaney procedury pozwoliło zredukować wymiary cyfrowego odcisku palca z  $9 \times 6$  do  $3 \times 2$  pikseli. Tak więc kompaktową reprezentację odcisku palca  $\mathbf{I}'$  obliczoną na 6 zamiast 54 pikselach można podać na wejście algorytmów identyfikacji, zamiast oryginalnego obrazu  $\mathbf{I}$ . Zaletą powyższej procedury jest zatem brak konieczności przechowywania „dużych” odcisków palca zdjęć, co może mieć ogromne znaczenie w bazach danych zajmujących się gromadzeniem takich właśnie materiałów. Wprawdzie, w przeciwieństwie do algorytmu MSE-DSI, algorytm wymaga przechowywania macierzy jako cyfrowych odcisków palca, jednak jak pokazemy w części wyników eksperymentalnych, osiąga wyraźnie wyższą skuteczność klasyfikacji sensorów obrazujących. W przeciwieństwie jednak do algorytmu Lukás, macierze reprezentujące cyfrowe odciski palca mają zauważalnie mniejsze wymiary, co powinno ułatwić ich przechowywanie na dysku twardym. Podkreślmy także, że algorytm CompaRe nie wymaga odsumiania zdjęć  $\mathbf{I}$ .

### 3.3.3 Algorytm Vignetting-CT

Sensory obrazujące można także próbować identyfikować wykorzystując wady optyczne, którymi obarczone są aparaty cyfrowe. Do wad optycznych można zaliczyć: winietowanie obiektywu, dystorsję obiektywu czy aberrację chromatyczną. Winietowanie obiektywu jest wadą, która



**Algorytm 5** Algorytm Vignetting-CT**Wejście:** Obraz  $\mathbf{I}$  w formacie RGB o wymiarach  $M \times N$  pikseli;**Wyjście:** Wartość średnia intensywności pikseli  $\hat{d}$  obliczona z  $\mathbf{I}^{(\mathcal{R})}$ .

```

1: Ustal wartość  $\delta$ 
2:  $m = \lfloor \delta \cdot M \rfloor$ 
3:  $n = \lfloor \delta \cdot N \rfloor$ 
4: for  $i=1$  to  $m$  do
5:   for  $j=1$  to  $n$  do
6:     
$$\hat{s}_{a_1} = \frac{\sum_{i=1}^m \sum_{j=1}^n |\mathbf{I}_{ij} - F(\mathbf{I}_{ij}^{(\mathcal{R})})|}{mn}$$

7:     
$$\hat{s}_{a_2} = \frac{\sum_{i=1}^m \sum_{j=N-(n-1)}^N |\mathbf{I}_{ij} - F(\mathbf{I}_{ij}^{(\mathcal{R})})|}{mn}$$

8:     
$$\hat{s}_{a_3} = \frac{\sum_{i=M-(m-1)}^M \sum_{j=1}^n |\mathbf{I}_{ij} - F(\mathbf{I}_{ij}^{(\mathcal{R})})|}{mn}$$

9:     
$$\hat{s}_{a_4} = \frac{\sum_{i=M-(m-1)}^M \sum_{j=N-(n-1)}^N |\mathbf{I}_{ij} - F(\mathbf{I}_{ij}^{(\mathcal{R})})|}{mn}$$

10:   end for
11: end for
12:  $\hat{d} = \frac{1}{4} \sum_{k=1}^4 \hat{s}_{a_k}$ 
13: return  $\hat{d}$ 

```

objawia się jako spadek jasności zdjęcia zazwyczaj we wszystkich jego narożnikach względem środka [89, 93]. Identyfikację sensorów w oparciu o wady optyczne przeprowadza się głównie w rozumieniu aspektu SCMI [93], jednak w niniejszej pracy podejmiemy próbę wykorzystania takich metod w aspekcie ISCI.

Proponujemy algorytm Vignetting-Camera Tracing (Vignetting-CT), którego zasada działania sprowadza się do obliczenia średnich wartości intensywności pikseli w narożnikach zdjęcia. Oznaczmy przez  $\mathbf{I}$  zdjęcie wejściowe, przez  $\mathbf{I}^{(\mathcal{R})}$  czerwony kanał kolorystyczny zdjęcia  $\mathbf{I}$ , a przez  $F$  filtr odszumiający. Proponowana procedura jest przedstawiona jako algorytm 5.

Parametr  $\delta$  określa procentową wielkość części obrazu (linia 1), które mają być analizowane (linie 2-3). Następnie obliczane są średnie wartości intensywności pikseli w odszumionych częściach czerwonego kanału kolorystycznego zdjęcia  $F(\mathbf{I}^{(\mathcal{R})})$  (linie 6-9), których wymiary wa-

runkuje parametr  $\delta$ . Ostatecznie, obliczana jest wartość intensywności pikseli  $\hat{d}$  jako średnia  $s_{a_1}$ ,  $s_{a_2}$ ,  $s_{a_3}$  oraz  $s_{a_4}$  (linia 12). Proponujemy, aby wartość  $\hat{d}$  uznać jako cyfrowy odcisk palca do rozpoznania aparatu.

Przyjmijmy, że  $\mathbf{S} = |\mathbf{I} - F(\mathbf{I}^{(\mathcal{R})})|$ . Analiza wartości intensywności pikseli w  $\mathbf{S}$  pozwoliła zauważyć, że średnia intensywność pikseli w  $\mathbf{S}$  jest wyższa (co oznacza większą jasność) niż w rozważanych narożnikach. Przykładowy obraz jest przedstawiony na rys. 3.1. Przykładowe wartości dla kilkunastu obrazów z aparatu Nikon D70s (1) przedstawiono w tab. 3.3.



Rysunek 3.1: Przykładowy obraz  $\mathbf{I}$  (po lewej) i jego residuum  $\mathbf{S}$  (po prawej) obliczone jako bezwzględna różnica  $\mathbf{S} = |\mathbf{I} - F(\mathbf{I}^{(\mathcal{R})})|$ . Średnia wartość intensywności pikseli w  $\mathbf{S}$  wynosi 0,6, wartości intensywności w narożnikach  $\mathbf{S}$  wynoszą:  $\hat{s}_{a_1} = 0,36$ ,  $\hat{s}_{a_2} = 0,49$ ,  $\hat{s}_{a_3} = 0,34$ ,  $\hat{s}_{a_4} = 0,54$

Podobnie, jak w przypadku algorytmu MSE-DSI, uzyskane wartości intensywności pikseli odpowiadają pojedynczemu obrazowi. Aby więc uzyskać reprezentatywny wynik dla określonego aparatu, czynności algorytmu Vignetting-CT należy powtórzyć dla większej liczby zdjęć z określonego urządzenia (proponujemy co najmniej 45 zdjęć [104]). Posiadając „nowe” zdjęcie, które chcemy sklasyfikować, obliczamy dla niego wartości algorytmu Vignetting-CT, które następnie za pomocą algorytmu uczenia maszynowego (np.  $k$ -NN, SVM, itd.) możemy przyporządkować do określonego urządzenia.

Algorytm Vignetting-CT jest efektywny obliczeniowo, a liczba wykonywanych operacji jest zależna od wartości  $m$  oraz  $n$ . Ponadto, w naszej metodzie nie odsumiamy całości obrazu, a jedynie te fragmenty, w których będziemy analizować wartości intensywności pikseli. Co więcej, odsumianiu poddajemy jedynie pojedynczy kanał kolorystyczny, w odróżnieniu od algorytmu Lukás, który wymaga odsumienia wszystkich kanałów kolorystycznych zdjęcia  $\mathbf{I}$  w pełnej rozdzielczości. Przykładowo, jeśli obraz  $\mathbf{I}$  ma wymiary  $M = 6016$  na  $N = 4016$  oraz przyjmijmy,

Tabela 3.3: Przykładowe wartości intensywności pikseli dla aparatu Nikon D70s (1) w residuum **S**. Im wartość jest bliższa 0, tym ciemniejszy jest obszar zdjęcia

Obraz	Średnia intensywności pikseli w <b>S</b>	$\hat{s}_{a_1}$	$\hat{s}_{a_2}$	$\hat{s}_{a_3}$	$\hat{s}_{a_4}$
11.jpg	0,59	0,34	0,32	0,40	0,41
12.jpg	0,69	0,44	0,43	0,46	0,45
13.jpg	0,31	0,22	0,18	0,24	0,15
14.jpg	0,28	0,23	0,24	0,24	0,26
15.jpg	0,45	0,25	0,29	0,41	0,28
16.jpg	0,45	0,15	0,09	0,27	0,40
17.jpg	0,34	0,19	0,19	0,25	0,15
18.jpg	0,47	0,22	0,19	0,23	0,28
19.jpg	0,24	0,23	0,23	0,22	0,22
20.jpg	0,31	0,20	0,26	0,21	0,19
21.jpg	0,40	0,30	0,31	0,33	0,35
22.jpg	0,36	0,27	0,26	0,27	0,26
23.jpg	0,65	0,46	0,38	0,58	0,55
24.jpg	0,54	0,38	0,37	0,39	0,41

że  $\delta = 0,05$ , wówczas  $m = 300$  i  $n = 200$ . Oznacza to, że obliczenie średniej wartości intensywności pikseli dla  $M \times N$  wymaga przeanalizowania w zagnieżdżonej pętli 24 160 256 pikseli, podczas gdy dla uzyskanych wartości  $m$  i  $n$  będzie to jedynie  $4 \cdot 60\,000 = 240\,000$  pikseli. Widać więc wyraźnie redukcję liczby operacji wykonywanych wewnątrz pętli, co znacząco przyspiesza analizę. Przykładowe wymiary obrazów do przetworzenia w ramach algorytmu Lukás oraz sieci konwolucyjnych, a także dla Vignetting-CT dla  $\delta = 0,05$  są przedstawione w tab. 3.4.

Podobnie, jak w przypadku algorytmu Lukás, proponowana metoda opiera się na analizie niedoskonałości sensorów, które prowadzą do powstawania „przekłamań” wartości intensywności pikseli. Dla przykładu, jeśli wartość intensywności pewnego pikseli w fotografowanej scenie powinna wynieść 161, a kilka różnych sensorów „przydzieli” takiemu pikselowi wartości, np. 162, 159 lub 164, to będzie można wykorzystać taką informację do identyfikacji tych sensorów. Oczywiście, taka analiza jest tym dokładniejsza, im liczniejsza jest grupa analizowanych pikseli. W przypadku proponowanego algorytmu należy liczyć się z obniżeniem dokładności klasyfikacji z powodu znacząco zredukowanej liczby przetwarzanych pikseli, a więc liczba wykrytych w ten sposób defektów będzie niższa. Podobnie, jak w przypadku algorytmu MSE-DSI, zaletą

Tabela 3.4: Wymiary obrazów (w pikselach), którymi uczone są algorytmy Lukás oraz Vignetting-CT

Aparat cyfrowy	Wymiary I (Lukás i CNN)	Wymiary (Vignetting-CT, $\delta = 0,05$ )
Canon 5D Mark IV	6720 × 4480	336 × 224
Canon RP	6240 × 4160	312 × 208
Nikon D7200	6000 × 4000	300 × 200
Nikon Z7	8256 × 5504	412 × 275
Sony A7R III	7952 × 5304	398 × 625
Nikon D200	3872 × 2592	194 × 130
Samsung NV 15	3648 × 2736	182 × 136

procedury Vignetting-CT jest łatwość przechowywania cyfrowych odcisków palców, które są reprezentowane w formie liczbowej zamiast macierzowej.

### 3.3.4 Algorytm Distortion-CT

Dystorsja obiektywu jest jego wadą optyczną, która uwidacznia się na zdjęciach jako zniekształcenie kształtów poprzez odchylenie od rzutu prostoliniowego (np. krawędź budynku na zdjęciu obarconym dystorsją wydaje się odchylona od pionu itp.). Najczęściej spotykanymi rodzajami dystorsji jest dystorsja beczkowa (ang. *barrel*) oraz dystorsja poduszkowa (ang. *pincushion*). W dystorsji beczkowej uwypuklony jest środek kadru, natomiast w dystorsji poduszkowej środek kadru wygląda na „zapadnięty” [66]. Mówiąc nieco kolokwialnie, dystorsja występuje wtedy, gdy linie tworzone przez piksele, które w świecie rzeczywistym powinny być pionowe bądź poziome, na zdjęciu są nierównoległe do jego krawędzi, bądź zakrzywione. Należy jednak wziąć pod uwagę, że fakt występowania zakrzywionych linii w wyniku przechylenia aparatu podczas kadrowania (bądź umyślnego przechylenia zdjęcia w oprogramowaniu) nie oznacza występowania dystorsji.

W literaturze istnieje wiele modeli opisujących dystorsję, są to najczęściej modele wielomianowe [43, 66]. Proponujemy wykorzystanie modelu dystorsji opartego na wzorze Browna [28, 43], który został przedstawiony jako równanie 3.10.

$$p_u = p_d(1 + kr^2) \quad (3.10)$$

gdzie:

- $p_u(x_u, y_u)$  – punkt obrazu (piksel) pozbawiony dystorsji;

- $p_d(x_d, y_d)$  – punkt obrazu (piksel) z dystorsją;
- $k$  – parametr dystorsji;
- $r = \sqrt{(x_d - x_u)^2 + (y_d - y_u)^2}$ .

Po prostych przekształceniach, uzyskujemy:

$$k = \frac{p_u - p_d}{p_d \cdot r^2} \quad (3.11)$$

Rozważmy procedurę Distortion-Camera Tracing (Distortion-CT), zapisaną jako algorytm 6.

---

**Algorytm 6** Algorytm Distortion-CT
 

---

**Wejście:** Obraz  $I$  w reprezentacji RGB o wymiarach  $M \times N$  pikseli;

**Wyjście:** Parametr dystorsji  $k$ .

- 1: Ustal liczbę pikseli obciążonych dystorsją  $P$
  - 2: **for**  $i = 0$  **to**  $P$  **do**
  - 3:   **if** istnieje tylko jeden piksel  $p_u$  wolny od dystorsji **then**
  - 4:      $k^{(i)} = \frac{p_u - p_d^{(i)}}{p_d^{(i)} \cdot r^2(i)}$
  - 5:   **else if** istnieje wiele pikseli  $p_u^{(i)}$  wolnych od dystorsji **then**
  - 6:      $k^{(i)} = \frac{p_u^{(i)} - p_d^{(i)}}{p_d^{(i)} \cdot r^2(i)}$
  - 7:   **end if**
  - 8: **end for**
  - 9: **return**  $k$
- 

Proponujemy, aby obliczyć parametr dystorsji  $k$  dla określonej liczby zdjęć z różnych aparatów cyfrowych w celu sprawdzenia, czy występują pewne tendencje tego parametru, które mogą zostać wykorzystane do identyfikacji urządzenia. Zakładamy, że piksele obciążone na zdjęciu dystorsją „generują” zniekształcenia kształtów geometrycznych (np. krawędź budynku, która w rzeczywistości powinna stykać się z podłożem pod kątem prostym, na zdjęciu wydaje się zakrzywiona bądź przechylona pod innym kątem). Piksele obciążone dystorsją  $p_d^{(1)}, \dots, p_d^{(P)}$  można wyznaczyć za pomocą oprogramowania, np. Hugin Photo Sticher [5], bądź obliczyć ręcznie. Zatem znając piksele obciążone dystorsją oraz te, które są od niej wolne, możemy rozwiązać równanie 3.10, w którym  $k$  jest niewiadomą. Dla obu zbiorów pikseli, możemy otrzymać kilka różnych wartości parametru  $k$  – w takim przypadku proponujemy, aby obliczyć średnią wartość

uzyskanych parametrów  $k$ . Proponujemy także, aby parametr  $k$  traktować jako unikalny, który może być wykorzystany do rozróżniania aparatów.

Aby obliczenie parametru  $k$  było możliwe, obraz powinien posiadać przynajmniej jeden piksel, który jest wolny od dystorsji. Taki punkt można wyznaczyć na dwa sposoby. W przyjętym modelu zakładamy istnienie **jednego** punktu (piksele)  $p_u(x_u, y_u)$ . W literaturze często przyjmuje się, że takim punktem jest środek obrazu [66]<sup>3</sup>. Drugi sposób zakłada wyznaczenie **zbioru**  $i$  punktów  $p_u^{(i)}(x_u^{(i)}, y_u^{(i)})$  wolnych od dystorsji. W tym celu zdjęcie posiadające piksele obarczone dystorsją poddaje się algorytmowi korekcji dystorsji [100, 123]. Po takiej operacji można wyznaczyć współrzędne kartezjańskie punktów „uwolnionych” od dystorsji i wykorzystać je do analizy.

Zauważmy, że parametr  $k$  może przyjmować wartości dodatnie lub ujemne. W przypadku  $k > 0$  przyjmujemy, że dystorsja jest poduszkowa (ang. *pincushion*). Dla  $k < 0$ , dystorsja jest beczkowa (ang. *barrel*) [3].

Oczywiście warto też pamiętać, że dystorsja nie jest jedynie wadą obiektywu – w niektórych przypadkach może też być cechą wynikającą z praw fizyki (np. fotografowanie z bliska wysokiego obiektu obiektywem szerokokątnym – przykład na rys. 3.2). Rys. 3.3 przedstawia z kolei zdjęcie, na którym nie występują żadne piksele obarczone dystorsją.

Rozważmy prosty przykład, w którym obliczymy wartość parametru  $k$  dla pewnego aparatu cyfrowego.

**Przykład 3.2.** Załóżmy, że mamy współrzędne kartezjańskie następującego zbioru pikseli obarczonych dystorsją  $p_d$  oraz odpowiadających im pikseli  $p_u$  wolnych od dystorsji:

$$p_d^{(1)} = (100, 80), p_u^{(1)} = (110, 90)$$

$$p_d^{(2)} = (210, 100), p_u^{(2)} = (200, 110)$$

$$p_d^{(3)} = (215, 177), p_u^{(3)} = (217, 184)$$

Obliczamy:  $r^{(1)}$ ,  $r^{(2)}$  oraz  $r^{(3)}$ .

$$r^{(1)} = \sqrt{(100 - 110)^2 + (80 - 90)^2} = 14,14$$

$$r^{(2)} = \sqrt{(210 - 200)^2 + (100 - 110)^2} = 14,14$$

<sup>3</sup>Dla obrazów posiadających parzyste wymiary nie jest możliwe bezpośrednie wybranie takiego punktu (piksele), gdyż istnieją cztery punkty (piksele), które leżą w centrum obrazu – w takim przypadku do analizy wybieramy dowolny jeden z nich. Inną potencjalną trudnością może być fakt, że piksel leżący w centrum zdjęcia może być obciążony dystorsją, jednak w praktyce zdarza się to raczej rzadko, dlatego przyjmujemy brak występowania dystorsji na takim pikselu.



Rysunek 3.2: Zdjęcie obiektu z widoczną dystorsją beczkową ( $k = -0,012$ ), niektóre piksele obarczone dystorsją są zaznaczone kolorem czerwonym (wyznaczono za pomocą oprogramowania Hugin [5])



Rysunek 3.3: Zdjęcie, na którym dystorsja nie występuje

$$r^{(3)} = \sqrt{(215 - 217)^2 + (177 - 184)^2} = 7,28$$

Wykorzystując równanie 3.10, obliczamy wartości  $k^{(1)}$  dla  $p_d^{(1)}$  i  $p_u^{(1)}$ :

$$k_X^{(1)} = \frac{110 - 100}{100 \cdot 14,14^2} = 0,0005; \quad k_Y^{(1)} = \frac{90 - 80}{80 \cdot 14,14^2} = 0,0006$$

Analogicznie, dla kolejnych par pikseli  $p_d^{(2)}$ ,  $p_u^{(2)}$  oraz  $p_d^{(3)}$ ,  $p_u^{(3)}$ :

$$k_X^{(2)} = \frac{200 - 210}{210 \cdot 14,14^2} = -0,0002; \quad k_Y^{(2)} = \frac{110 - 100}{100 \cdot 14,14^2} = 0,0005$$

$$k_X^{(3)} = \frac{217 - 215}{215 \cdot 7,28^2} = 0,0002; \quad k_Y^{(3)} = \frac{184 - 177}{177 \cdot 7,28^2} = 0,0007$$

Ostatecznie, obliczamy średnią wartość wszystkich obliczonych wartości  $k_{X,Y}^{(i)}$ :

$$k = \frac{1}{6} \left( 0,0005 + 0,0006 + (-0,0002) + 0,0005 + 0,0002 + 0,0007 \right) = 0,0004$$

Stąd,  $k = 0,0004$  jest szukany parametrem dystorsji.

Parametr  $k$  powinien być obliczony dla większej liczby zdjęć z określonego urządzenia. W dalszej kolejności można obliczyć wartość średnią uzyskanych  $k_i$  dla danego aparatu, a uzyskanymi wartościami nauczyć wybrany klasyfikator w celu sklasyfikowania nowego zdjęcia. Nie definiujemy jednak, jak duża powinna być liczba pikseli  $P$  obarczonych dystorsją, na podstawie której obliczamy parametr  $k$ .

### Uwagi do algorytmów Vignetting-CT oraz Distortion-CT

Należy zauważyć, że funkcjonalność algorytmów Vignetting-CT oraz Distortion-CT jest ograniczona. Identyfikacja jest możliwa jedynie w takim przypadku, gdy analizowane zdjęcia przedstawiają dokładnie ten sam kadr, a także, gdy są wykonane w takich samych warunkach oświe-



tleniowych (algorytm Vignetting-CT). Oznacza to, iż analiza zdjęć z nawet nieznacznie przesuniętymi kadrami (bądź użytymi różnymi ogniskowymi) będzie skutkowałą obniżeniem jakości klasyfikacji końcowej. W praktyce trudno będzie jednak znaleźć rzeczywistą sytuację, w której możliwe będzie dysponowanie obrazami przedstawiającymi dokładnie ten sam kadr. Taki scenariusz jest jednak zazwyczaj możliwy w przypadku dedykowanych zbiorów danych, w których zdjęcia pochodzące z różnych aparatów, istotnie przedstawiają taką samą scenę [63]. Ponadto, wykonanie algorytmu Distortion-CT nie zawsze jest możliwe, ponieważ zjawisko dystorsji może nie występować. Nieco większą uniwersalność oferuje algorytm Vignetting-CT. Jeśli zjawisko winietowania nie występuje w sposób wprost widoczny na zdjęciu, możemy nadal przeanalizować piksele na zdjęciu zgodnie z procedurą Vignetting-CT. Dlatego więc proponowane algorytmy można polecić do przetwarzania wstępnego zbioru zdjęć w celu szybkiego ustalenia, czy dane zdjęcie zostało wykonane przez jedno bądź więcej urządzeń.

### 3.3.5 Identyfikacja z wykorzystaniem konwolucyjnych sieci neuronowych

Sztuczne sieci neuronowe są sprzętowym lub programowym modelem działania ludzkiego mózgu. W dużym uproszczeniu, mózg ludzki składa się z komórek nerwowych zwanych neuronami, które zbierają dane (sygnały elektrochemiczne z innych komórek i receptorów) za pomocą dendrytów i wyprowadzają je za pomocą aksonów i synaps. Sztuczne sieci neuronowe są próbą odtworzenia substytutów funkcjonalnych mózgu, zdolnych do nauki i rozwiązywania problemów. Zależność pomiędzy danymi wejściowymi a wyjściowymi może być interpretowana jako rozwiązanie pewnego problemu, np. rozpoznawanie obrazu lub klasyfikacja danych.

Sieci neuronowe mają jednak pewne ograniczenia. W przypadku rozpoznawania złożonych struktur (np. twarzy), sieć musiałaby mieć wiele warstw i neuronów, a jej trenowanie byłoby bardzo czasochłonnym procesem. Rozwiązaniem tego problemu okazały się konwolucyjne (zwane też splotowymi) sieci neuronowe (CNN). Sieci CNN posiadają układ połączeń pomiędzy neuronami inspirowany budową narządu wzroku u ludzi. Neurony tworzą wiele warstw, z których każda ma specyficzne zastosowanie. Od wielu lat cieszą się popularnością w wielu aspektach badawczych [125, 151, 157]. Znajdują zastosowanie w szerokiej gamie obszarów takich, jak widzenie komputerowe (ang. *computer vision*) [52, 106], klasyfikacja [152, 159], rozpoznawanie obiektów [10, 76, 94, 122], przetwarzanie tekstów [119, 120], czy przetwarzanie obrazów [72, 98]. Identyfikacja aparatów cyfrowych na podstawie zdjęć nie jest wyjątkiem, a niektóre z prac, które poruszają ten wątek, zostały już przytoczone w niniejszej rozprawie [26, 54, 95, 110, 126, 144, 145, 150]. Proponowanie zatem nowej sieci konwolucyjnej,

służącej do rozpoznawania sensorów raczej nie jest już możliwe, jednak z uwagi na prowadzone nieprzerwanie prace badawcze w ośrodkach naukowych na całym świecie, które mają na celu doskonalenie sieci CNN, można zaproponować pewne usprawnienie istniejących rozwiązań. Rozważmy sieć konwolucyjną o następującej strukturze, z eksperymentalnie dobranymi metaparametrami:

1. Pierwsza konwolucyjna warstwa zawierająca 32 filtry z oknem  $5 \times 5$  i krokiem 1 wraz z ReLU jako metodą aktywacji;
2. Warstwa *max-pooling* rozmiaru  $2 \times 2$  z krokiem 2;
3. Druga konwolucyjna warstwa zawierająca 64 filtry z oknem  $5 \times 5$  oraz ReLU jako metodą aktywacji;
4. Warstwa *max-pooling* rozmiaru  $2 \times 2$ ;
5. Trzecia konwolucyjna warstwa zawierająca 128 filtrów z oknem  $5 \times 5$  oraz ReLU jako metodą aktywacji;
6. Warstwa *max-pooling* rozmiaru  $2 \times 2$ ;
7. Warstwy w pełni połączone do klasyfikacji.

Jak pokażemy w rozdz. 4, powyższa sieć może być z powodzeniem wykorzystana do efektywnej identyfikacji sensorów obrazujących. Każda z warstw konwolucyjnych jest aktywowana za pomocą funkcji aktywacji. W literaturze dostępnych jest wiele różnych funkcji aktywacji, np. ReLU, ELU, Leaky ReLU, SELU, itd. Jedną z najbardziej znanych funkcji aktywacji jest Rektyfikowana Jednostka Liniowa (ang. *Rectified Linear Unit* – ReLU) [118]. Przypomnijmy definicję tej funkcji:

$$\text{ReLU}(x) = \max(0, x) \quad (3.12)$$

gdzie  $x$  jest wejściem neuronu. Modyfikacja funkcji ReLU, zwana Leaky ReLU, jest zdefiniowana następująco [153]:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{dla } x > 0 \\ \alpha x & \text{dla } x \leq 0 \end{cases} \quad (3.13)$$

Jeśli wejście  $x$  jest większe od zera, wyjściem jest  $x$ . Jeśli wartość  $x$  jest mniejsze od zera, funkcja zwraca na wyjściu iloczyn  $\alpha x$ . Wartość  $\alpha$  jest zazwyczaj wartością z zakresu  $[0,1; 0,3]$ .

W ostatnim czasie coraz częściej można zaobserwować wykorzystanie funkcji zwanej *Scaled Exponential Linear Unit* – SELU [87]. Aktywacje SELU samoczynnie normalizują sieć neuronową. Dzięki normalizacji składniki sieci (wagi, obciążenia (ang. *biases*) i aktywacje) będą miały średnią równą zero i odchylenie standardowe równe jeden. Definicja funkcji SELU jest następująca:

$$\text{SELU}(x) = \lambda \begin{cases} x & \text{dla } x > 0 \\ \alpha \exp(x) - \alpha & \text{dla } x \leq 0 \end{cases} \quad (3.14)$$

gdzie  $\alpha \approx 1,6733$  oraz  $\lambda \approx 1,0507$ . Proponujemy, aby do aktywacji warstw konwolucyjnych wykorzystać funkcję SELU zamiast ReLU. Dotychczasowe sieci konwolucyjne, które można znaleźć w literaturze, najczęściej wykorzystują funkcję ReLU do aktywacji warstw konwolucyjnych [26, 144]. Weryfikacja eksperymentalna pozwoliła zauważyć, że zastosowanie funkcji SELU zamiast ReLU pozwala znacząco skrócić czas uczenia sieci konwolucyjnej. Omówimy szerzej ten aspekt w rozdz. 4.

## 3.4 Podsumowanie

W niniejszym rozdziale scharakteryzowano algorytmy identyfikacji sensorów obrazujących. Przedstawione zostały algorytmy i metody znane z literatury (Lukás, Bondi, Tuama, Valsesia oraz Li). Zaproponowano także autorskie algorytmy: MSE-DSI, CompaRe, Vignetting-CT, Distortion-CT oraz przeanalizowano metodę opartą na konwolucyjnych sieciach neuronowych. Wszystkie proponowane algorytmy mogą być wykorzystywane do identyfikacji sensorów w oparciu o aspekt ISCI.

Algorytm MSE-DSI pod względem istoty działania jest zbliżony do algorytmu Lukás, jednakże nie wymaga odszumiania wszystkich kanałów kolorystycznych zdjęcia wejściowego, a także ogranicza się do przetwarzania jedynie niewielkich wycinków zdjęć. Dzięki temu, liczba operacji, które są potrzebne do jego wykonania jest znacząco zredukowana. Kolejną zaletą algorytmu jest identyfikacja sensorów na podstawie wartości liczbowej (MSE), która pełni funkcję cyfrowego odcisku palca, a nie macierzy, której wymaga algorytm Lukás. Fakt ten znacząco ułatwia przechowywanie i zarządzanie cyfrowymi odciskami palca na dysku twardym komputera.

Algorytm CompaRe jest „odpowiedzią” na algorytm Valsesia, którego celem jest identyfikowanie sensorów na podstawie macierzy cyfrowego odcisku palca, jednak o wymiarach znacząco zredukowanych względem wymiarów zdjęć wejściowych. W przeciwieństwie do propozycji Valsesia, algorytm CompaRe nie wymaga mnożenia macierzy, co zasadniczo jest operacją wymagającą obliczeniowo [39, 88].

Algorytmy Vignetting-CT oraz Distortion-CT mogą być wykorzystane do bardzo szybkiej identyfikacji aparatów w oparciu o zdjęcia tych samych kadrów, wykonane w tych samych warunkach oświetleniowych. Wprawdzie możliwość praktycznego zastosowania tych algorytmów jest raczej ograniczona, mogą być one jednak przydatne w kontekście szybkiej analizy urządzeń pochodzących z naukowych zbiorów danych, takich jak na przykład baza Dresden Image Database.

W rozdziale zaproponowano także modyfikację typowej struktury sieci konwolucyjnej, polegającą na zastosowaniu funkcji SELU, zamiast ReLU. Rozwiązanie takie pozwala znacząco skrócić czas uczenia sieci konwolucyjnej.

# Rozdział 4

## Weryfikacja eksperymentalna

W niniejszym rozdziale zostaną przedstawione wyniki klasyfikacji aparatów cyfrowych na podstawie zdjęć przez proponowane algorytmy w odniesieniu do algorytmów z literatury. Analiza jest przeprowadzona z wykorzystaniem reprezentatywnej liczby zdjęć, pochodzących z wielu nowoczesnych urządzeń. Opisane zostaną wykorzystane zbiory zdjęć oraz miary oceny jakości klasyfikacji. Przedstawione zostaną macierze konfuzji prezentujące dokładne wyniki klasyfikacji, a także porównamy czas działania algorytmów.

### 4.1 Zbiory zdjęć

Do eksperymentów zostały wykorzystane trzy zbiory zdjęć. Pierwszy obejmuje zdjęcia pochodzące z proponowanego zbioru IMAGINE, drugi stanowią zdjęcia ze zbioru Dresden Image Database [63], natomiast trzecim zbiorem są zdjęcia pochodzące z prywatnych zbiorów autora rozprawy. We wszystkich przypadkach wykorzystujemy kolorowe zdjęcia JPEG, które nie były modyfikowane w żadnym zewnętrznym oprogramowaniu.

Zdjęcia ze zbioru IMAGINE obejmują 53 nowoczesne urządzenia, które zostały szczegółowo scharakteryzowane w rozdz. 2. Oznaczenia urządzeń: Acer Liquid Jade S [Ac1], Apple iPhone 5S [Ap1], Canon EOS 1D X Mark II [Ca1], Canon EOS 5D Mark IV [Ca2], Canon EOS 6D Mark II [Ca3], Canon EOS 750D [Ca4,Ca5] (2 urządzenia), Canon EOS 760D [Ca6,Ca7] (2 urządzenia), Canon EOS M3 [Ca8,Ca9] (2 urządzenia), Canon EOS M5 [C10,C11] (2 urządzenia), Canon EOS M50 [C12,C13] (2 urządzenia), Canon EOS 90D [C14], Canon EOS M100 [C15], Canon EOS M200 [C16], Canon EOS R [C17], Canon EOS R5 [C18], Canon EOS R6 [C19], Canon EOS RP [C20], Canon PowerShot G9 X Mark II [C21], Canon PowerShot SX270 HS [C22], DJI Spark [DJ1], Fujifilm X-200 [Fu1], Lenovo K5 Plus [Le1], LG K10 [LG1], Microsoft Lumia 640 [Mi1],

Nikon D5 [Ni1], Nikon D6 [Ni2], Nikon D500 [Ni3], Nikon D610 [Ni4], Nikon D750 [Ni5,Ni6] (2 urządzenia), Nikon D780 [Ni7], Nikon D810 [Ni8], Nikon D850 [Ni9], Nikon D3100 [N10,N11] (2 urządzenia), Nikon D5600 [N12,N13] (2 urządzenia), Nikon D7200 [N14,N15] (2 urządzenia), Nikon P100 [N16], Nikon Z6 [N17,N18] (2 urządzenia), Nikon Z6 II [N19], Nikon Z7 [N20], Nikon Z7 II [N21], Nokia 2.2 [No1], Samsung Galaxy A40 [Sa1], Samsung Galaxy Ace 3 [Sa2], Samsung Galaxy S7 [Sa3,Sa4] (2 urządzenia), Samsung Galaxy SIII mini [Sa5], Samsung Galaxy Tab A 10.1 [Sa6], Samsung Galaxy Trend 2 Lite [Sa7], Samsung Omnia II [Sa8], Sony A1 [So1], Sony A7R III [So2], Sony A7S [So3], Sony A9 [So4], Sony ActionCam AS200V [So5], Sony RX100 VI [So6], Yuneec Breeze 4K [Yu1]. Całkowita liczba wykorzystanych zdjęć wynosi 1919, co oznacza, że co najmniej 30 zdjęć z każdego urządzenia zostało wykorzystanych do analizy.

Zbiór Dresden Image Database [63] (w dalszej części rozprawy będziemy go nazywać w skrócie Dresden) jest niezwykle popularny w rozważanej tematyce i od wielu lat cieszy się dużym zainteresowaniem ze strony badaczy [45, 62, 104]. Zawiera on setki zdjęć z różnych typów aparatów, które obejmują (między innymi): Agfa DC 733s [Ag1], Agfa DC 830i [Ag2], Agfa Sensor 505 [Ag3], Agfa Sensor 530s [Ag4], Canon Ixus 55 [Ca1], Canon Ixus 70 [Ca2-Ca4] (3 urządzenia), Casio EX Z150 [Ca5-9] (5 urządzeń), Kodak M1063 [Ko1-5] (5 urządzeń), Nikon CoolPix S710 [Ni1-5] (5 urządzeń), Nikon D70 [Ni6-7] (2 urządzenia), Nikon D70s [Ni8-9] (2 urządzenia), Nikon D200 [N10-11] (2 urządzenia), Olympus 1050SW [Ol1-5] (5 urządzeń), Praktica DCZ5 [Pr1-5] (5 urządzeń), Rollei RCP 7325XS [Ro1-3] (3 urządzenia), Samsung L74 [Sa1-3] (3 urządzenia) oraz Samsung NV15 [Sa4-6] (3 urządzenia). W większości przypadków, zdjęcia przedstawiają te same kadry sfotografowane za pomocą różnych urządzeń. Wszystkie rozważane aparaty są wyposażone w sensory obrazujące wykonane w technologii CCD (ang. *charge-coupled device*). Do analizy zostało wykorzystanych 11 787 zdjęć z 48 aparatów. W nawiasach kwadratowych oznaczamy skróty aparatów, których używać będziemy w macierzach konfuzji.

Trzeci zbiór (zwany dalej „zbiorem smartfonów”) zawiera zdjęcia wykonane współczesnymi smartfonami. Wykorzystano urządzenia: Apple iPhone 6 [Ap1], Asus ZenFone 2 [As1], HTC One M9 [HTC], Huawei P8 [Hua], LG G3 [LG1], LG G4 [LG2], Nokia Lumia 1020 [Lu1], Nokia Lumia 1520 [Lu2], Samsung Galaxy Note 4 [Sa1], Samsung Galaxy S6 [Sa2], Sony Xperia Z3 [So1] oraz Sony Xperia Z3+ [So2]. Wszystkie urządzenia zawierają sensory wykonane w technologii CMOS (ang. *complementary metal-oxide-semiconductor*). Wykorzystano 264 zdjęcia z 12 urządzeń.

Zarówno proponowane algorytmy, jak i metody z literatury będziemy testować głównie na zbiorach IMAGINE oraz Dresden Image Database z uwagi na reprezentatywną liczbę urządzeń oraz zdjęć. Podkreślimy, że wszystkie algorytmy będą uruchamiane na każdym ze zbiorów osobno. Działanie takie ma na celu wykazać przydatność nie tylko proponowanych algorytmów, ale

również użyteczność zbioru IMAGINE w odniesieniu do zbioru Dresden Image Database.

Ponadto, algorytm MSE-DSI został również przetestowany w kontekście identyfikacji skanerów płaskich na podstawie skanów. Wykorzystano 10 następujących urządzeń: Brother MFC 9970CDW, Canon C2020i, HP Deskjet F4180, HP Laser Jet M1005 MFP, HP ScanJet 3670, HP ScanJet PLS 2800, OKI MC562w, Plustek, Ricoh SP 112SU oraz Samsung SCX-3205. Łącznie użyto 290 skanów kolorowych zdjęć w formacie JPEG (po 29 z każdego urządzenia). Wszystkie skanery były podłączone do komputerów z zainstalowanym systemem Microsoft Windows w wersji 8.1 lub 10. Wykorzystana została aplikacja „Faksowanie i skanowanie w systemie Windows” z domyślnymi ustawieniami; zdjęcia zostały zeskanowane w rozdzielczości 300 dpi (ang. *dots per inch*).

Eksperymenty zostały przeprowadzone z wykorzystaniem dwóch komputerów:

1. Notebook MSI (nazywany dalej komputerem 1), wyposażony w procesor Intel Core i5-7300HQ@2,5GHz, 32 gigabajty pamięci RAM, kartę graficzną nVidia GTX 1050 posiadającą 4 gigabajty pamięci wideo oraz dysk M2 SSD trzeciej generacji;
2. Komputer PC (nazywany dalej komputerem 2), wyposażony w procesor Intel Core i5-10400@2,9GHz, 32 gigabajty pamięci RAM, kartę graficzną nVidia RTX 3060 posiadającą 12 gigabajtów pamięci wideo oraz dysk M2 SSD czwartej generacji.

Proponowane algorytmy zostały zaimplementowane w środowisku MATLAB; skrypty implementujące konwolucyjne sieci neuronowe zostały zaprogramowane w języku Python z użyciem framework'a Keras.

## 4.2 Klasyfikacja

### 4.2.1 Miary oceny

Ocenę klasyfikacji przeprowadzono w oparciu o standardową i powszechnie znaną miarę stosowaną w uczeniu maszynowym: poprawność (dokładność) klasyfikacji (ang. *accuracy* – ACC). Jest ona zdefiniowana następująco:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

TP/TN oznacza „prawdziwie pozytywny/prawdziwie negatywny”; FP/FN oznacza „fałszywie pozytywny/fałszywie negatywny”. TP oznacza numer instancji prawidłowo zaklasyfikowanych do danej klasy; TN oznacza instancje prawidłowo odrzucone. FP oznacza instancje nieprawidłowo przydzielone do danej klasy; FN to instancje nieprawidłowo odrzucone.

Wyniki klasyfikacji przedstawimy za pomocą macierzy konfuzji, które są uznaną i powszechnie używaną metodą prezentacji uzyskanych rezultatów [55, 104, 107, 113, 138].

Należy także podkreślić, że z uwagi na dużą liczbę urządzeń, w niniejszej pracy będziemy przedstawiać najważniejsze wyniki zamiast całości. Przykładowo, dla zbioru Dresden Image Database, z którego dysponujemy zdjęciami dla 48 urządzeń, wyniki przedstawimy jedynie z wybranych aparatów (gdyż pełna macierz konfuzji z wynikami klasyfikacji musiałaby mieć w takim przypadku wymiary  $48 \times 48$ ); w przypadku 53 urządzeń z proponowanego zbioru IMAGINE, macierz konfuzji miałaby wymiary  $53 \times 53$ .

### 4.2.2 Algorytmy z literatury

Wyniki proponowanych algorytmów porównamy z następującymi algorytmami i metodami: Lukás [104], Bondi [26], Tuama [145], Mandelli [108], Kirchner & Johnson [85], Valsesia [148], Li [99] oraz Khanna [83]. W przypadku metod opartych na konwolucyjnych sieciach neuronowych (Bondi, Tuama, Mandelli oraz Kirchner & Johnson) stosujemy pewną modyfikację oryginalnych struktur sieci polegającą na zastąpieniu funkcji aktywacji ReLU funkcją SELU w celu przyspieszenia uczenia sieci.

### 4.2.3 Algorytm Lukás – procedura klasyfikacji

W przypadku algorytmu Lukás klasyfikacja nowego zdjęcia jest przeprowadzana w oparciu o współczynnik korelacji  $\rho$  (rozdz. 3, wzór 3.8), który przyjmuje wartości z przedziału  $[0, 1]$  (0 – brak korelacji, 1 – maksymalna korelacja). „Ręczna” inspekcja wartości tego współczynnika pozwoliła zauważyć, że jeśli określone zdjęcie nie zostało wykonane badanym aparatem, to wartości  $\rho$  są bardzo małe, np.  $\rho = 0,001$  lub nawet  $\rho = 0,0001$ . Dla zdjęć wykonanych określonym aparatem wartości osiągają najczęściej  $\rho > 0,5$ , jednakże wielokrotnie zdarzało się, że ten współczynnik był równy  $\rho = 0,1$ , a nawet  $\rho = 0,01$  w przypadku zdjęcia wykonanego danym urządzeniem. Niemal nigdy nie zdarzało się, by dla zdjęć niewykonanych badanym aparatem współczynnik osiągał  $\rho = 0,01$  lub większe wartości – zazwyczaj były to wartości o rząd, a nawet o dwa rzędy mniejsze. Dlatego więc przyjmujemy, że już  $\rho > 0,01$  ujawnia, że badane zdjęcie zostało wykonane testowanym aparatem [19]. Przykładowe wartości współczynnika  $\rho$  (dla zbiorów IMAGINE oraz Dresden) można zobaczyć w dodatku A do niniejszej pracy.



## 4.3 Wyniki klasyfikacji

### 4.3.1 Algorytm MSE-DSI

Algorytm MSE-DSI został przetestowany w dwóch kontekstach. W ramach pierwszego kontekstu sprawdzono jakość klasyfikacji algorytmu w identyfikacji modeli aparatów cyfrowych (w tym urządzeń mobilnych, np. smartfonów). Wyniki zostały porównane z algorytmem Lukás [104] oraz z klasyfikatorami opartymi na konwolucyjnych sieciach neuronowych, zaproponowanymi w pracach Bondi [26] oraz Tuama [145]. Drugim aspektem było sprawdzenie algorytmu w identyfikacji 10 modeli skanerów płaskich, a wyniki porównano z algorytmem zaproponowanym przez Khanna [83, 84]. Istota działania algorytmu Khanna jest zbliżona do algorytmu Lukás. Jak wspominaliśmy w rozdz. 3, proponujemy uczenie algorytmu MSE-DSI za pomocą wycinków zdjęć o wymiarach  $512 \times 512$ , zamiast zdjęć w ich pełnej rozdzielczości. Przeprowadzono eksperymenty (dla przejrzystości pomijamy ich prezentację w niniejszej rozprawie), które wykazały, że wymiary  $512 \times 512$  zapewniają akceptowalną skuteczność klasyfikacji przy jednoczesnym szybkim czasie przetwarzania zdjęć. Zastosowanie znacznie mniejszych wycinków o wymiarach  $128 \times 128$  lub  $256 \times 256$  zbyt wyraźnie obniżało jakość klasyfikacji, która była niższa od modelu losowego. Oczywiście, uczenie algorytmu z użyciem wycinków o większych wymiarach, np.  $1024 \times 1024$  zwiększa skuteczność klasyfikacji, jednak (co intuicyjne), zwiększa także czas przetwarzania obrazów.

**Klasyfikacja – aparaty cyfrowe** Klasyfikację przeprowadzono w oparciu o algorytm  $k$ -najbliższych sąsiadów (ang. *k-nearest neighbors*) z eksperymentalnie wybraną wartością  $k = 5$ . Wartości algorytmu MSE-DSI zostały obliczone dla wszystkich obrazów z określonego aparatu, następnie obliczono ich wartość średnią, która reprezentuje określony aparat. Zdjęcia są klasyfikowane na zasadzie „wielu głosów” (ang. *plurality voting*), tj. do klasy najczęstszej spośród  $k$  najbliższych wartości MSE-DSI. Eksperymenty przeprowadzono z 10-krotną walidacją krzyżową (ang. *cross validation*). Klasyfikacja z wykorzystaniem algorytmów z literatury została przeprowadzona zgodnie z procedurami zaproponowanymi przez ich autorów.

Z uwagi na dużą liczbę macierzy konfuzji, dla przejrzystości w niniejszym rozdziale prezentujemy jedynie macierze konfuzji algorytmu MSE-DSI oraz Bondi na zbiorze IMAGINE (tab. 4.1 i 4.2). Macierze konfuzji przedstawiające rezultaty klasyfikacji dla dla zbioru IMAGINE (algorytmy Lukás, Tuama), a także bazy Dresden Image Database (algorytmy MSE-DSI, Lukás, Bondi oraz Tuama) zostały zaprezentowane w dodatku B do niniejszej pracy, odpowiednio jako tab. B.1-B.2 oraz jako tab. B.3-B.6 (podrozdz. B.1).

Poprawność klasyfikacji rozważanych algorytmów dla wszystkich urządzeń z obu zbioru



Tabela 4.3: Poprawność klasyfikacji ACC [%] obliczona na podstawie klasyfikacji wszystkich urządzeń

	IMAGINE	Dresden
Algorytm	ACC	ACC
MSE-DSI	74,0	77,0
Lukás	95,0	95,0
Bondi	95,0	94,0
Tuama	94,0	95,0

Uzyskane wyniki na podstawie dwóch zbiorów danych jednoznacznie wskazują niższą skuteczność klasyfikacji proponowanego algorytmu MSE-DSI w odniesieniu do algorytmu Lukás, Bondi oraz Tuama. Zarówno dla proponowanego zbioru IMAGINE, jak i zbioru Dresden Image Database, algorytm MSE-DSI osiąga poprawność klasyfikacji (odpowiednio) ACC = 73,0 i ACC = 75,0%. Dla algorytmu Lukás jest to (odpowiednio): ACC = 95,0 i ACC = 94,0%; Bondi: ACC = 96,0 i ACC = 96,0%; Tuama ACC = 96,0 i ACC = 96,0%. Dane zaprezentowane w macierzach konfuzji wyraźnie wskazują na stabilność klasyfikacji algorytmów z literatury, w przypadku których błędne rozpoznania aparatów cyfrowych na podstawie zdjęć zdarzają się rzadko. W przypadku proponowanego algorytmu, „różnorodność” klasyfikacji jest zauważalnie większa, np. Canon EOS R6 [C19] w 70% przypadków został rozpoznany poprawnie, jednak 11% zdjęć wykonanych tym urządzeniem zostało wskazanych jako wykonanych urządzeniem Nikon D500 [Ni3]. Na ogół klasyfikacyjne pomyłki algorytmu MSE-DSI w większości przypadków nie przekraczają 7%, a ogólna dokładność oscyluje wokół poziomu 75%, co prowadzi do intuicyjnego wniosku, że 3 na 4 zdjęcia są poprawnie przyporządkowane do źródłowego aparatu cyfrowego.

Wyniki uzyskane na podstawie pełnej klasyfikacji (tj. z wykorzystaniem wszystkich urządzeń ze zbioru Dresden oraz IMAGINE) są podobne do wyników zaprezentowanych w macierzach konfuzji. W przypadku zbioru IMAGINE, wyniki rozważanych algorytmów (odpowiednio) są następujące: 74,0; 95,0; 95,0 i 94,0%. Dla zbioru Dresden, dokładność ACC dla algorytmu MSE-DSI wynosi 77,0%; dla algorytmów Lukás, Bondi, Tuama, odpowiednio 95,0; 94,0 i 95,0%.

Częstsze pomyłki proponowanego algorytmu wynikają z mniejszej ilości przetwarzanych danych (przypomnijmy, że algorytm MSE-DSI uczymy za pomocą wycinków zdjęć o wymiarach  $512 \times 512$  na podstawie jednego kanału kolorystycznego). Tak więc zarówno algorytm Lukás, jak i konwolucyjne sieci są bardziej precyzyjnie wytrenowane. Niemniej, wyniki klasyfikacji algorytmu MSE-DSI można uznać za satysfakcjonujące, ponieważ są zauważalnie wyższe od klasyfikacji modelem losowym. Podsumowując, otrzymujemy intuicyjny wniosek, że mniejsza liczba danych

uczących powoduje obniżenie jakości klasyfikacji.

Warto dodać, że w pracy [16] (C2) przedstawiliśmy wyniki klasyfikacji także dla pozostałych kanałów kolorystycznych, jednak nie zaobserwowaliśmy istotnych zmian w jakości klasyfikacji, dlatego dla przejrzystości pomijamy prezentację tych wyników w niniejszej pracy.

**Klasyfikacja – skanery płaskie** Wyniki klasyfikacji dla 10 egzemplarzy skanerów płaskich zostały przedstawione jako macierze konfuzji w tab. 4.4 oraz 4.5.

Tabela 4.4: Macierz konfuzji, identyfikacja modelu (algorytm MSE-DSI), ACC = 82,0% (symbol \* oznacza wartości mniejsze niż 1,0%)

	Bro	Can	HP1	HP2	HP3	HP4	OKI	Plu	Ric	Sam
Bro	89,7	3,4	*	*	*	6,9	*	*	*	*
Can	10,3	41,4	*	10,3	20,7	17,2	*	*	*	*
HP1	*	*	41,4	*	*	*	*	10,3	*	48,3
HP2	*	*	*	93,1	*	*	*	*	6,9	*
HP3	*	17,2	*	6,9	75,9	*	*	*	*	*
HP4	*	13,8	*	*	*	86,2	*	*	*	*
OKI	*	*	*	*	*	*	100,0	*	*	*
Plu	*	*	*	*	*	*	*	100,0	*	*
Ric	*	*	*	*	*	*	*	*	100,0	*
Sam	*	*	24,1	*	*	*	*	*	*	75,9

Podobnie, jak w przypadku identyfikacji aparatów cyfrowych, algorytm MSE-DSI uzyskał niższą poprawność klasyfikacji niż algorytm Khanna, choć różnica jest mniej wyraźna, jak w przypadku klasyfikacji aparatów cyfrowych. Dla proponowanego algorytmu jest to 82,0%, natomiast dla algorytmu Khanna 91,0%. Wyjaśnienie jest analogiczne, jak w przypadku aparatów cyfrowych – algorytm MSE-DSI przetwarza mniejszą ilość danych.

### 4.3.2 Algorytm CompaRe

W celu weryfikacji skuteczności klasyfikacji algorytmu CompaRe, przeprowadzono dwa eksperymenty. W ramach eksperymentu I uzyskane wyniki porównano z algorytmami wykorzystującymi kompaktową reprezentację cyfrowego odcisku palca: Valsesia [148] oraz Li [99]. W ramach eksperymentu II proponowany algorytm porównano ze standardowymi algorytmami identyfikującymi sensory, takimi jak: algorytm Lukás oraz sieci konwolucyjne: Bondi, Tuama, Mandelli i Kirchner & Johnson.

Tabela 4.5: Macierz konfuzji, identyfikacja modelu (algorytm Khanna), ACC = 91,0% (symbol \* oznacza wartości mniejsze niż 1,0%)

	Bro	Can	HP1	HP2	HP3	HP4	OKI	Plu	Ric	Sam
Bro	85,0	*	*	5,0	*	*	1,0	*	8,0	2,0
Can	*	96,0	*	*	2,0	*	*	1,0	*	*
HP1	2,0	*	90,0	*	*	*	5,0	*	*	1,0
HP2	*	*	8,0	87,0	5,0	*	*	*	*	*
HP3	*	2,0	1,0	2,0	91,0	*	1,0	*	1,0	1,0
HP4	1,0	*	1,0	*	*	92,0	*	2,0	*	2,0
OKI	10,0	*	1,0	1,0	1,0	*	85,0	*	1,0	*
Plu	1,0	*	1,0	1,0	*	*	2,0	93,0	1,0	1,0
Ric	1,0	1,0	1,0	*	1,0	*	2,0	*	92,0	1,0
Sam	3,0	*	2,0	3,0	*	*	1,0	*	*	90,0

Wyniki klasyfikacji algorytmów CompaRe, Valsesia oraz Lukás (zbiór IMAGINE) przedstawiono w tab. 4.6–tab 4.8. Wyniki dla pozostałych algorytmów na zbiorze IMAGINE (Li, Bondi, Tuama, Mandelli, Kirchner & Johnson), jak również zbioru Dresden (wszystkie algorytmy) są przedstawione w dodatku B, odpowiednio w tab. B.7–B.11 oraz B.12–B.19 (podrozdz. B.2). Ponadto, w eksperymencie II proponowany algorytm przetestowaliśmy w dwóch podejściach. Pierwszym podejściem było nauczenie wszystkich algorytmów (Lukás—Kirchner & Johnson) za pomocą cyfrowych odcisków palca wygenerowanych algorytmem CompaRe. Drugim podejściem było przetestowanie wspomnianych algorytmów za pomocą ich własnych metod generowania cyfrowych odcisków palca w celu bezpośredniego porównania skuteczności identyfikacji z proponowanym algorytmem. Wyniki przedstawione w macierzach konfuzji reprezentują pierwsze podejście, a więc uczenie cytowanych algorytmów w oparciu o cyfrowe odciski palca wygenerowane algorytmem CompaRe (dla przejrzystości pomijamy prezentację macierzy konfuzji dla drugiego podejścia, gdyż uzyskane wyniki są porównywalne (szczegóły w tab. 4.9), a wyniki można znaleźć w treści artykułu [21]).

Procedura klasyfikacji została przeprowadzona w sposób analogiczny do procedury algorytmu Lukás. Na podstawie każdego zdjęcia  $\mathbf{I}_{(c)}^{(i)}$  z określonego aparatu cyfrowego  $c$  obliczono (korespondującą dla każdego zdjęcia) kompaktową reprezentację cyfrowego odcisku palca zgodnie z procedurą algorytmu CompaRe. Następnie obliczono ich wartość średnią  $\mathbf{K}_{(c)}$ , zgodnie ze wzorem 3.6, która jest kompaktową reprezentacją cyfrowego odcisku palca określonego aparatu. Klasyfikacja nowego zdjęcia  $\mathbf{I}^{(x)}$  odbywa się poprzez obliczenie jego kompaktowej reprezentacji cyfrowego odcisku palca  $\mathbf{K}^{(x)}$ , a następnie przez obliczenie współczynnika korelacji

$\rho$  (wzór 3.8) pomiędzy  $\mathbf{K}_{(c)}$ , a  $\mathbf{K}^{(x)}$ . Jeśli wartości  $\rho > 0,01$ , uznajemy, że zdjęcie  $\mathbf{I}^{(x)}$  zostało wykonane aparatem  $c$ ; w przeciwnym razie zakładamy, że zdjęcie  $\mathbf{I}^{(x)}$  nie zostało wykonane aparatem  $c$  (podrozdz. 4.2.3).

Jak wspomniano w rozdz. 3, wartości  $p$  algorytmu CompaRe mogą się różnić dla różnych aparatów, a jednoznaczne wskazanie konkretnej wartości  $p$  nie jest możliwe ze względu różne wymiary zdjęć generowanych przez aparaty cyfrowe. Proponujemy eksperymentalnie wybrać  $p$  tak, aby kompaktowa reprezentacja  $\mathbf{I}'$  miała wymiary  $640 \times 480$  pikseli lub – jeśli nie jest to możliwe – podobne, na przykład  $608 \times 456$ ,  $568 \times 378$ . Z naszych eksperymentów wynika, że najczęstszą wartością  $p$  dla zbioru IMAGINE jest  $p = 8$  (obrazy o wymiarach około 24 milionów pikseli), jednak dla większych obrazów (np. 45 milionów pikseli) wartości  $p$  zazwyczaj wynosiły  $p = 10$ , a także  $p = 12$ . Można zatem twierdzić, że im większe wymiary obrazu wejściowego  $\mathbf{I}$ , tym większa wartość  $p$ . W przypadku mniejszych obrazów o maksymalnych wymiarach 12 milionów pikseli (Dresden Image Database) wystarczające są wartości  $p = 4$ ,  $p = 5$ ,  $p = 6$  i  $p = 7$ . Dlatego proponujemy wybrać  $p \in [4, 12]$ , gdzie  $p \in \mathbb{N}$ . Jeśli jednak nie byłoby możliwe wybranie  $p \in [4, 12]$ , proponujemy przyjąć  $p = 2$ , ponieważ  $M$  i  $N$  powinny być zawsze parzyste. Warto wspomnieć, że im większa wartość  $p$ , tym szybciej generowana jest kompaktowa reprezentacja  $\mathbf{I}'$ , jednak wartość  $p$  nie wpływa na dokładność klasyfikacji.

Zaimplementowaliśmy algorytm Valsesia w środowisku MATLAB i zbadaliśmy, jakie powinny być wymiary losowej macierzy projekcji  $\Phi$ , aby macierz  $\mathbf{A}$  była znacznie „mniejsza” niż macierz  $\mathbf{D}$ , przy jednoczesnej zadowalającej skuteczności identyfikacji. Sprawdziliśmy najmniejsze możliwe wartości  $P$ , czyli od  $P = 1, P = 2, \dots$ , co dałoby macierz  $\mathbf{A}$  o rozmiarze  $\mathbf{A}^{1 \times M}, \mathbf{A}^{2 \times M}, \dots$ . Okazało się, że już  $P = 2$  jest wystarczające, by algorytm Valsesia identyfikował aparaty z dużą dokładnością. Jednakże algorytm ten do obliczenia macierzy losowych projekcji wymaga znacznej liczby operacji obliczeniowych. Po pierwsze, należy wygenerować macierz niezależnych zmiennych losowych o identycznym rozkładzie (ang. *independent and identically distributed random variables*) Po drugie, taką macierz należy pomnożyć przez cyfrowy odcisk palca oryginalnego aparatu. Mnożenie macierzy jest jednak zadaniem wymagającym obliczeniowo [39, 88].

Poprawność klasyfikacji ACC dla urządzeń ze zbioru IMAGINE wynosi 96,0; 93,0 i 96,0%, odpowiednio dla algorytmów CompaRe, Valsesia i Lukás. W przypadku algorytmu Li, Bondi, Tuama, Mandelli oraz Kirchner & Johnson jest to 93,0; 93,0; 92,0; 92,0 i 92,0%. Wyniki uzyskane dla zbioru Dresden wynoszą 96,0; 95,0 i 94,0%, odpowiednio dla algorytmu CompaRe, Valsesia oraz Li. Poprawność klasyfikacji (zbiór Dresden) dla algorytmów Lukás, Bondi, Tuama, Mandelli, Kirchner & Johnson wynosi odpowiednio 96,0; 95,0; 94,0; 94,0 oraz 92,0%.

Pełna poprawność klasyfikacji (tj. z wykorzystaniem wszystkich urządzeń ze zbioru Dres-

Tabela 4.6: Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm CompaRe), ACC = 96,0% (symbol \* oznacza wartości mniejsze niż 5,0%)

	Ca3	Ca4	Ca5	C12	C15	Le1	LG1	Ni3	Ni4	Ni8	N17	So3	So4
Ca3	96,0	*	*	*	*	*	*	*	*	*	*	*	*
Ca4	*	96,0	*	*	*	*	*	*	*	*	*	*	*
Ca5	*	*	96,0	*	*	*	*	*	*	*	*	*	*
C12	*	*	*	95,0	*	*	*	*	*	*	*	*	*
C15	*	*	*	*	95,0	*	*	*	*	*	*	*	*
Le1	*	*	*	*	*	96,0	*	*	*	*	*	*	*
LG1	*	*	*	*	*	*	96,0	*	*	*	*	*	*
Ni3	*	*	*	*	*	*	*	96,0	*	*	*	*	*
Ni4	*	*	*	*	*	*	*	*	96,0	*	*	*	*
Ni8	*	*	*	*	*	*	*	*	*	95,0	*	*	*
N17	*	*	*	*	*	*	*	*	*	*	96,0	*	*
So3	*	*	*	*	*	*	*	*	*	*	*	96,0	*
So4	*	*	*	*	*	*	*	*	*	*	*	*	95,0

Tabela 4.7: [Eksperyment I] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Valsesia), ACC = 93,0% (symbol \* oznacza wartości mniejsze niż 5,0%)

	Ca3	Ca4	Ca5	C12	C15	Le1	LG1	Ni3	Ni4	Ni8	N17	So3	So4
Ca3	93,0	*	*	*	*	*	*	*	*	*	*	*	*
Ca4	*	92,0	*	*	*	*	*	*	*	*	*	*	*
Ca5	*	*	92,0	*	*	*	*	*	*	*	*	*	*
C12	*	*	*	94,0	*	*	*	*	*	*	*	*	*
C15	*	*	*	*	93,0	*	*	*	*	*	*	*	*
Le1	*	*	*	*	*	92,0	*	*	*	*	*	*	*
LG1	*	*	*	*	*	*	92,0	*	*	*	*	*	*
Ni3	*	*	*	*	*	*	*	94,0	*	*	*	*	*
Ni4	*	*	*	*	*	*	*	*	93,0	*	*	*	*
Ni8	*	*	*	*	*	*	*	*	*	93,0	*	*	*
N17	*	*	*	*	*	*	*	*	*	*	94,0	*	*
So3	*	*	*	*	*	*	*	*	*	*	*	95,0	*
So4	*	*	*	*	*	*	*	*	*	*	*	*	95,0

den oraz IMAGINE) w oparciu o cyfrowe odciski palca wygenerowane algorytmem CompaRe została przedstawiona w tab. 4.9 (oznaczmy przez  $ACC^1$ ). W przypadku zbioru IMAGINE, re-

Tabela 4.8: [Eksperyment II] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Lukás), ACC = 96,0% (symbol \* oznacza wartości mniejsze niż 5,0%)

	Ca3	Ca4	Ca5	C12	C15	Le1	LG1	Ni3	Ni4	Ni8	N17	So3	So4
Ca3	96,0	*	*	*	*	*	*	*	*	*	*	*	*
Ca4	*	97,0	*	*	*	*	*	*	*	*	*	*	*
Ca5	*	*	95,0	*	*	*	*	*	*	*	*	*	*
C12	*	*	*	96,0	*	*	*	*	*	*	*	*	*
C15	*	*	*	*	95,0	*	*	*	*	*	*	*	*
Le1	*	*	*	*	*	96,0	*	*	*	*	*	*	*
LG1	*	*	*	*	*	*	97,0	*	*	*	*	*	*
Ni3	*	*	*	*	*	*	*	96,0	*	*	*	*	*
Ni4	*	*	*	*	*	*	*	*	97,0	*	*	*	*
Ni8	*	*	*	*	*	*	*	*	*	96,0	*	*	*
N17	*	*	*	*	*	*	*	*	*	*	95,0	*	*
So3	*	*	*	*	*	*	*	*	*	*	*	96,0	*
So4	*	*	*	*	*	*	*	*	*	*	*	*	96,0

zultaty są podobne do wyników zaprezentowanych w macierzach konfuzji. W przypadku zbioru IMAGINE, wyniki rozważanych algorytmów (odpowiednio CompaRe—Kirchner & Johnson) są następujące: 74,0; 94,0; 92,0; 96,0; 95,0; 94,0; 94,0 i 92,0%. Dla zbioru Dresden, dokładność ACC dla algorytmu CompaRe wynosi 96,0%; dla algorytmów Valsesia, Li, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson, odpowiednio 93,0; 91,0; 95,0; 94,0; 95,0; 93,0 i 92,0%. Tabela zawiera także rezultaty klasyfikacji w oparciu o cyfrowe odciski palca generowane za pomocą własnych procedur cytowanych algorytmów (oznaczymy przez ACC<sup>2</sup>). Wyniki te potwierdzają wysoki poziom klasyfikacji dla wszystkich rozważanych algorytmów, a nawet pokazują nieznaczną przewagę algorytmu CompaRe nad algorytmami Valsesia oraz Li.

Co ciekawe, pomimo że algorytm Lukás został zaprezentowany w 2006 roku i nie wykorzystuje technik uczenia głębokiego, osiąga porównywalną (a nawet nieznacznie wyższą) skuteczność identyfikacji sensorów w odniesieniu do metod wykorzystujących konwolucyjne sieci neuronowe.

### 4.3.3 Algorytm Vignetting-CT

Identyfikację sensorów w oparciu o algorytm Vignetting-CT przeprowadzono na zbiorach IMAGINE, Dresden Image Database oraz zbiorze smartfonów. Jakość klasyfikacji porównano z algorytmem Lukás. Rezultaty klasyfikacji dla zbioru smartfonów prezentują macierze konfuzji



Tabela 4.9: Dokładność ACC [%] obliczona na podstawie klasyfikacji wszystkich urządzeń. ACC<sup>1</sup> oznacza dokładność algorytmów uczonych cyfrowymi odciskami palca wygenerowanymi przez algorytm CompaRe; ACC<sup>2</sup> oznacza dokładność algorytmów za pomocą cyfrowych odcisków palca wygenerowanych własnymi procedurami

Algorytm	IMAGINE		Dresden	
	ACC <sup>1</sup>	ACC <sup>2</sup>	ACC <sup>1</sup>	ACC <sup>2</sup>
CompaRe	97,0	—	96,0	—
Valsesia	94,0	93,0	93,0	93,0
Li	92,0	93,0	91,0	92,0
Lukás	95,0	95,0	95,0	95,0
Bondi	95,0	95,0	95,0	95,0
Tuama	94,0	95,0	95,0	94,0
Mandelli	94,0	93,0	93,0	94,0
Kirchner	92,0	93,0	92,0	93,0

w tab. 4.10 oraz 4.11. W dodatku B (podrozdz. B.3) przedstawiono wyniki klasyfikacji dla zbiorów IMAGINE oraz Dresden w tab. B.20-B.23. Pełna poprawność klasyfikacji, obliczona na bazie eksperymentów dla wszystkich urządzeń z danego zbioru jest zawarta w tab. 4.12. Eksperymenty przeprowadzono dla  $\delta = 0,05$ .

Podobnie, jak w przypadku algorytmu MSE-DSI, klasyfikację przeprowadzono z wykorzystaniem algorytmu  $k$ -najbliższych sąsiadów (ang. *k-nearest neighbors*) z eksperymentalnie wybraną wartością  $k = 5$ . Wartości  $\hat{d}$  algorytmu Vignetting-CT zostały obliczone dla obrazów z określonego aparatu, następnie obliczono ich wartość średnią, reprezentującą dany aparat. Zdjęcia są klasyfikowane na zasadzie „wielu głosów”, tj. do klasy najczęstszej spośród  $k$  najbliższych wartości  $\hat{d}$  algorytmu Vignetting-CT. Eksperymenty przeprowadzono z 10-krotną walidacją krzyżową. Oczywiście do klasyfikacji można zastosować inny algorytm uczenia maszynowego.

Algorytm Vignetting-CT uzyskuje wyraźnie niższą poprawność klasyfikacji w odniesieniu do algorytmu Lukás na poziomie każdego z wykorzystanych zbiorów zdjęć. W przypadku zbioru smartfonów, IMAGINE oraz Dresden, proponowany algorytm uzyskuje odpowiednio: 62,0; 66,0 i 66,0%. Dla porównania, algorytm Lukás osiąga odpowiednio 84,0; 94,0 i 92,0% na tych samych zbiorach zdjęć. Podobnie, jak w przypadku algorytmów MSE-DSI oraz CompaRe, algorytm Lukás wykazuje się stabilnością klasyfikacji, a błędne rozpoznania sensorów obrazujących zdarzają się sporadycznie. Proponowany algorytm Vignetting-CT notuje wyraźnie więcej pomyłek klasyfikacyjnych, jak również jest mniej stabilny. Przykładowo, dla zbioru smartfonów klasyfikacja urządzenia LG G3 osiąga zaledwie 37,5%; smartfony Nokia Lumia 1020 oraz 1520

Tabela 4.10: Macierz konfuzji (zbiór smartfonów), identyfikacja modelu (algorytm Vignetting-CT), ACC = 82,0% (symbol \* oznacza wartości mniejsze niż 5,0%)

	Ap1	As1	HTC	Hua	LG1	LG2	Lu1	Lu2	Sa1	Sa2	So1	So2
Ap1	71,0	*	*	*	7,0	*	*	*	*	*	8,0	*
As1	*	95,0	*	*	*	*	*	*	*	*	*	*
HTC	*	*	74,0	*	*	*	*	6,0	*	*	10,0	*
Hua	*	*	*	86,0	*	*	*	6,0	*	*	*	*
LG1	*	*	*	*	90,0	*	*	*	*	*	*	*
LG2	*	*	*	*	*	95,0	*	*	*	*	*	*
Lu1	*	*	*	*	*	*	80,0	6,0	*	6,0	*	*
Lu2	*	*	*	12,0	*	*	14,0	51,0	*	*	11,0	*
Sa1	*	*	*	6,0	*	*	*	*	93,0	*	*	*
Sa2	*	*	*	*	*	*	4,5	*	13,6	76,0	*	*
So1	*	*	*	*	*	*	*	*	*	*	95,0	*
So2	*	*	*	7,0	*	9,0	*	*	*	*	*	75,0

Tabela 4.11: Macierz konfuzji (zbiór smartfonów), identyfikacja modelu (algorytm Lukás), ACC = 89,0% (symbol \* oznacza wartości mniejsze niż 5,0%)

	Ap1	As1	HTC	Hua	LG1	LG2	Lu1	Lu2	Sa1	Sa2	So1	So2
Ap1	92,0	*	*	*	*	*	*	*	*	*	*	*
As1	*	85,0	*	*	*	*	*	*	*	*	*	*
HTC	*	*	85,0	*	*	*	*	6,0	*	*	*	*
Hua	*	*	*	87,0	*	*	*	*	*	*	*	*
LG1	*	*	*	*	90,0	*	*	*	*	*	*	*
LG2	*	*	*	*	*	96,0	*	*	*	*	*	*
Lu1	*	*	*	*	*	*	86,0	*	*	*	*	*
Lu2	*	8,0	*	*	*	*	*	85,0	*	*	*	*
Sa1	*	*	*	*	*	*	*	*	92,0	*	*	*
Sa2	*	*	*	6,0	*	*	*	*	*	90,0	*	*
So1	*	*	*	*	*	*	*	*	*	*	93,0	*
So2	*	*	*	*	*	6,0	*	*	*	*	*	91,0

zostały sklasyfikowane z dokładnością 50,0%. Natomiast w przypadku urządzeń Asus ZenFone 2, Samsung Galaxy S6 oraz Sony Xperia Z3 dokładność wynosi, odpowiednio 75,0; 72,7 i 71,4%. Niemniej, ogólna klasyfikacja oscyluje wokół poziomu około 65,0% (dla wszystkich zbiorów), co jest wyższym wskaźnikiem odnosząc się do modelu losowego. Klasyfikacja w odniesieniu do

zbiorów IMAGINE oraz Dresden przedstawia się na nieco wyższym poziomie, a co ważniejsze, błędy klasyfikacji wydają się być bardziej „przewidywalne”. Odsetek błędnie sklasyfikowanych urządzeń na ogół nie przekracza poziomu 12%.

Co ciekawe, oba algorytmy wykazują najwięcej pomyłek w klasyfikacji w przypadku zbioru smartfonów. Najbardziej prawdopodobnym wytłumaczeniem takiego zjawiska jest fakt, iż nowoczesne smartfony intensywnie ingerują w finalny wygląd zdjęcia, stosując różne filtry kolorystyczne. Jak wspomniano w [77], takie właśnie podejście może obniżać dokładność klasyfikacji.

Tabela 4.12: Dokładność ACC [%] obliczona na podstawie klasyfikacji wszystkich urządzeń

Algorytm	IMAGINE	Dresden
	ACC	ACC
Vignetting-CT	66,0	66,0
Lukás	94,0	92,0

#### 4.3.4 Algorytm Distortion-CT

Przeanalizowaliśmy parametr dystorsji obiektywu  $k$  dla obrazów przedstawiających ten sam kadr, ale z różnych urządzeń. Parametr  $k$  wyznaczyliśmy korzystając z oprogramowania Hugin Photo Stitcher [5]. Analiza eksperymentalna pokazuje, że pomimo fotografowania tej samej sceny, parametr dystorsji  $k$  osiąga różne wartości. Warto zauważyć, że nawet różne egzemplarze tego samego modelu aparatu generują różne wartości parametru  $k$ . Taki przykład przedstawia rys. 4.1, dla którego parametr  $k$  dla tego samego kadru z dwóch różnych smartfonów (Huawei P8 i Samsung Galaxy S6) osiąga różne wartości. Podobna sytuacja jest przedstawiona na rys. 4.2, dla dwóch różnych urządzeń Nikon CoolPix S710 (zbiór Dresden Image Database). W tab. 4.13 oraz 4.14 zebrano przykładowe wartości parametru dystorsji  $k$ . Ze względu na dużą liczbę urządzeń i obrazów, przedstawiamy w tabelach dla przejrzystości tylko część pełnych wyników. Parametr  $k$  został obliczony dla co najmniej 40 pikseli z każdego obrazu. W przypadku zdjęć, na których nie wykryto dystorsji, analizę pominięto.

Uzyskane wyniki wskazują, że pomimo fotografowania tych samych kadrów, parametr  $k$  osiąga różne wartości. Widać to także w przypadku różnych egzemplarzy tego samego modelu aparatu. Przykładowo, dla zdjęcia 14.jpg z urządzeń Huawei P8 oraz Samsung Galaxy S6, parametr  $k$  osiąga odpowiednio: 0,001 oraz 0,028 (zbiór smartfonów). Dla zdjęcia 92.jpg dla aparatów Canon IXUS 70 (1) i (2), wartości  $k$  osiągają odpowiednio 0,501 oraz 0,222; w przypadku aparatu Nikon D70 (1) i (2),  $k$  jest równe 0,91 i 0,68 (baza Dresden).



Rysunek 4.1: Dystorsja na podstawie zdjęć tego samego kadru z różnych urządzeń (przykładowe linie obarczone dystorsją oznaczono kolorem czerwonym). Zdjęcie po lewej: Huawei P8, parametr dystorsji  $k = -0,6782$ ; zdjęcie po prawej: Samsung Galaxy S6, parametr dystorsji  $k = -0,2$



Rysunek 4.2: Dystorsja na podstawie zdjęć tego samego kadru z różnych urządzeń (przykładowe linie obarczone dystorsją oznaczono kolorem czerwonym). Zdjęcie po lewej: Nikon CoolPix S710 (2), parametr dystorsji  $k = -0,10323$ ; zdjęcie po prawej: Nikon CoolPix S710 (1), parametr dystorsji  $k = -0,01799$

Proponowane podejście dostarcza informacji, czy dany zestaw zdjęć został wykonany przez jeden lub więcej aparatów, jednak trudno oczekiwać identyfikacji, który model lub (uogólniając) marka została użyta. Zaletą proponowanej metody jest jednak jej szybkość, ponieważ parametr  $k$  można obliczyć w czasie rzeczywistym. Co więcej, parametr  $k$  można łatwo wyznaczyć za pomocą edytorów obrazów. Proponowana metoda może być przydatna w prostych przypadkach porównywania podobnych zdjęć, jednak może okazać się niepraktyczna dla zestawów różnych zdjęć. Jednym z powodów jest fakt, że parametr dystorsji  $k$  zmienia się wraz z odległością od obiektu, polem widzenia (ang. *field of view*) i ogniskową obiektywu (ang. *focal length*). Nie zmienia to jednak faktu, że rozważana metoda może być przydatna do szybkiego określenia,

Tabela 4.13: Wartości parametru dystorsji  $k$  na podstawie zdjęć tego samego kadru (zbiór smartfonów, zdjęcia wykonane na ekwiwalencie ogniskowej 28mm)

Zdjęcie	Huawei P8	Samsung Galaxy S6
11.jpg	0,266	0,000
12.jpg	-0,027	-0,060
13.jpg	0,026	0,000
14.jpg	0,001	0,028
15.jpg	0,000	-0,002
16.jpg	0,008	0,032
17.jpg	0,000	0,006
18.jpg	-0,027	-0,042
19.jpg	-0,027	-0,060
20.jpg	-0,001	-0,017
21.jpg	-0,187	-0,060
22.jpg	-0,529	-0,012
23.jpg	-0,027	-0,060
24.jpg	0,008	-0,060

czy określony zbiór zdjęć został wykonany jednym, bądź większą liczbą urządzeń.

## 4.4 Czas przetwarzania obrazów

W niniejszym podrozdziale porównamy czas, jaki jest potrzebny na uczenie rozważanych algorytmów. Przedstawimy rezultaty uzyskane w oparciu o proponowane algorytmy, w odniesieniu do algorytmów z literatury. W każdym porównaniu algorytmy przebadamy na tych samych zestawach zdjęć.

### 4.4.1 Algorytm MSE-DSI

**Identyfikacja aparatów cyfrowych** Przetestowaliśmy, jaki czas jest potrzebny, by nauczyć algorytm MSE-DSI oraz algorytmy i metody zaproponowane przez Lukás, Bondi oraz Tuama. Eksperymenty zostały przeprowadzone z wykorzystaniem komputera 2 na dokładnie tych samych zestawach zdjęć.

W tab. 4.15 przedstawiono zestawienie czasu potrzebnego do nauczania testowanych algorytmów na podstawie 1919 zdjęć (zbiór IMAGINE) oraz 11 787 zdjęć ze zbioru Dresden Image

Tabela 4.14: Wartości parametru dystorsji  $k$  na podstawie zdjęć tego samego kadru (zbiór Dresden). Wykorzystano trzy egzemplarze aparatu Canon IXUS 70 oraz dwa egzemplarze Nikon D70 (zdjęcia wykonane na ekwiwalencie ogniskowej 27mm)

Zdjęcie	Canon IXUS 70			Nikon D70	
	(1)	(2)	(3)	(1)	(2)
85.jpg	-0,002	-0,001	-0,012	-0,150	-0,230
86.jpg	-0,006	-0,034	-0,091	0,010	-0,070
87.jpg	-0,590	-0,119	0,001	-0,230	-0,361
88.jpg	0,170	0,420	0,207	-0,003	0,001
89.jpg	0,053	0,020	0,081	0,190	0,019
90.jpg	-0,050	-0,061	-0,018	-0,024	-0,039
91.jpg	0,634	0,520	0,784	0,499	0,506
92.jpg	0,501	0,222	0,360	0,910	0,680
93.jpg	-0,005	-0,040	-0,081	-0,200	-0,090
94.jpg	-0,620	-0,890	-0,700	-0,023	-0,601
95.jpg	-0,940	-0,880	-0,950	-0,550	-0,598
96.jpg	0,040	0,077	0,901	-0,004	-0,020
97.jpg	-0,001	0,004	-0,095	0,170	0,050
98.jpg	-0,062	-0,181	-0,560	-0,340	-0,210

Database. Rys. 4.3 oraz 4.4 przedstawiają graficzną interpretację rozważanej sytuacji.

Eksperymenty wykazały, że zarówno algorytm Lukás, jak i metody oparte na CNN przetwarzają obrazy znacznie dłużej niż proponowany algorytm MSE-DSI. Obliczenie pojedynczej wartości MSE algorytmu MSE-DSI trwa około 0,03 minuty (ok. 2 sekundy) zarówno dla zbioru IMAGINE jak i Dresden; wygenerowanie pojedynczego cyfrowego odcisku palca za pomocą algorytmu Lukás wynosi 2 minuty dla zbioru IMAGINE oraz około 1,5 minuty na bazie Dresden. Czas na naukę jednej epoki z rozważanymi sieciami konwolucyjnymi w przypadku zbioru IMAGINE to aż 65 godzin dla sieci Bondi i 18 godzin dla sieci Tuama; dla bazy Dresden jest to odpowiednio 41 i 16 godzin (Bondi, Tuama). Sieci wymagają co najmniej 18 i 25 epok (odpowiednio zbiór IMAGINE oraz Dresden) na nauczenie zapewniające poprawność klasyfikacji prezentowaną w macierzach konfuzji. Pomimo na ogół podobnych struktur sieci Bondi oraz Tuama, uczenie metodą Bondi trwa nieakceptowalnie długo. Trudno jednak jednoznacznie wyjaśnić takie zjawisko.

Podsumowując, obliczenie wartości MSE dla 1919 testowanych obrazów ze zbioru IMAGINE zajęło nieco ponad jedną godzinę, dla bazy Dresden jest to około 5 godzin (295 minut).

Tabela 4.15: Czas przetwarzania zdjęć

Zbiór IMAGINE		
Algorytm	Średnio (1 zdjęcie) [min]	Ogółem [min]
MSE-DSI	0,03	63,97
Lukás	2,00	3838,00
Średnio (1 epoka) [h]		Ogółem [h]
Bondi	65,00	—
Tuama	18,00	324,00
Zbiór Dresden		
Algorytm	Średnio (1 zdjęcie) [min]	Ogółem [min]
MSE-DSI	0,03	294,68
Lukás	1,58	18662,75
Średnio (1 epoka) [h]		Ogółem [h]
Bondi	41,00	—
Tuama	16,00	400,00

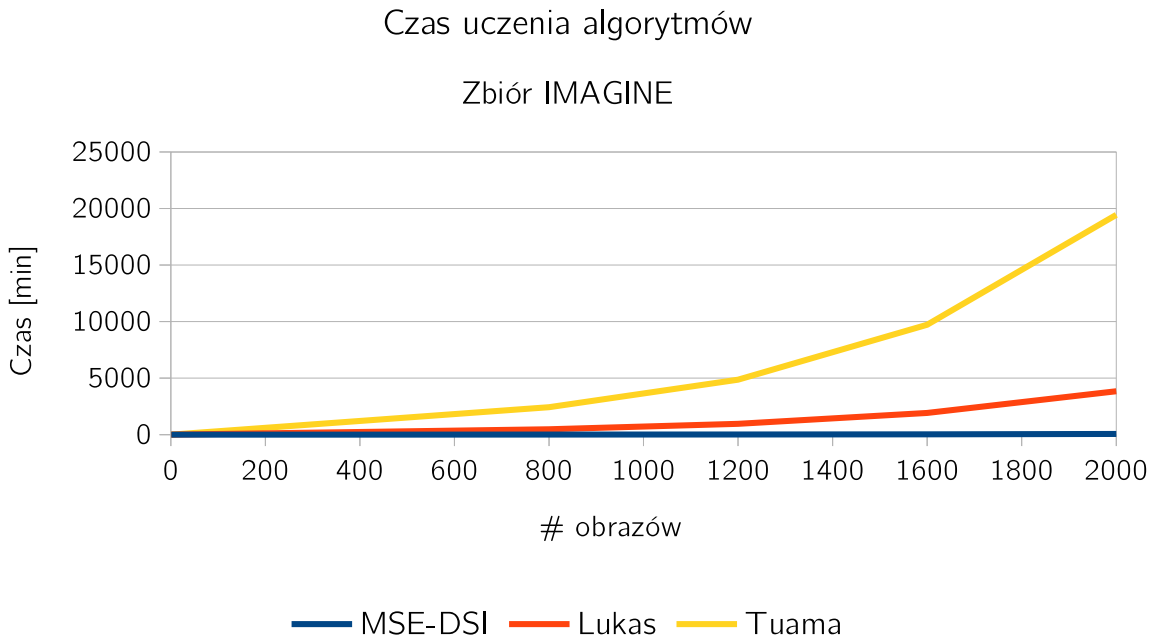
Algorytm Lukás wymaga około 64 godzin (3838 minut) na wygenerowanie cyfrowych odcisków palca ze zbioru IMAGINE oraz ponad 300 godzin (około 18 663 minut) dla bazy Dresden. Uczenie sieci Tuama wymaga 324 i 400 godzin, odpowiednio na zbiorze IMAGINE oraz Dresden. Uczenie sieci Bondi nie jest możliwe do przeprowadzenia w warunkach „domowych”; jest ono jednak możliwe do realizacji na urządzeniu klasy np. nVidia DGX-2.

**Identyfikacja skanerów płaskich** W tab. 4.16 oraz na rys. 4.5 przedstawiono średnie i całkowite czasy obliczenia cyfrowych odcisków palca dla algorytmów MSE-DSI oraz Khanna na bazie skanów. Eksperymenty zostały przeprowadzone na komputerze 1.

Tabela 4.16: Czas przetwarzania (w sekundach) 290 skanów

Algorytm	Średnio (1 zdjęcie) [s]	Ogółem [s]
MSE-DSI	2,0	580,0
Khanna	120,0	34800,0

Analogicznie do przypadku identyfikacji aparatów cyfrowych, eksperymenty wykazały, że nauczanie algorytmu MSE-DSI zajmuje mniej czasu niż w przypadku algorytmu Khanna. Średnio jeden obraz jest przetwarzany przez MSE-DSI około 2 sekundy, stąd obliczenie wartości MSE dla 290 obrazów zajmuje około 10 minut (580 sekund). Algorytm Khanna uczy się jednego



Rysunek 4.3: Porównanie czasu uczenia algorytmów, identyfikacja aparatów cyfrowych (zbiór IMAGINE)

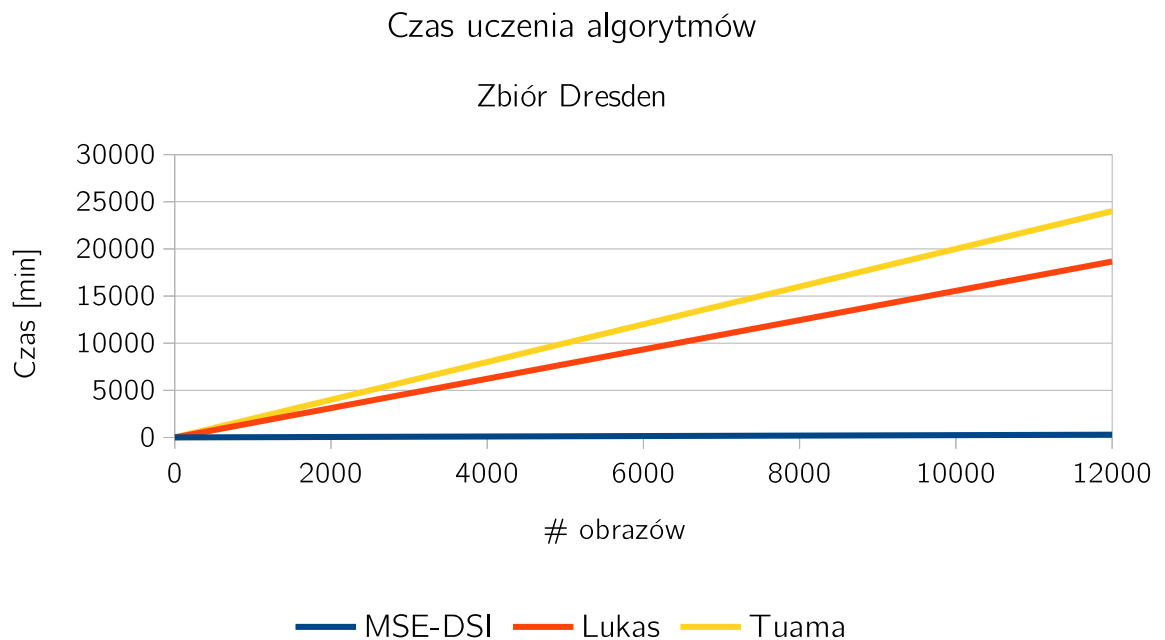
obrazu przez około 2 minuty, co daje łączny czas na poziomie około 10 godzin (ok. 35 000 sekund).

**Podsumowanie** Głównym powodem dłuższego przetwarzania obrazów przez algorytmy literaturowe jest odsumowanie wszystkich trzech kanałów kolorystycznych zdjęć wejściowych. W proponowanej metodzie przetwarzamy jedynie fragmenty obrazu o wymiarach  $512 \times 512$  pikseli, a ponadto, przetwarzamy tylko jeden kanał kolorystyczny. Zestawiając wyniki czasowe z dokładnością klasyfikacji, rezultaty uzyskane przez algorytm MSE-DSI można uznać za satysfakcjonujące, gdyż algorytm oferuje nieznacznie niższą skuteczność klasyfikacji przy wyraźnie krótszym czasie działania.

#### 4.4.2 Algorytm CompaRe

W niniejszym podrozdziale przedyskutujemy efektywność algorytmu CompaRe w porównaniu do algorytmów z literatury pod względem czasu uczenia algorytmów. Przeprowadzono dwa



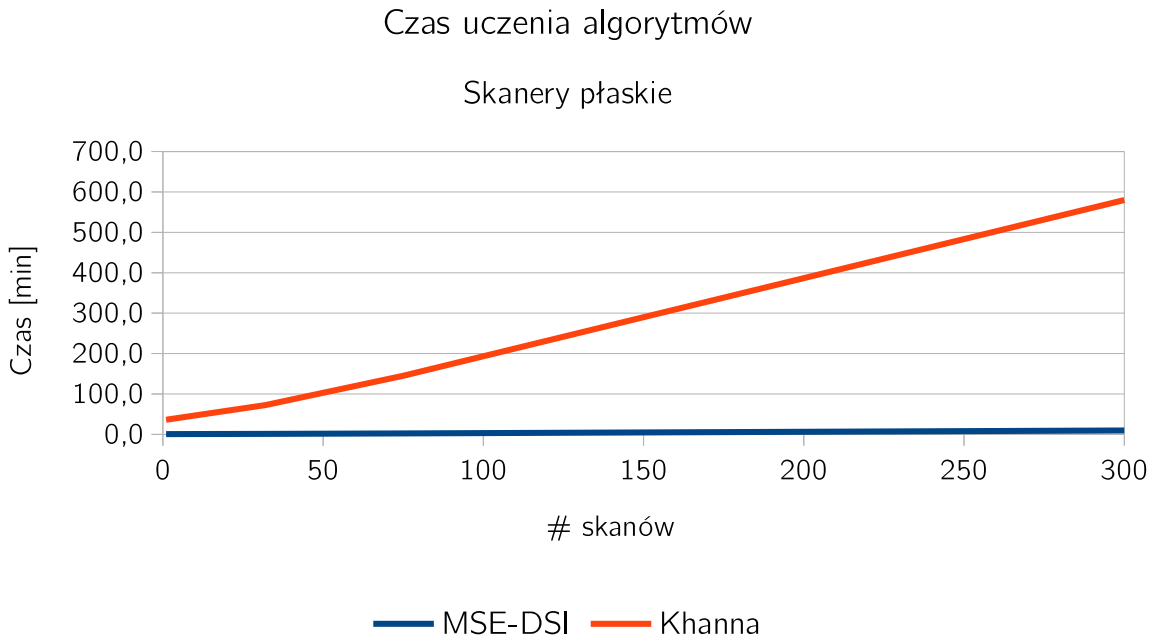


Rysunek 4.4: Porównanie czasu uczenia algorytmów, identyfikacja aparatów cyfrowych (zbiór Dresden Image Database)

eksperymenty z użyciem komputera 2. W ramach pierwszego eksperymentu porównano czas potrzebny na wygenerowanie cyfrowych odcisków palca (porównanie z algorytmami Valsesia oraz Li); w drugim przeanalizowano czas uczenia w odniesieniu do algorytmów: Lukás, Bondi, Tuama, Mandeli oraz Kirchner & Johnson. Wyniki przeanalizowano łącznie dla około 12 000 zdjęć (zbiory IMAGINE oraz Dresden).

**Eksperyment I** Porównanie czasu generowania cyfrowego odcisku palca dla jednego zdjęcia algorytmu CompaRe z metodami Valsesia oraz Li jest przedstawione jako tab. 4.17.

Algorytm CompaRe generuje kompaktową reprezentację cyfrowego odcisku palca w najkrótszym czasie, porównując z algorytmami Valsesia oraz Li. Dla zbiorów (odpowiednio) IMAGINE oraz Dresden, średni czas wygenerowania jednego odcisku palca to około 45 i 35 sekund dla algorytmu CompaRe, 140 i 115 sekund dla algorytmu Valsesia oraz 56 i 48 sekund dla algorytmu Li. Łączny czas na obliczenie cyfrowych odcisków palca (uśredniając dla obu zbiorów zdjęć – łącznie blisko 14 000 zdjęć) to w przypadku algorytmu CompaRe 152 godziny, Valsesia – 485 godzin oraz 198 godzin dla algorytmu Li (rys. 4.6). Najwolniejszy okazał się algorytm

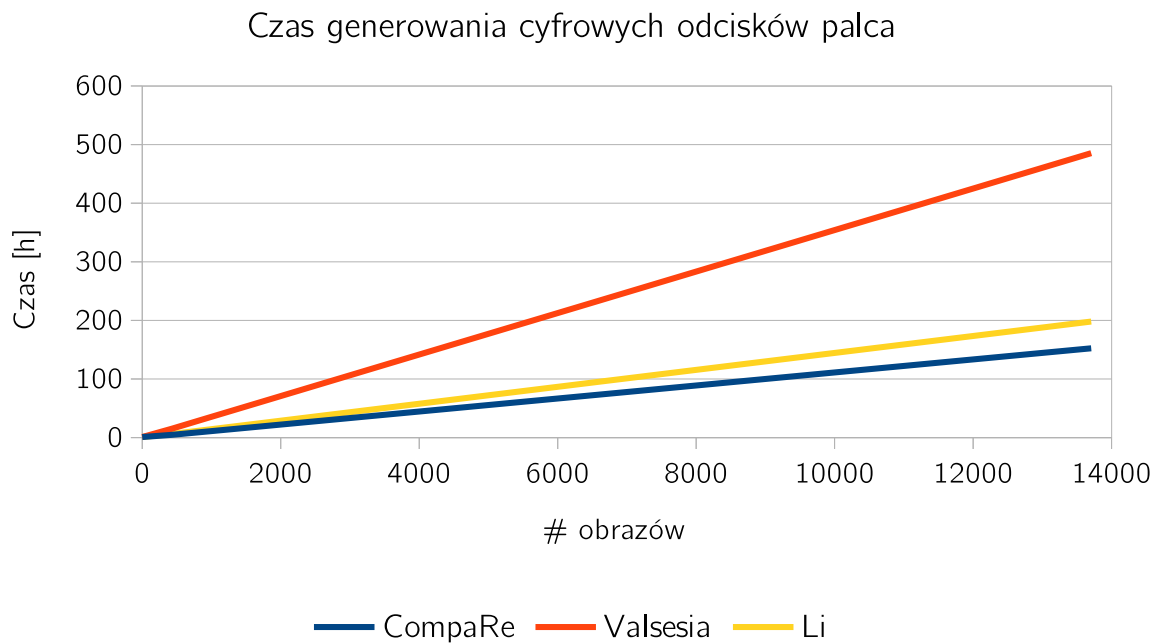


Rysunek 4.5: Porównanie czasu uczenia algorytmów, identyfikacja skanerów płaskich

Tabela 4.17: Średni czas (w sekundach) potrzebny na wygenerowanie pojedynczego cyfrowego odcisku palca (zbiory IMAGINE i Dresden)

	IMAGINE Czas [s]	Dresden Czas [s]
CompaRe	45,0	35,0
Valsesia	140,0	115,0
Li	56,0	48,0

Valsesia. Wynik ten był jednak spodziewany, ponieważ metoda wymaga wygenerowania losowej macierzy projekcji i przemnożenia jej z obrazem wejściowym o dużych wymiarach, co jest kosztowne obliczeniowo. Wprawdzie algorytm ten można przyspieszyć poprzez zaimplementowanie mnożenia macierzy za pomocą dyskretnej transformaty Fouriera, która obniża złożoność do liniowej klasy logarytmicznej, jednak nadal konieczne jest generowanie macierzy losowych projekcji. W przypadku bazy Dresden Image Database wyniki są analogiczne, chociaż wartości czasowe są nieco niższe. Wynika to z tego, że obrazy wchodzące w skład bazy Dresden mają mniejsze wymiary, dzięki czemu ich przetwarzanie odbywa się w krótszym czasie.

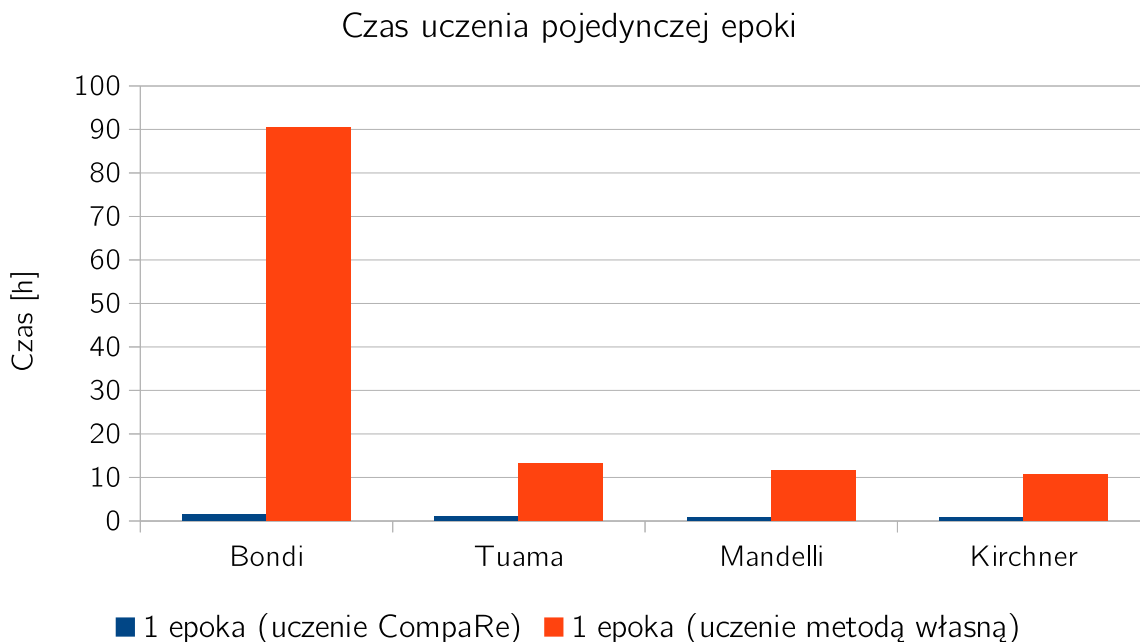


Rysunek 4.6: Czas potrzebny na wygenerowanie cyfrowych odcisków palca

**Eksperyment II** W ramach drugiego eksperymentu porównujemy czasy uczenia algorytmu Lukás oraz sieci konwolucyjnych Bondi, Tuama, Mandelli oraz Kirchner & Johnson. Czasy uczenia pojedynczej epoki dla sieci konwolucyjnych są przedstawione w tab. 4.18 dla obu zbiorów zdjęć. Na rys. 4.7 przedstawiono graficzną interpretację opisywanej sytuacji (w uśrednieniu dla obu zbiorów zdjęć). Czas uczenia algorytmu Lukás został przedstawiony na rys. 4.8. Zauważmy, że porównujemy zarówno czas uczenia danej metody za pomocą cyfrowych odcisków palca wygenerowanych metodą CompaRe, jak i za pomocą własnych metod, które stosują rozważane sieci.

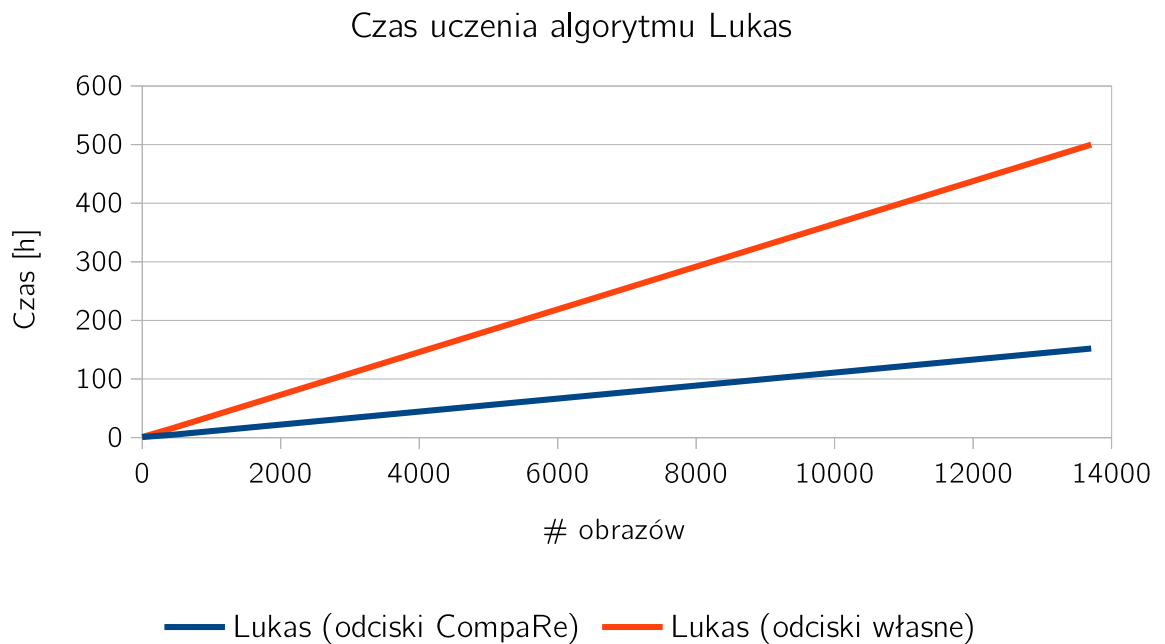
Tabela 4.18: Czas (w godzinach) potrzebny na nauczenie pojedynczej epoki

		Bondi	Tuama	Mandelli	Kirchner
IMAGINE	1 epoka (CompaRe) [h]	1,5	1,13	0,9	0,9
	1 epoka (własna) [h]	96,0	14,4	12,8	11,0
Dresden	1 epoka (CompaRe) [h]	1,4	1,1	0,8	0,8
	1 epoka (własna) [h]	85,0	12,1	10,2	10,3



Rysunek 4.7: Czas potrzebny na uczenie sieci konwolucyjnych dla pojedynczej epoki (wyniki uśrednione dla zbiorów IMAGINE oraz Dresden)

W przypadku metod opartych na CNN liczba epok koniecznych do uzyskania wyników klasyfikacji na takim poziomie, jak przedstawiono w macierzach konfuzji, wynosiła co najmniej 18 (dla wszystkich metod). Rezultaty przedstawione na rys. 4.7 wyraźnie wskazują, że niezależnie od zastosowanego algorytmu, uczenie cyfrowymi odciskami palców generowanymi przez algorytm CompaRe jest znacznie szybsze niż w przypadku cyfrowych odcisków generowanych przez rozważane algorytmy. Uczenie sieci konwolucyjnej Tuama wymaga odpowiednio ponad 14 i 12 godzin dla zbiorów IMAGINE oraz Dresden. Ponownie najdłuższy czas uczenia został zaobserwowany w przypadku sieci konwolucyjnej Bondi, gdzie czas nauki jednej epoki osiągnął 96 godzin dla zbioru IMAGINE i 85 godzin dla zbioru Dresden. Wyniki metod Mandelli oraz Kirchner & Johnson są zbliżone do metody Tuama i wynoszą, odpowiednio (około) 13 i 10 godzin oraz 11 i 10 godzin (odpowiednio zbiory IMAGINE i Dresden). Uczenie wymienionych metod za pomocą cyfrowych odcisków palca wygenerowanych algorytmem CompaRe znacznie skraca czas nauki, który waha się w okolicach jednej godziny na epokę (oba zbiory zdjęć), wyjątkiem jest jednak sieć Bondi, gdzie czas uczenia wynosi 1,5 godziny. Czynnikiem, który spowodował poprawę wydajności w porównaniu do wspomnianych metod, jest przetwarzanie



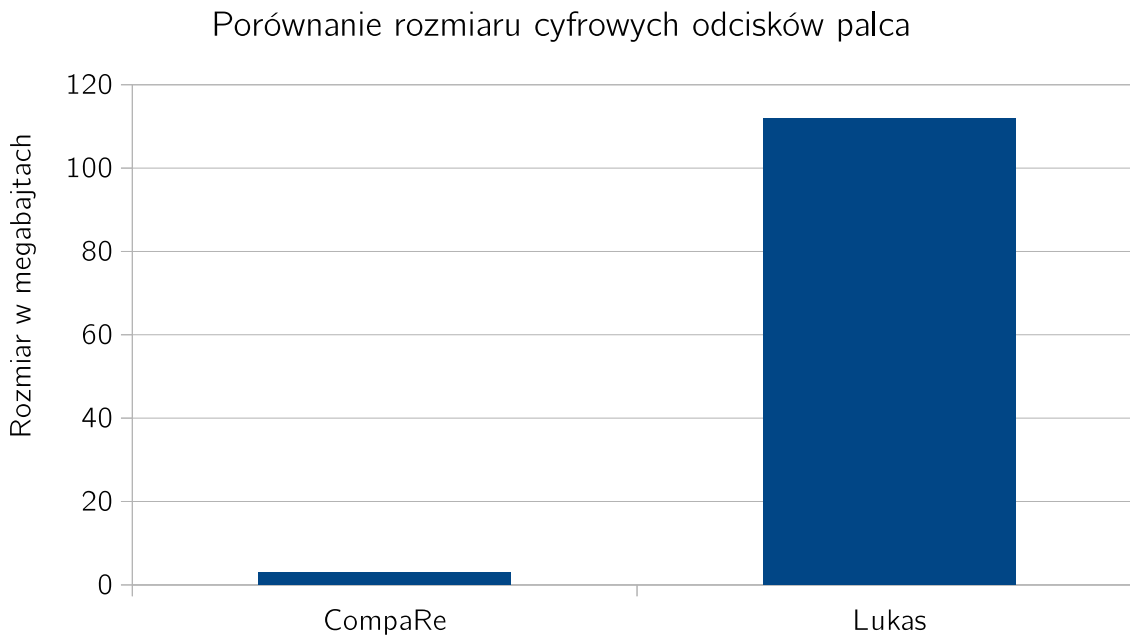
Rysunek 4.8: Czas potrzebny na uczenie algorytmu Lukás (wyniki uśrednione dla obu zbiorów zdjęć)

macierzy o mniejszych wymiarach.

Algorytm Lukás także osiągnął zauważalnie dłuższe czasy przetwarzania obrazów. Jak wskazuje rys. 4.8, nauczenie tego algorytmu zdjęciami z obu zbiorów zdjęć zajęło około 500 godzin (własna metoda generowania cyfrowych odcisków palca), podczas gdy uczenie tego algorytmu za pomocą cyfrowych odcisków palca wygenerowanych algorytmem CompaRe zajmuje około 152 godziny.

Można także porównać fizyczny rozmiar plików cyfrowych odcisków palca. W przypadku algorytmu Lukás, pliki są reprezentowane z rozszerzeniem \*.mat (dostarczona przez autorów implementacja w środowisku MATLAB [105]). Intuicyjnie, im większe są wymiary obrazu wejściowego, tym większy jest cyfrowy odcisk palca. Algorytm CompaRe także zaimplementowano w oprogramowaniu MATLAB, a średni rozmiar plików to około 3,0 megabajtów, podczas gdy dla algorytmu Lukás jest to około 112,0 megabajtów (oba zbiory zdjęć). Podkreślmy, że niektóre z cytowanych sieci konwolucyjnych używają cyfrowych odcisków palca wygenerowanych w dokładnie ten sam sposób, jak algorytm Lukás (należy jedynie przekonwertować pliki \*.mat na \*.jpg, by podać je na wejście CNN). Zaletą mniejszych rozmiarów cyfrowych odcisków

palca może być praktyczne zastosowanie na przykład w kryminalistycznych centrach danych (ang. *forensic centres*), które przechowują duże ilości takich materiałów. Porównanie rozmiarów przedstawiono na rys. 4.9. Warto podkreślić, że cyfrowe odciski palca obliczone dla wszystkich wykorzystywanych w niniejszej rozprawie zdjęć (około 14 000), w rozumieniu algorytmu Lukás zajmują około 1,2 terabajta. Ich odpowiedniki wygenerowane algorytmem CompaRe wymagają jedynie około 65 gigabajtów. Różnica jest więc bardzo wyraźna.



Rysunek 4.9: Porównanie rozmiarów cyfrowych odcisków palca zapisanych jako pliki \*.mat

**Podsumowanie** Eksperymentalna analiza pokazała przewagę algorytmu CompaRe nad algorytmami z literatury pod względem szybkości przetwarzania obrazów. Zarówno w przypadku generowania cyfrowych odcisków palca (porównanie z algorytmami Valsesia oraz Li) oraz dla „standardowych” algorytmów i metod (Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson), proponowany algorytm uzyskuje krótsze czasy przetwarzania obrazów. Kompaktowa reprezentacja cyfrowego odcisku palca jest również korzystna dla konwolucyjnych sieci neuronowych. Jeśli obrazy wejściowe mają wymiary, na przykład  $7952 \times 5304$ , a odpowiadający im cyfrowy odcisk palca ma te same wymiary, nie jest możliwe podanie na wejście CNN tak dużego obrazu. Wymusza to wtedy redukcję wymiarów obrazu, aby możliwe było jego przetwarzanie. Cyfrowe

odciski palców generowane przez algorytm CompaRe o wymiarach zbliżonych do  $640 \times 480$  pikseli mogą być stosowane w sieciach CNN bez konieczności takiego przetwarzania (oczywiście pod warunkiem posiadania karty graficznej o odpowiedniej ilości pamięci wideo, co jednak może być utrudnione w przypadku typowego komputera klasy PC). Posiadanie cyfrowych odcisków palca w kompaktowej reprezentacji ułatwia też ich przechowywanie na przestrzeni dyskowej ze względu na zauważalnie mniejsze rozmiary plików.

### 4.4.3 Algorytm Vignetting-CT

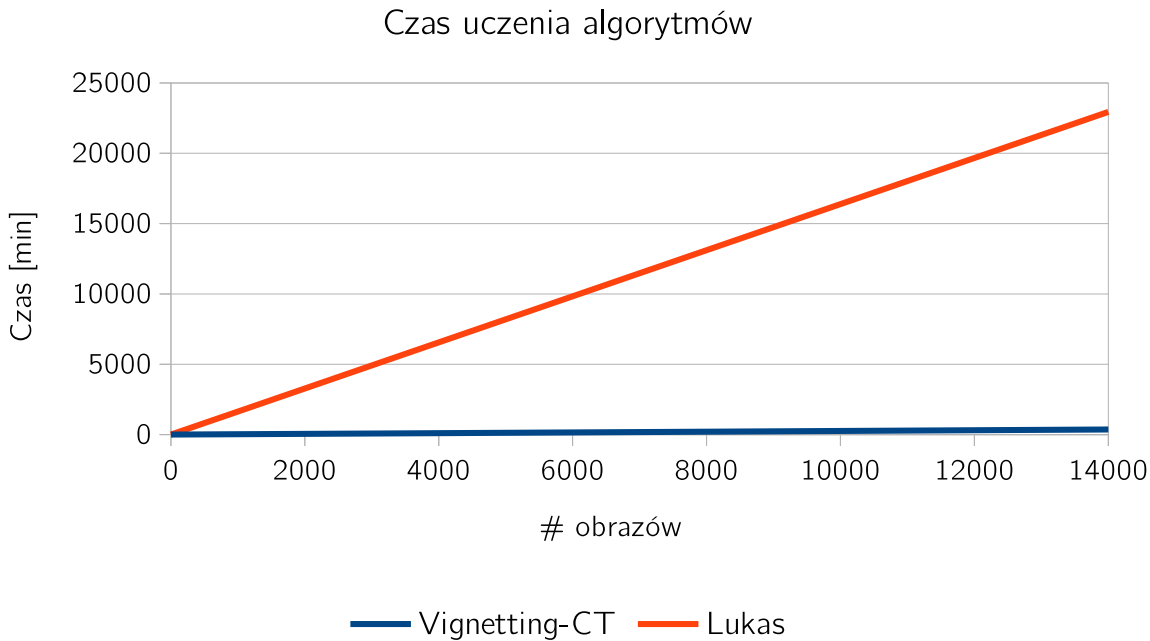
Eksperymenty zostały przeprowadzone z wykorzystaniem komputera 1. Dokładne rezultaty (dla wszystkich zbiorów zdjęć) zostały przedstawione w tab. 4.19, na rys. 4.10 przedstawiono graficzną interpretację opisywanej sytuacji w uśrednieniu dla wszystkich zbiorów.

Tabela 4.19: Czas przetwarzania zdjęć (w minutach)

	Algorytm	Średnio (1 zdjęcie) [min]	Ogółem [min]
IMAGINE	Vignetting-CT	0,03	64,00
	Lukás	2,00	3838,00
Dresden	Vignetting-CT	0,03	295,00
	Lukás	1,58	18663,00
Zb. smartfonów	Vignetting-CT	0,03	7,00
	Lukás	1,67	440,00

Wyniki pokazują, że algorytm Lukás przetwarza obrazy znacznie dłużej w porównaniu do algorytmu Vignetting-CT. Proponowana metoda uczy się obrazów niemal w czasie rzeczywistym (około 0,03 minuty, a więc ok. 2 sekundy na obraz), podczas gdy Lukás potrzebuje średnio około 1,5-2 minut na przetworzenie pojedynczego obrazu (zależnie od zbioru zdjęć). Oczywiście czas ten jest zależny od wymiarów obrazu. Obrazy o mniejszych wymiarach (np. 6 milionów pikseli –  $3000 \times 2000$  pikseli z bazy Dresden) są przetwarzane w krótszym czasie, natomiast duże obrazy ze zbioru IMAGINE o rozdzielczości 24 i więcej milionów pikseli (np.  $6000 \times 4000$  pikseli) są przetwarzane wyraźnie dłużej. W przypadku algorytmu Vignetting-CT nie ma większego znaczenia, z którego zbioru zdjęć pochodzi analizowane zdjęcie. Obliczenie wartości  $\hat{d}$  dla  $\delta = 0,05$  odbywa się niemal natychmiastowo i nigdy nie zajmuje więcej niż 2 sekundy. Wynika to z przetwarzania niewielkich fragmentów obrazu.

Podsumowując, czas przetwarzania blisko 14 000 obrazów (zbiory IMAGINE, Dresden oraz zbiór smartfonów) wyniósł nieco ponad 6 godzin (około 366 minut) w przypadku algorytmu Vignetting-CT i około 383 godziny (22 941 minut) dla algorytmu Lukás. Taka wydajność



Rysunek 4.10: Porównanie czasów uczenia algorytmów

czasowa może wykluczyć użycie algorytmu Lukás na masową skalę. Przypomnijmy jednak, że jak wspomnieliśmy w rozdz. 3, algorytm Vignetting-CT nie nadaje się do tak rzetelnej identyfikacji aparatów, jak algorytm Lukás; może on jednak posłużyć do wstępnego przetworzenia zbioru danych w celu ustalenia, czy określony zestaw zdjęć pochodzi z jednego bądź wielu aparatów cyfrowych.

#### 4.4.4 Algorytm Distortion-CT

Obliczenie parametru  $k$  sprowadza się do rozwiązania równania 3.11 i nie stanowi wymagającego zadania obliczeniowego nawet w przypadku dużej liczby pikseli obarczonych dystorsją. Porównanie czasów przetwarzania obrazów znajduje się w tab. 4.20.

Obliczenie parametru  $k$  nawet dla około 10 000 pikseli obarczonych dystorsją w środowisku MATLAB odbywa się w czasie rzeczywistym (czas poniżej 0,5 sekundy). Oznacza to, że łączne przetworzenie wszystkich zdjęć z trzech zbiorów danych zajmuje niecałe 25 minut. Wyniki dla algorytmu Lukás są identyczne, jak w przypadku porównania z algorytmem Vignetting-CT.



Tabela 4.20: Czas przetwarzania zdjęć (w minutach)

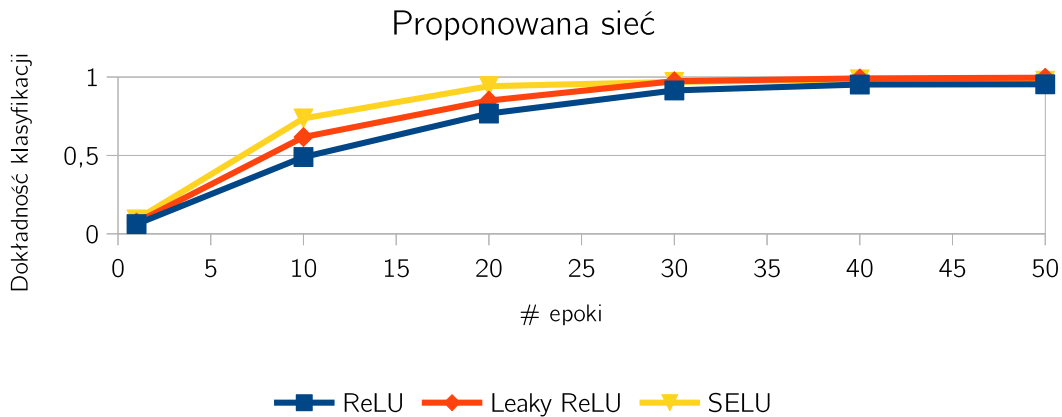
	Algorytm	Średnio (1 zdjęcie) [min]	Ogółem [min]
IMAGINE	Distortion-CT	0,002	3,00
	Lukás	2,00	3838,00
Dresden	Distortion-CT	0,002	20,00
	Lukás	1,58	18663,00
Zb. smartfonów	Distortion-CT	0,002	0,00
	Lukás	1,67	440,00

#### 4.4.5 Identyfikacja w oparciu o konwolucyjne sieci neuronowe

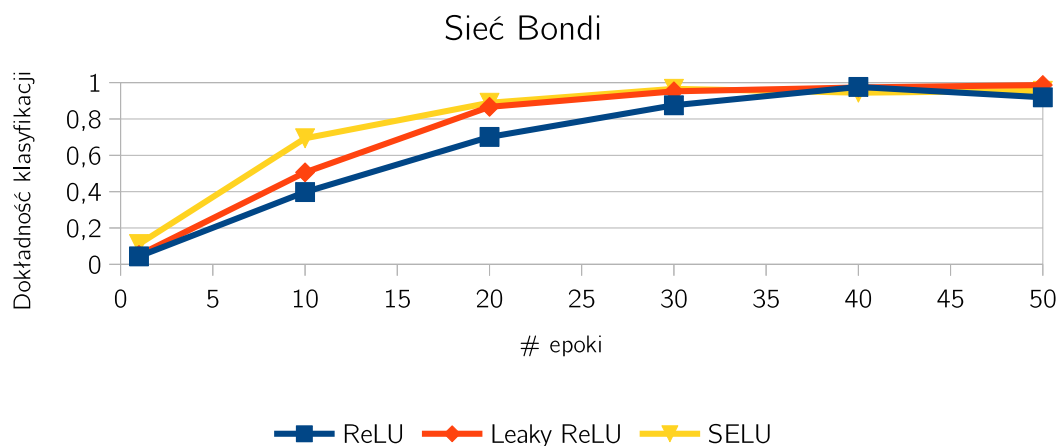
Zbadaliśmy również możliwość „przyspieszenia” uczenia sieci konwolucyjnych, służących do rozpoznawania sensorów na podstawie zdjęć. Eksperymenty zostały przeprowadzone na komputerze 2. Inspiracją była praca przedstawiona przez Obregon [55], w której rozważano analizę różnych parametrów sieci konwolucyjnych i ich wpływ na identyfikację. Nasze wyniki odniesiemy do sieci zaproponowanych przez Bondi oraz Tuama.

Przeanalizowaliśmy wpływ różnych funkcji aktywacji na dokładność treningu (ang. *train accuracy*) w sieciach konwolucyjnych. W tym celu porównaliśmy czas potrzebny na takie nauczanie sieci CNN, by możliwa była identyfikacja aparatów na poziomie, odpowiednio 80,0 i 90,0%. Zauważmy, że duża liczba sieci konwolucyjnych wykorzystuje ReLU jako funkcję aktywacji poszczególnych warstw splotowych [26, 145]. Zamiast ReLU, proponujemy użycie funkcji Leaky ReLU oraz SELU (funkcje omówione w rozdz. 3). Wyniki wskazują, że użycie funkcji aktywacji SELU pozwala na znacznie szybszy trening sieci w porównaniu do funkcji ReLU oraz Leaky ReLU. Uczenie już po 15 epokach dla wszystkich rozważanych sieci zapewnia 80,0% dokładności treningu, podczas gdy uczenie przez 20 epok osiąga już 90,0% dokładności dla funkcji SELU. Funkcja Leaky ReLU potrzebuje co najmniej 10 epok więcej na uzyskanie takich samych wyników, podczas gdy ReLU osiąga najniższe wyniki, wymagając nawet dwa razy więcej epok niż funkcja SELU. Dlatego osiągnięcie co najmniej 90,0% dokładności uczenia dla funkcji aktywacji ReLU wymaga co najmniej 35 epok. Rozważane sieci CNN zostały przetestowane na dokładnie tym samym zestawie zdjęć. Graficzna interpretacja omawianego scenariusza przedstawiona jest na rys. 4.11-4.13.

Dla przejrzystości, pomijamy prezentację macierzy konfuzji z wynikami identyfikacji aparatów dla poszczególnych sieci CNN oraz funkcji aktywacji. Możemy powiedzieć, że uczenie sieci wszystkimi rozważanymi funkcjami aktywacji pozwala na dokładność identyfikacji na poziomie co najmniej 95,0%, a wyniki są podobne, jak w przypadku algorytmu CompaRe (pod-

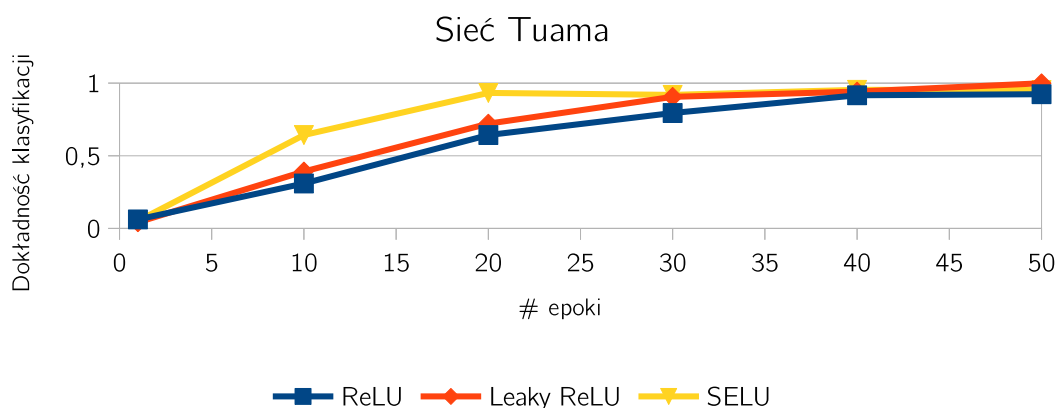


Rysunek 4.11: Dokładność treningu (50 epok) – porównanie funkcji aktywacji ReLU, Leaky ReLU oraz SELU – proponowana sieć CNN



Rysunek 4.12: Dokładność treningu (50 epok) – porównanie funkcji aktywacji ReLU, Leaky ReLU oraz SELU – sieć Bondi

rozd. 4.3.2). W przypadku uczenia z aktywacjami warstw konwolucyjnych w oparciu o funkcję SELU, takie same wyniki możemy uzyskać przy znacznie mniejszej liczbie epok. Wskaźniki TP dla niemal wszystkich rozważanych CNN wynoszą co najmniej 90,0% po treningu już 18-20 epok z funkcją SELU, podczas gdy uzyskanie takiego wyniku z funkcjami Leaky ReLU oraz ReLU wymaga odpowiednio 30-35 i co najmniej 45 epok.



Rysunek 4.13: Dokładność treningu (50 epok) – porównanie funkcji aktywacji ReLU, Leaky ReLU oraz SELU – sieć Tuama

## 4.5 Podsumowanie

W niniejszym rozdziale przedstawiono wyniki klasyfikacji proponowanych algorytmów w odniesieniu do algorytmów z literatury. Eksperymenty zostały przeprowadzone z wykorzystaniem reprezentatywnej liczby zdjęć, pochodzących zarówno ze starszych urządzeń (zbiór Dresden Image Database), jak i z nowoczesnych urządzeń (proponowany w rozprawie zbiór IMAGINE). Przedstawiono i zinterpretowano macierze konfuzji odnoszące się zarówno do proponowanych algorytmów, jak i literaturowych. Eksperymenty zostały przeprowadzone z wykorzystaniem blisko 14 000 zdjęć, pochodzących z ponad stu urządzeń.

Wyniki wskazały, że spośród proponowanych algorytmów najwyższą skuteczność identyfikacji wykazuje algorytm CompaRe. Jakość klasyfikacji nie odbiega poziomem od szeregu nowoczesnych metod dostępnych w literaturze, które identyfikują sensory zarówno na bazie kompaktowej formy cyfrowego odcisku palca, jak i w sposób standardowy. Wykorzystanie algorytmu CompaRe pozwala na wzrost szybkości przetwarzania zdjęć, jak również ograniczenie zasobów sprzętowych koniecznych do przechowywania cyfrowych odcisków palca.

Algorytm MSE-DSI oferuje kompromis pomiędzy akceptowalną jakością klasyfikacji, a szybkością działania w odniesieniu do algorytmów z literatury. Jego poprawność klasyfikacji jest jednak wyraźnie wyższa od modelu losowego, co może uczynić ten algorytm przydatnym w kontekście wstępnego przetwarzania zdjęć (ang. *preprocessing*). Ponadto jego zaletą jest identyfikacja aparatów w oparciu o liczbową (zamiast macierzową) miarę MSE, co dodatkowo ułatwia przechowywanie cyfrowych odcisków palca na przestrzeni dyskowej komputera.

Algorytmy Vignetting-CT oraz Distortion-CT oferują bardzo szybką identyfikację na bazie

wad optycznych winietowania oraz dystorsji obiektywu. Oba algorytmy mogą być przydatne do szybkiego rozpoznania, czy określony zbiór zdjęć został wygenerowany jednym czy większą liczbą urządzeń. Istotnymi ograniczeniami algorytmów Vignetting-CT oraz Distortion-CT jest konieczność analizy takich samych kadrów (w przypadku Vignetting-CT także w takich samych warunkach oświetleniowych). Eksperymenty przeprowadzone z wykorzystaniem zbioru smartfonów wykazały niższą skuteczność identyfikacji algorytmu Lukás. Może to być spowodowane potencjalną silną ingerencją algorytmów przetwarzania obrazu zainstalowanych w smartfonach na generowane zdjęcia.

Przeanalizowano czasy przetwarzania zdjęć dla wszystkich algorytmów. Wszystkie proponowane algorytmy oferują czasy przetwarzania zdjęć zauważalnie niższe porównując do algorytmów z literatury. Powodem jest większa ilość przetwarzanych danych przez algorytmy literaturowe. W proponowanych algorytmach najczęściej przetwarzamy jeden kanał kolorystyczny analizowanych zdjęć zamiast trzech, jak również proponujemy przetwarzanie małych fragmentów obrazów zamiast wykorzystania pełnej ich rozdzielczości.

# Rozdział 5

## Analiza statystyczna uzyskanych wyników

W niniejszym rozdziale zaprezentujemy obszerną analizę statystyczną wyników przedstawionych w rozdz. 4. Weryfikacji statystycznej poddamy proponowany zbiór zdjęć IMAGINE oraz algorytmy MSE-DSI, CompaRe oraz Vignetting-CT. Głównym celem jest porównanie wartości TP (a więc wartości na głównej przekątnej macierzy konfuzji) uzyskanych przez proponowane algorytmy ze wartościami TP algorytmów z literatury i sprawdzenie, czy istnieją między nimi różnice statystyczne. Celem jest także porównanie wartości TP uzyskanych przez wszystkie algorytmy (proponowane oraz literaturowe) w ramach zbiorów danych (IMAGINE oraz Dresden Image Database). Krótko mówiąc, porównamy wyniki każdego algorytmu w odniesieniu do zbioru IMAGINE oraz Dresden, np. wyniki algorytmu CompaRe na zbiorze IMAGINE kontra wyniki algorytmu CompaRe na zbiorze Dresden.

W pierwszej części rozdziału omówimy podstawowe statystyki opisowe uzyskanych wyników, w drugiej sformułujemy odpowiednie hipotezy i poddamy je analizie statystycznej. Rozdział kończy krótkie podsumowanie.

### 5.1 Wstęp teoretyczny do analizy statystycznej

#### 5.1.1 Statystyki opisowe

Celem statystyk opisowych jest podsumowanie wyników w oparciu o podstawowe wskaźniki statystyczne. Posiadane dane można ocenić poprzez np. porównanie miar tendencji centralnej (np. średniej: arytmetycznej, geometrycznej, harmonicznej; mediany, dominanty, zwanej także

modą), miar rozkładu (skośność, błąd skośności, kurtoza, błąd kurtozy), miar rozproszenia (wariancja, odchylenie standardowe, rozstęp, współczynnik zmienności) oraz percentyli (wartość minimalna i maksymalna, dolny oraz górny kwartył).

### 5.1.2 Testowanie hipotez

Proces uogólnienia wyników otrzymanych dla próby możemy podzielić na dwie części:

- (1) Estymację – szacowanie wartości parametrów populacji na podstawie próby statystycznej;
- (2) Weryfikację hipotez statystycznych – sprawdzanie określonych założeń sformułowanych dla parametrów populacji generalnej na podstawie wyników z próby.

W niniejszym rozdziale skoncentrujemy się na drugim aspekcie, czyli na weryfikacji hipotez statystycznych. W ogólności, weryfikację hipotez przeprowadza się w trzech krokach:

- (1) Zdefiniowanie hipotez, które będą weryfikowane za pomocą testów statystycznych. Każdy test statystyczny podaje ogólną postać hipotezy zerowej  $\mathcal{H}_0$  (ang. *null hypothesis*) oraz alternatywnej  $\mathcal{H}_1$  (ang. *alternative hypothesis*):
  - $\mathcal{H}_0$ : **nie ma** ważnej statystycznie różnicy między (rozkładami, średnimi, wariancjami, odchyleniem standardowym, ...) populacji;
  - $\mathcal{H}_1$ : **istnieje** ważna statystycznie różnica między (rozkładami, średnimi, wariancjami, odchyleniem standardowym, ...) populacji;
- (2) Sprawdzenie, która z hipotez ( $\mathcal{H}_0$  lub  $\mathcal{H}_1$ ) jest bardziej prawdopodobna. W tym celu należy dobrać odpowiedni test statystyczny. W każdym teście oblicza się statystykę testową, która podlega odpowiedniemu dla niej rozkładowi teoretycznemu. Wartość takiej statystyki porównuje się z wartością krytyczną określonego rozkładu i na tej podstawie określa się, czy hipotezę zerową można przyjąć, bądź odrzucić na rzecz hipotezy alternatywnej;
- (3) Zinterpretowanie i opis wyników testowania w odniesieniu do badanego wycinka rzeczywistości.

Istotnym czynnikiem jest wybór poziomu istotności  $\alpha$ . Najczęściej w badaniach przyjmuje się  $\alpha = 0,05$ , co oznacza akceptację, że w 5% przypadków hipoteza zerowa zostanie odrzucona, podczas gdy jest ona prawdziwa. Poziom istotności  $\alpha$  reprezentuje więc prawdopodobieństwo popełnienia błędu pierwszego rodzaju (prawdopodobieństwo odrzucenia hipotezy  $\mathcal{H}_0$ , gdy jest ona prawdziwa).

Przyjęcie bądź odrzucenie hipotezy może także nastąpić na drodze wyznaczenia surowej  $p$  wartości (ang. *p-value*) prawdopodobieństwa popełnienia błędu pierwszego rodzaju. Obliczoną wartość  $p$  porównujemy z poziomem istotności  $\alpha$ . Wtedy:

- Jeśli  $p \leq \alpha$ : odrzucamy  $\mathcal{H}_0$ , przyjmując  $\mathcal{H}_1$ ;
- Jeśli  $p > \alpha$ : nie ma podstaw, aby odrzucić  $\mathcal{H}_0$ .

Istotnym czynnikiem warunkującym użycie odpowiedniego testu statystycznego jest określenie skali, w jakiej reprezentowane są badane dane. Wyróżniamy trzy główne typy skal pomiarowych:

(1) Skala interwałowa (przedziałowa; ang. *interval scale*)

Zmienna jest wyrażona na skali interwałowej (przedziałowej), jeśli:

- Można ją uporządkować;
- Można wyznaczyć różnicę lub iloraz między obserwowanymi zmiennymi i zazwyczaj określona jest jednostka miary, np. prędkość, masa, czas;

(2) Skala porządkowa (ang. *ordinal scale*)

Zmienna jest wyrażona na skali porządkowej, jeśli:

- Można ją uporządkować;
- Nie można wyznaczyć różnicy/ilorazu między obserwowanymi zmiennymi w „rozsądny” sposób, np. kolejność zawodników na podium;

(3) Skala nominalna (ang. *nominal scale*)

Zmienna jest wyrażona na skali nominalnej, jeśli:

- Nie można jej uporządkować;
- Nie można wyznaczyć różnicy/ilorazu między obserwowanymi zmiennymi, np. płeć, kraj zamieszkania, obywatelstwo.

Warunkiem stosowania wszystkich wymienionych w niniejszym podrozdziale testów jest pomiar obserwowanych zmiennych na skali interwałowej (przedziałowej).

### 5.1.3 Opis wykorzystanych testów

W niniejszym podrozdziale opiszemy skrótowo wykorzystane testy parametryczne i nieparametryczne. Aby możliwe było poddanie posiadanych danych reprezentujących dwie niezależne populacje testom parametrycznym, muszą one spełniać (między innymi) warunek normalności rozkładu. Krótko mówiąc, badane dane powinny mieć rozkład zgodny z rozkładem normalnym [92, 102, 103, 132]. Istnieje wiele testów normalności rozkładu, np. test Kołmogorova-Smirnova [92], test Lillieforsa [102, 103], a także test Shapiro-Wilka [132], który dzięki poprawce Roystona [127] nadaje się do weryfikowania normalności w grupach zawierających do 5000 obserwacji. Następnym krokiem może być analiza równości wariancji, którą można zrealizować za pomocą testu Fishera-Snedecora [116]. Kolejnym krokiem może być weryfikacja równości średnich za pomocą testu *t*-Studenta [71]. Należy jednak mieć na względzie, że test *t*-Studenta wymaga wiedzy o równości bądź braku równości wariancji w badanych populacjach. Jeśli więc test Fishera-Snedecora wskaże, że nie ma podstaw do odrzucenia hipotezy o równości wariancji, korzystamy ze „standardowego” testu *t*-Studenta. Jeśli jednak są podstawy, by odrzucić hipotezę o równości wariancji, wówczas korzystamy z testu *t*-Studenta z poprawką Cochran-Coxa [128]. Jeśli dysponujemy danymi, które pochodzą z prób zależnych, wówczas korzystamy z testu *t*-Studenta dla grup zależnych.

W przypadku, gdy testy normalności rozkładu wskażą podstawy do odrzucenia hipotezy zerowej o zgodności badanych danych z rozkładem normalnym, wówczas korzystamy z testów nieparametrycznych. Dla zmiennych niezależnych, możemy skorzystać z testu U Manna-Whitneya [109] do weryfikacji hipotezy o nieistotności różnic pomiędzy medianami.

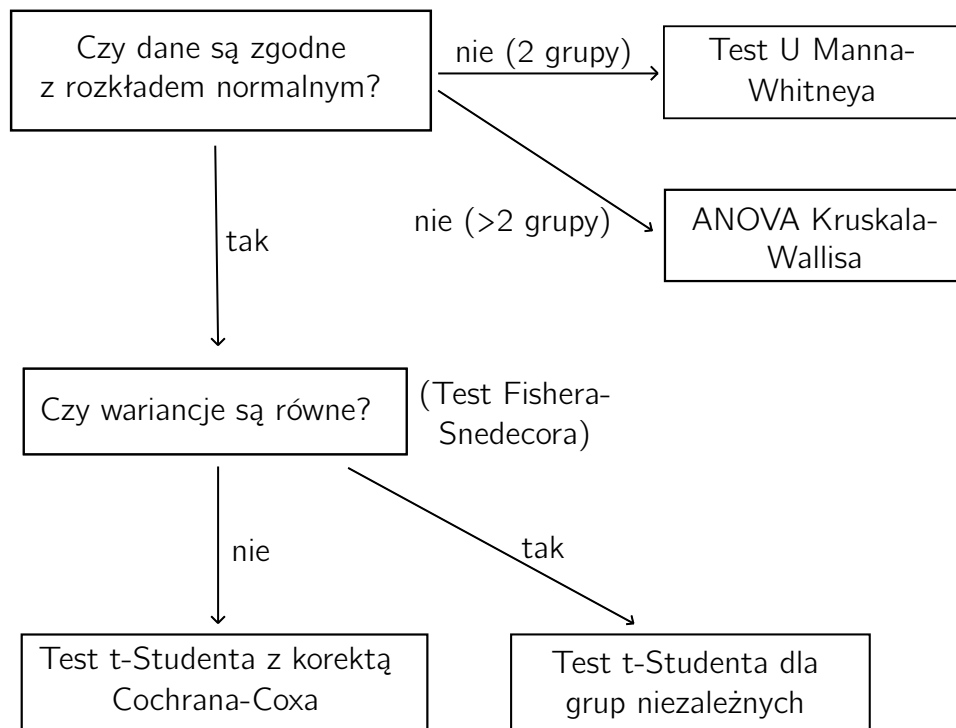
Wyżej wymienione testy są przeznaczone do analizowania danych reprezentujących dwie populacje. Jeśli jednak liczba populacji jest większa niż dwie, wówczas stosujemy testy przeznaczone do weryfikacji hipotez dla więcej niż dwóch populacji. W takim kontekście parametrycznym odpowiednikiem testu Fishera-Snedecora jest test Browna-Forsythea i Levene'a [27] na równość wariancji. Odpowiednikiem testu *t*-Studenta jest ANOVA dla grup niezależnych. Alternatywą dla nieparametrycznego testu U Manna-Whitneya jest test ANOVA Kruskala-Wallisa [90, 91]. Jeśli są podstawy do odrzucenia hipotezy zerowej o równości median, możemy dokładnie ocenić występujące tendencje wykonując analizę POST-HOC, na przykład metodą Dunn-Bonferroni [50].

Graficzna interpretacja omówionego scenariusza jest przedstawiona na rys. 5.1.

#### Testy normalności rozkładu

Ustalenie, czy badane dane są zgodne z rozkładem normalnym jest jednym z warunków stosowania testów parametrycznych. Skorzystamy z testu Shapiro-Wilka, by ustalić, czy badane





Rysunek 5.1: Procedura weryfikacji statystycznej uzyskanych wyników

przez nas dane są zgodne z rozkładem normalnym. W tym celu badane dane poddaje się testom normalności rozkładu. Hipotezy są zdefiniowane następująco:

- $\mathcal{H}_0$ : rozkład badanej cechy w populacji jest rozkładem normalnym;
- $\mathcal{H}_1$ : rozkład badanej cechy w populacji jest różny od rozkładu normalnego.

Warunki stosowania:

- Pomiar na skali interwałowej (przedziałowej).

Obliczoną wartość statystyki testowej porównujemy z wartością krytyczną statystyki, którą możemy odczytać z tablic rozkładu testu Shapiro-Wilka. Jeśli jest ona większa od wartości krytycznej, to przyjmujemy hipotezę  $\mathcal{H}_0$ . W przeciwnym razie odrzucamy  $\mathcal{H}_0$  by przyjąć  $\mathcal{H}_1$ . Statystykę testową Shapiro-Wilka obliczamy ze wzoru:

$$W = \frac{\sum_{i=1}^n a_i x_i}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (5.1)$$

gdzie:

- $a_i$  – współczynniki wyznaczone w oparciu o wartości oczekiwane dla statystyk uporządkowanych, przypisanych wag oraz macierzy kowariancji (odczytujemy je z tablic Shapiro-Wilka);
- $x_i$  – kolejne obserwacje z badanej próby;
- $\bar{x}$  – wartość średnia danych z próby.

Na podstawie statystyki testowej możemy także obliczyć wartość surowego prawdopodobieństwa przyjęcia hipotezy zerowej (zwyczajowo oznaczane w literaturze jako  $p$  lub  $p$ -value). Wówczas:

- Jeśli  $p \leq \alpha$ , to odrzucamy  $\mathcal{H}_0$  przyjmując  $\mathcal{H}_1$ ;
- Jeśli  $p > \alpha$ , to nie mamy podstaw, aby odrzucić  $\mathcal{H}_0$ .

Postępowanie to jest wspólne dla wszystkich testów statystycznych.

### Test Fishera-Snedecora

Test Fishera-Snedecora (ang. *F-Snedecor test*) opiera się na zmiennej  $F$  sformułowanej przez Fishera, której rozkład opisał Snedecor. Test ten służy do weryfikacji hipotezy o równości wariancji badanej zmiennej w dwóch populacjach [116]. Warunki stosowania:

- Pomiar na skali interwałowej;
- Normalność rozkładu badanej zmiennej w obu populacjach;
- Model niezależny.

Hipotezy są zdefiniowane następująco:

- $\mathcal{H}_0: \sigma_1^2 = \sigma_2^2$  (nie ma istotnej statystycznie różnicy pomiędzy wariancjami w obu populacjach);
- $\mathcal{H}_1: \sigma_1^2 \neq \sigma_2^2$  (istnieje istotna statystycznie różnica pomiędzy wariancjami w obu populacjach).

Statystykę testową obliczamy ze wzoru:

$$F = \frac{\sigma_1^2}{\sigma_2^2} \quad (5.2)$$

gdzie  $\sigma_1^2$  oraz  $\sigma_2^2$  to odpowiednio wariancje pierwszej i drugiej populacji. Statystyka ta podlega rozkładowi F Snedecora z  $n_1 - 1$  oraz  $n_2 - 1$  stopniami swobody.

**Test  $t$ -Studenta dla grup niezależnych**

Test  $t$ -Studenta dla grup niezależnych wykorzystujemy do weryfikacji hipotezy o równości średnich badanej zmiennej w dwóch grupach [71]. Warunki stosowania:

- Pomiar na skali interwałowej;
- Normalność rozkładu badanej zmiennej w obu populacjach;
- Model niezależny;
- Równość wariancji badanej zmiennej obu populacji.

Hipotezy są zdefiniowane następująco:

- $\mathcal{H}_0: \mu_1 = \mu_2$  (nie ma istotnej statystycznie różnicy pomiędzy średnimi w obu populacjach);
- $\mathcal{H}_1: \mu_1 \neq \mu_2$  (istnieje istotna statystycznie różnica pomiędzy średnimi w obu populacjach).

gdzie  $\mu_1$  oraz  $\mu_2$  to odpowiednio wartości średnie pierwszej i drugiej populacji. Statystyka testowa  $t$  obliczana jest ze wzoru:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}{n_1 + n_2 - 2} \left( \frac{1}{n_1} + \frac{1}{n_2} \right)}} \quad (5.3)$$

gdzie:

- $\bar{x}_1$  oraz  $\bar{x}_2$  to średnie w pierwszej i drugiej próbie;
- $n_1$  oraz  $n_2$  – licznosci w pierwszej i drugiej próbie;
- $\sigma_1^2$  oraz  $\sigma_2^2$  – wariancje w pierwszej i drugiej próbie.

Statystyka testowa ma rozkład  $t$ -Studenta z  $n_1 + n_2 - 2$  stopniami swobody.

### Test $t$ -Studenta dla grup niezależnych z poprawką Cochran-Coxa

Test  $t$ -Studenta dla grup niezależnych z poprawką Cochran-Coxa stosujemy w przypadku, gdy wariancje badanych zmiennych w obu populacjach są różne [128]. Warunki stosowania (poza wymogiem równości wariancji) oraz hipotezy są zdefiniowane identycznie, jak w przypadku testu  $t$ -Studenta dla grup niezależnych. Statystyka testowa jest natomiast obliczana z następującego wzoru:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (5.4)$$

Statystyka testowa ma rozkład  $t$ -Studenta z liczbą stopni swobody  $df$  obliczaną ze wzoru (Satterthwaite [131]):

$$df = \frac{\left(\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}\right)^2}{\left(\frac{\sigma_1^2}{n_1}\right)^2 \cdot \frac{1}{(n_1 - 1)} + \left(\frac{\sigma_2^2}{n_2}\right)^2 \cdot \frac{1}{(n_2 - 1)}} \quad (5.5)$$

### Test $U$ Manna-Whitneya

Test  $U$  Manna-Whitneya jest nieparametryczną alternatywą dla testu  $t$ -Studenta. Służy do weryfikacji hipotezy o nieistotności różnic pomiędzy medianami badanej zmiennej w dwóch populacjach [109]. Warunki stosowania są następujące:

- Pomiar na skali interwałowej lub porządkowej;
- Model niezależny.

Hipotezy definiujemy następująco:

- $\mathcal{H}_0: \theta_1 = \theta_2$  (nie ma istotnej statystycznie różnicy pomiędzy medianami w obu populacjach);
- $\mathcal{H}_1: \theta_1 \neq \theta_2$  (istnieje istotna statystycznie różnica pomiędzy medianami w obu populacjach).

gdzie  $\theta_1$  oraz  $\theta_2$  – mediany w pierwszej i drugiej próbie.

Jeśli liczność analizowanych prób nie przekracza 20 obserwacji, statystyka testowa  $U$  obliczana jest ze wzoru:

$$U = n_1 n_2 \frac{n_1(n_1 + 1)}{2} - R_1 \quad (5.6)$$

lub

$$U' = n_1 n_2 \frac{n_2(n_2 + 1)}{2} - R_2 \quad (5.7)$$

gdzie:

- $n_1$  oraz  $n_2$  – liczności w pierwszej i drugiej próbie;
- $R_1$  oraz  $R_2$  – sumy rang dla prób.

Do analizy wybieramy mniejszą z wartości  $U$  lub  $U'$ . Jeśli jednak badane próby zawierają więcej niż 20 obserwacji, wówczas statystykę testową obliczamy z następującego wzoru:

$$Z = \frac{U - \frac{n_1 n_2}{2}}{\sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}} \quad (5.8)$$

### Test ANOVA Kruskala-Wallis

Test ANOVA Kruskala-Wallis (zwany też jednoczynnikową analizą wariancji dla rang Kruskala-Wallis) jest rozszerzeniem testu U Manna-Whitneya na więcej niż dwie populacje. Test ten wykorzystywany jest do weryfikacji hipotezy o nieistotności różnic pomiędzy medianami badanej zmiennej w kilku ( $k > 2$ ) populacjach [90, 91]. Warunki stosowania:

- Pomiar na skali interwałowej lub porządkowej;
- Model niezależny.

Hipotezy są zdefiniowane następująco:

- $\mathcal{H}_0$ :  $\theta_1 = \theta_2 = \dots = \theta_k$  (nie ma istotnej statystycznie różnicy pomiędzy medianami w badanych  $k$  populacjach);
- $\mathcal{H}_1$ : nie wszystkie  $\theta_j$  są sobie równe (dla  $j = 1, 2, \dots, k$ ).

gdzie  $\theta_1, \theta_2, \dots, \theta_k$  to mediany badanej zmiennej w populacjach, z których pobrano próby. Statystykę testową obliczamy ze wzoru:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^k \left( \frac{\left( \sum_{i=1}^{n_j} R_{ij} \right)^2}{n_j} \right) - 3(N+1) \quad (5.9)$$

gdzie:

- $N = \sum_{j=1}^k n_j$ ;
- $n_j$  – liczności prób (dla  $j = 1, 2, \dots, k$ );
- $R_{ij}$  – rangi przypisane do wartości zmiennej ( $i = 1, 2, \dots, n_j, j = 1, 2, \dots, k$ ).

Statystyka ma rozkład  $\chi^2$  z  $k - 1$  stopniami swobody.

## 5.2 Statystyki opisowe

W niniejszym podrozdziale omówimy statystyki opisowe zarówno proponowanych algorytmów, jak i odnoszących się do nich algorytmów z literatury. Doprecyzujemy, że prezentowane wartości są obliczone na podstawie wartości wskaźników TP z macierzy konfuzji, które reprezentują dokładność identyfikacji aparatu cyfrowego na podstawie zdjęć.

### 5.2.1 Algorytm MSE-DSI

#### Identyfikacja aparatów cyfrowych

W tab. 5.1 oraz 5.2 przedstawiono statystyki opisowe porównujące algorytm MSE-DSI z algorytmami z literatury, odpowiednio dla zbiorów IMAGINE oraz Dresden Image Database.

Tabela 5.1: Statystyki opisowe: algorytm MSE-DSI kontra literatura (zbiór IMAGINE)

Algorytm	śr. arytm.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
MSE-DSI	73,1	75	wiele	10,8	3,3	12	69	81
Lukás	95,3	95	95	0,5	0,7	2	94	96
Bondi	95,6	95	95	1,4	1,2	4	94	98
Tuama	95,9	96	95	0,7	0,8	2	95	97

Wyniki zebrane w tabelach wskazują na niższą skuteczność klasyfikacyjną algorytmu MSE-DSI w rozumieniu podstawowych wskaźników statystycznych. Zarówno dla zbiorów IMAGINE, jak i Dresden, proponowany algorytm uzyskuje średnią arytmetyczną, odpowiednio 73,1 oraz 79,1, podczas gdy dla algorytmów Lukás, Bondi i Tuama wartość ta wynosi od 93 do 96. Także wartości mediany są niższe w przypadku MSE-DSI, natomiast wyższe wartości odchylenia standardowego, wariancji oraz rozstępu wskazują na niższą „stabilność” proponowanej metody względem metod z literatury.

Tabela 5.2: Statystyki opisowe: algorytm MSE-DSI kontra literatura (zbiór Dresden Image Database)

Algorytm	śr. arytm.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
MSE-DSI	79,1	80,5	81	24,6	5	24	65	89
Lukás	93,7	93,5	93	4,2	2	7	90	97
Bondi	95,5	96	96	1,8	1,4	5	93	98
Tuama	95,5	96	96	1,4	1,2	5	93	98

### Identyfikacja skanerów płaskich

Dane opisujące wskaźniki statystyczne identyfikacji skanerów płaskich są zawarte w tab. 5.3.

Tabela 5.3: Statystyki opisowe: algorytm MSE-DSI kontra Khanna (identyfikacja skanerów płaskich)

Algorytm	śr. arytm.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
MSE-DSI	81,3	88	100	431,3	20,8	59	41	100
Khanna	90,1	90,5	wiele	12,5	3,5	11	85	96

W przypadku identyfikacji skanerów płaskich algorytm MSE-DSI osiąga na ogół gorsze wskaźniki niż algorytm Khanna, różnice nie są jednak tak wyraźne, jak w przypadku identyfikacji aparatów cyfrowych. Średnia arytmetyczna proponowanej metody wynosi 81,3, natomiast dla algorytmu Khanna 90,1. W kontekście mediany obie metody osiągają niemal tę samą wartość, z bardzo niewielką przewagą algorytmu Khanna, dla którego wskaźnik ten osiąga 90,5 (88 dla MSE-DSI). Niemniej jednak relatywnie duża wartość odchylenia standardowego sugeruje mniejszą „stabilność” algorytmu MSE-DSI. Szczególnie przeciętnie wypada miara rozstępu, która dla MSE-DSI osiąga wartość aż 59, podczas gdy dla metody Khanna jest to jedynie 11. Niemniej jednak można uznać, że wyniki obu algorytmów są do siebie zbliżone.

### 5.2.2 Algorytm CompaRe

Statystyki opisujące wyniki uzyskane algorytmem CompaRe w odniesieniu do algorytmów kompaktowej reprezentacji cyfrowego odcisku palca zostały przedstawione w tab. 5.4 oraz 5.5; w tab. 5.6 oraz 5.7 zestawiono wskaźniki statystyczne dla pozostałych algorytmów z literatury.

Wyniki uzyskane przez wszystkie algorytmy są bardzo zbliżone do siebie niemal we wszystkich rozważanych wskaźnikach statystycznych. Można jednak zauważyć przewagę algorytmu

Tabela 5.4: Statystyki opisowe: algorytm CompaRe kontra algorytmy kompaktowej reprezentacji (zbiór IMAGINE)

Algorytm	śr. arytm.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
CompaRe	95,7	96	96	0,2	0,5	1	95	96
Valsesia	93,2	93	wiele	1,1	1,1	3	92	95
Li	93,4	94	94	0,9	0,9	3	92	95

Tabela 5.5: Statystyki opisowe: algorytm CompaRe kontra algorytmy kompaktowej reprezentacji (zbiór Dresden)

Algorytm	śr. arytm.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
CompaRe	95,9	96	96	0,5	0,7	2	95	97
Valsesia	94,0	94	94	0,5	0,7	2	93	95
Li	93,0	93	93	1,2	1,1	4	91	95

Tabela 5.6: Statystyki opisowe: algorytm CompaRe kontra pozostałe algorytmy (zbiór Dresden)

Algorytm	śr. arytm.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
CompaRe	95,7	96	96	0,2	0,5	1	95	96
Lukás	96,0	96	96	0,5	0,7	2	95	97
Bondi	93,2	93	94	1,7	1,3	5	90	95
Tuama	91,7	92	wiele	3,4	1,8	6	89	95
Mandelli	92,4	93	93	3,2	1,8	6	89	95
Kirchner	91,7	92	92	2,5	1,6	6	89	95

Tabela 5.7: Statystyki opisowe: algorytm CompaRe kontra pozostałe algorytmy (zbiór Dresden)

Algorytm	śr. arytm.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
CompaRe	95,9	96	96	0,5	0,7	2	95	97
Lukás	96,1	96	96	0,5	0,7	3	94	97
Bondi	94,8	95	95	0,4	0,6	2	94	96
Tuama	94,1	94	94	0,8	0,9	3	93	96
Mandelli	94,4	94	94	0,8	0,9	3	93	96
Kirchner	91,8	92	92	2,7	1,6	6	88	94

CompaRe, szczególnie w porównaniu do algorytmów Valsesia oraz Li, gdzie średnia arytmetyczna, mediana i rozstęp pomiędzy wartością minimalną i maksymalną jest najmniejszy. Minimalna



wartość TP w przypadku algorytmu CompaRe jest na poziomie maksymalnych wartości uzyskanych przez algorytmy Valsesia oraz Li. Wartości wariancji oraz odchyłeń standardowych dla wszystkich algorytmów są na bardzo niskim poziomie, co potwierdza ich stabilność.

W przypadku pozostałych algorytmów (Lukás—Kirchner & Johnson) wyniki są bardziej wyrównane, choć w niektórych przypadkach również niewielką przewagę osiąga proponowany algorytm. Najniższe wartości wskaźników w rozumieniu średniej arytmetycznej, mediany, dominanty oraz rozstępu obserwujemy dla metody Kirchner & Johnson, zarówno dla zbiorów IMAGINE, jak i Dresden. Dowodzi to zatem, że pomimo oczywistej wszechstronności konwolucyjnych sieci neuronowych, zastosowanie zbyt „płytkiej” sieci może skutkować obniżeniem jakości klasyfikacji. W pozostałych przypadkach wyniki są znacznie bardziej wyrównane, choć dla sieci CNN Tuama (zbiór IMAGINE) również możemy zauważyć nieco niższe wskazania. Są one ujawnione w postaci wyższej wartości odchylenia standardowego (odnosząc do pozostałych algorytmów) oraz wartością minimalną na poziomie 89,0%. Algorytm CompaRe ujawnia najmniejszy rozstęp pomiędzy wartością minimalną a maksymalną i wraz z algorytmem Lukás, osiąga najwyższą wartość średniej arytmetycznej, która jest równoznaczna z poprawnością klasyfikacji ACC. Co ciekawe, wspomniany algorytm Lukás uzyskuje najwyższą średnią arytmetyczną, nie tylko nieznacznie pokonując algorytm CompaRe, ale także pozostałe metody oparte na uczeniu głębokim. Dowodzi to zatem niezwyklej wszechstronności tego algorytmu, który pomimo ponad 15-letniego istnienia, nie „odstaje” od współczesnej konkurencji.

### 5.2.3 Algorytm Vignetting-CT

Statystyki opisujące wyniki algorytmu Vignetting-CT w zestawieniu z algorytmami z literatury przedstawiono w tab. 5.8, 5.9 oraz 5.10.

Tabela 5.8: Statystyki opisowe: algorytm Vignetting-CT kontra pozostałe algorytmy (zbiór IMAGINE)

Algorytm	śr. ar.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
Vignetting-CT	66,1	65	69	16,4	4,0	14	59	74
Lukás	95,0	95	95	0,6	0,8	3	94	97

Dane zawarte w tabelach wskazują na niższe osiągi klasyfikacyjne proponowanego algorytmu, porównując do algorytmu Lukás. Najbardziej wyraźne różnice obserwujemy dla zbiorów IMAGINE oraz Dresden, gdzie proponowana metoda rozpoznaje aparaty cyfrowe w około 64-66% przypadków, podczas gdy algorytm Lukás osiąga dokładność 93-95%. Analogicznie wyniki przedstawiają się w przypadku mediany i dominanty, których wartości są zbliżone do średniej

Tabela 5.9: Statystyki opisowe: algorytm Vignetting-CT kontra pozostałe algorytmy (zbiór Dresden)

Algorytm	śr. ar.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
Vignetting-CT	63,9	63,5	65	16,4	4,0	14	57	71
Lukás	93,5	94	94	1,8	1,3	5	91	96

Tabela 5.10: Statystyki opisowe: algorytm Vignetting-CT kontra pozostałe algorytmy (zbiór smartfonów)

Algorytm	śr. ar.	mediana	dominanta	wariancja	odch. st.	rozstęp	min	max
Vignetting-CT	81,8	83	95	176,6	13,3	44	51	95
Lukás	89,3	90	85	13,5	3,7	11	85	96

arytmetycznej dla obu algorytmów. Wyraźnie wyższa wariancja oraz odchylenie standardowe dla algorytmu Vignetting-CT wskazuje natomiast na mniejszą stabilność klasyfikacji. Również wartości minimalne oraz maksymalne są mniejsze niż w przypadku algorytmu z literatury.

Interesująco wygląda porównanie dla zbioru smartfonów. W tym przypadku wskaźniki statystyczne nie wskazują już wyraźnej przewagi algorytmu z literatury. Przykładowo, wartość średniej arytmetycznej dla algorytmu Lukás wynosi 89,3%; dla algorytmu Vignetting-CT jest to 81,8%; mediany osiągają wartości: 90 – Lukás i 83 – Vignetting-CT. Co ciekawe, wartość dominanta na poziomie 95 jest zauważalnie wyższa dla proponowanej metody (85 – Lukás). Wariancja oraz odchylenie standardowe nadal jednak wypada na niekorzyść metody Vignetting-CT, gdyż wynosi odpowiednio 176,6 oraz 13,3, podczas gdy dla metody Lukás jest to 13,5 i 3,7. Także w kontekście rozstępu proponowany algorytm osiąga gorszy wskaźnik, gdyż wartość minimalna klasyfikacji osiąga 51%, natomiast algorytm Lukás prezentuje 85%.

## 5.3 Weryfikacja hipotez statystycznych

Wszystkie testy przeprowadziliśmy na poziomie istotności  $\alpha = 0,05$ . W niniejszej rozprawie wszystkie prezentowane dane są wyrażone w skali interwałowej (przedziałowej).

### 5.3.1 Zbiór zdjęć IMAGINE

W niniejszym podrozdziale przedstawimy statystyczne porównanie proponowanego zbioru zdjęć IMAGINE oraz bazy Dresden Image Database. W tym celu porównamy wzajemnie parami

proponowane oraz literaturowe algorytmy na obu zbiorach, aby ustalić, czy istnieją statystyczne różnice pomiędzy dokładnością klasyfikacji danego algorytmu na zbiorze IMAGINE oraz na bazie Dresden. Analizie poddamy rezultaty TP rozważanych w rozprawie algorytmów dla obu wymienionych zbiorów zdjęć.

**Test normalności rozkładu** Analizę rozpoczynamy od testów normalności rozkładu, korzystamy z metody Shapiro-Wilka. Wartości statystyk testowych dla poszczególnych danych przedstawione są w tab. 5.11.

**Hipoteza 5.1.** *Wartości wskaźników TP reprezentujących wyniki klasyfikacji wszystkich rozważanych w rozprawie algorytmów (tab. 5.11) spełniają hipotezę zerową  $\mathcal{H}_0$  o zgodności z rozkładem normalnym, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku zgodności z rozkładem normalnym.*

*Dowód.* Wartość krytyczna dla testu Shapiro-Wilka odczytana z tablic dla grup o licznosci (odpowiednio) 48 i 53 na poziomie istotności  $\alpha = 0,05$  wynosi 0,947 oraz 0,950. Aby założyć zgodność badanych danych z rozkładem normalnym, obliczona wartość statystyki  $W$  powinna być większa od wartości krytycznej. Taką sytuację obserwujemy jedynie w przypadku algorytmu MSE-DSI, zarówno dla zbioru IMAGINE, jak i Dresden. Oznacza to, że tylko w tym przypadku możemy przyjąć hipotezę  $\mathcal{H}_0$  o zgodności z rozkładem normalnym. Dla pozostałych danych przyjmujemy hipotezę alternatywną  $\mathcal{H}_1$ . W związku z tym, do dalszej analizy powinniśmy użyć testu nieparametrycznego.  $\square$

**Wniosek 5.1.** *Na mocy testu Shapiro-Wilka:*

- (1) *Wskaźniki TP algorytmu MSE-DSI dla zbiorów zdjęć IMAGINE oraz Dresden są zgodne z rozkładem normalnym;*
- (2) *Wskaźniki TP pozostałych algorytmów na obu zbiorach zdjęć nie są zgodne z rozkładem normalnym.*

Tabela 5.11: Testy normalności rozkładu: zbiór IMAGINE kontra Dresden

Algorytm	Analizowane zmienne	Liczność	Statystyka $W$
MSE-DSI	MSE-DSI (IMAGINE)	53	0,96
	MSE-DSI (Dresden)	48	0,96
	Lukás (IMAGINE)	53	0,79
	Lukás (Dresden)	48	0,94
	Bondi (IMAGINE)	53	0,90
	Bondi (Dresden)	48	0,94
	Tuama (IMAGINE)	53	0,79
	Tuama (Dresden)	48	0,86
CompaRe	CompaRe (IMAGINE)	53	0,58
	CompaRe (Dresden)	48	0,81
	Valsesia (IMAGINE)	53	0,86
	Valsesia (Dresden)	48	0,81
	Li (IMAGINE)	53	0,84
	Li (Dresden)	48	0,92
	CompaRe (IMAGINE)	53	0,58
	CompaRe (Dresden)	48	0,81
	Lukás (IMAGINE)	53	0,80
	Lukás (Dresden)	48	0,82
	Bondi (IMAGINE)	53	0,88
	Bondi (Dresden)	48	0,75
	Tuama (IMAGINE)	53	0,89
	Tuama (Dresden)	48	0,85
	Mandelli (IMAGINE)	53	0,89
	Mandelli (Dresden)	48	0,85
Kirchner (IMAGINE)	53	0,92	
Kirchner (Dresden)	48	0,89	
Vignetting-CT	Vignetting-CT (IMAGINE)	53	0,95
	Vignetting-CT (Dresden)	48	0,94
	Lukás (IMAGINE)	53	0,77
	Lukás (Dresden)	48	0,94

**Test ANOVA Kruskala-Wallisa** W związku z tym, że nasze dane w większości przypadków są niezgodne z rozkładem normalnym, do dalszej analizy używamy testu nieparametrycznego.

Skorzystamy z metody ANOVA Kruskala-Wallisa, ponieważ chcemy porównać ze sobą więcej niż dwie grupy danych. Przeprowadzimy cztery grupy porównań (P1)–(P4), w których sprawdzimy, czy zarówno dla zbioru IMAGINE, jak i Dresden Image Database istnieje statystycznie ważna różnica pomiędzy wynikami algorytmów:

(P1) MSE-DSI, Lukás, Bondi, Tuama;

(P2) CompaRe, Valsesia, Li;

(P3) CompaRe, Lukás, Bondi, Tuama, Mandelli, Kirchner & Johnson;

(P4) Vignetting-CT, Lukás.

W każdym porównaniu porównamy wartości TP danego algorytmu uzyskanymi zarówno na zbiorze IMAGINE, jak i Dresden. Przykładowo, w (P2) dokonamy następujących porównań: algorytm CompaRe (dla IMAGINE) kontra CompaRe (dla Dresden); Valsesia (dla IMAGINE) kontra Valsesia (dla Dresden) oraz Li (dla IMAGINE) kontra Li (dla Dresden); analogicznie dla pozostałych grup porównań.

### Porównanie (P1)

**Hipoteza 5.2.** Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów MSE-DSI, Lukás, Bondi oraz Tuama pomiędzy zbiorami IMAGINE oraz Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.

*Dowód.* Obliczymy statystykę  $H$ . W tym celu, obliczymy niezbędne dane, by móc je podstawić do wzoru 5.9. Wartości są następujące:

- Dla  $N = \sum_{j=1}^k n_j$ , mamy  $k = 8$ ;  $n_1 = n_3 = n_5 = n_7 = 53$ ;  $n_2 = n_4 = n_6 = n_8 = 48$ , a więc uzyskujemy:  $N = 4 \cdot 53 + 4 \cdot 48 = 404$ ;
- Wyznamy sumy rang  $R_1, \dots, R_8$ . Wyniki są przedstawione w tab. 5.12.

Podstawmy do wzoru 5.9:

$$H = \frac{12}{404(404 + 1)} \cdot \left( \frac{2046,0^2}{53} + \frac{3105,0^2}{48} + \frac{12681,5^2}{53} + \frac{8998,5^2}{48} + \frac{14171,5^2}{53} + \frac{12681,0^2}{48} + \frac{15558,0^2}{53} + \frac{12568,5^2}{48} \right) - 3(404 + 1) = \frac{1}{13635} \cdot 19998547 - 1217 = 251,71$$

Tabela 5.12: Sumy rang, porównanie (P1)

Zbiór:	IMAGINE	Dresden
Algorytm	Suma rang	Suma rang
MSE-DSI	$R_1 = 2046,0$	$R_2 = 3105,0$
Lukás	$R_3 = 12681,5$	$R_4 = 8998,5$
Bondi	$R_5 = 14171,5$	$R_6 = 12681,0$
Tuama	$R_7 = 15558,0$	$R_8 = 12568,5$

Wartość krytyczna statystyki  $\chi^2$  z  $8 - 1 = 7$  stopniami swobody wynosi  $H_c = 14,067$ . W związku z tym, że

$$H > H_c,$$

powinniśmy odrzucić hipotezę zerową o statystycznej równości median rozważanych danych.  $\square$

**Wniosek 5.2.** *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP algorytmów MSE-DSI, Lukás, Bondi oraz Tuama pomiędzy zbiorami IMAGINE oraz Dresden Image Database na mocy testu ANOVA Kruskala-Wallis.*

Aby dokładnie ocenić tendencje występujących różnic, możemy wykonać analizę POST-HOC (skorzystamy z metody Dunn-Bonferroni [50]). Metoda POST-HOC ułatwia ocenę, które z analizowanych danych różnią się między sobą. Istotność różnic wyznaczamy w standardowy sposób, a więc porównując wartości  $p$  analizy POST-HOC z poziomem istotności  $\alpha$ . Wyniki analizy POST-HOC dla (P1) przedstawione są w tab. 5.13.

Tabela 5.13: Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P1)

Algorytm (wyniki dla IMAGINE)	Algorytm (wyniki dla Dresden)	$p$
MSE-DSI	MSE-DSI	1
Lukás	Lukás	0,67
Bondi	Bondi	1
Tuama	Tuama	1

Wyniki przedstawione w tab. 5.13 jasno wskazują, że w żadnym z przypadków nie ma podstaw, by uznać, że prezentowane dane różnią się od siebie istotnie. We wszystkich przypadkach (poza algorytmem Lukás)  $p$  osiąga maksymalną wartość, co nie daje żadnych podstaw do odrzucenia hipotezy o równości badanych danych. Dla algorytmu Lukás  $p = 0,67$ , jednak jest

to na tyle wysoka wartość, że również i w tym przypadku nie mamy podstaw do odrzucenia hipotezy zerowej. Tak więc możemy uznać, że zarówno proponowany algorytm, jak i metody z literatury osiągają ze statystycznego punktu widzenia te same rezultaty klasyfikacyjne, co potwierdza przydatność proponowanego zbioru IMAGINE.

Test ANOVA Kruskala-Wallisa wskazał jednak, że istnieją statystycznie ważne różnice dla median powyższych danych. Dotyczą one jednak różnic między konkretnymi algorytmami (na przykład istnieje statystyczna różnica pomiędzy wynikami algorytmu MSE-DSI oraz Lukás). Na tym etapie analizy nie jest ona jednak dla nas istotna; aspekt ten zbadamy w następnym podrozdziale.

**Wniosek 5.3.** *Nie istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmów MSE-DSI, Lukás, Bondi oraz Tuama na zbiorach IMAGINE oraz Dresden Image Database na mocy analizy POST-HOC.*

### Porównanie (P2)

**Hipoteza 5.3.** *Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów CompaRe, Valsesia oraz Li pomiędzy zbiorami IMAGINE oraz Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Dla grupy porównań (P2), dane niezbędne do obliczenia statystyki  $H$  przedstawiają się następująco:

- Dla  $N = \sum_{j=1}^k n_j$ , mamy  $k = 6$ ;  $n_1 = n_3 = n_5 = 53$ ;  $n_2 = n_4 = n_6 = 48$ , zatem uzyskujemy:  $N = 3 \cdot 53 + 3 \cdot 48 = 303$ ;
- Obliczmy sumy rang  $R_1, \dots, R_6$ . Wyniki są przedstawione w tab. 5.14.

Tabela 5.14: Sumy rang, porównanie (P2)

Zbiór:	IMAGINE	Dresden
Algorytm	Suma rang	Suma rang
CompaRe	$R_1 = 13003,0$	$R_2 = 12131,5$
Valsesia	$R_3 = 5011,0$	$R_4 = 6606,5$
Li	$R_5 = 5403,5$	$R_6 = 3900,5$

Obliczmy statystykę  $H$ :

$$H = \frac{12}{303(303 + 1)} \cdot \left( \frac{13003,0^2}{53} + \frac{12131,5^2}{48} + \frac{5011,0^2}{53} + \frac{6606,5^2}{48} + \frac{5403,5^2}{53} + \frac{3900,5^2}{48} \right) - 3(303 + 1) = \frac{1}{7676} \cdot 8506908 - 912 = 196,25$$

Wartość krytyczna statystyki  $\chi^2$  z  $6 - 1 = 5$  stopniami swobody wynosi  $H_c = 11,07$ . W związku z tym, że

$$H > H_c,$$

powinniśmy odrzucić hipotezę zerową o statystycznej równości median rozważanych danych.  $\square$

**Wniosek 5.4.** *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP algorytmów CompaRe, Valsesia oraz Li na zbiorach IMAGINE oraz Dresden Image Database na mocy testu ANOVA Kruskala-Wallis.*

Skorzystajmy z porównania POST-HOC (tab. 5.15):

Tabela 5.15: Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P2)

Algorytm (wyniki dla IMAGINE)	Algorytm (wyniki dla Dresden)	$p$
CompaRe	CompaRe	1
Valsesia	Valsesia	0,18
Li	Li	1

Analiza przedstawiona w tab. 5.15 wskazuje, że nie ma podstaw do odrzucenia hipotezy zerowej o równości median dla algorytmów CompaRe, Valsesia oraz Li w kontekście porównania na zbiorach IMAGINE oraz Dresden. Wprawdzie dla algorytmu Valsesia  $p = 0,18$ , co jest niezbyt wysoką wartością, jednak nie powoduje to konieczności odrzucenia  $\mathcal{H}_0$ . Oznacza to, że badane algorytmy nie wykazują statystycznej różnicy w klasyfikacji na zbiorze IMAGINE oraz Dresden. Analogicznie, jak w przypadku zestawienia (P1), statystyczne różnice wskazane przez metodą Kruskala-Wallis występują bezpośrednio pomiędzy algorytmami, co dokładnie przeanalizujemy w dalszych rozdziałach rozprawy.

**Wniosek 5.5.** *Nie istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmów CompaRe, Valsesia oraz Li na zbiorach IMAGINE oraz Dresden Image Database na mocy analizy POST-HOC.*



**Porównanie (P3)**

**Hipoteza 5.4.** Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson pomiędzy zbiorami IMAGINE oraz Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.

*Dowód.* Obliczmy statystykę  $H$  dla porównań (P3):

- Dla  $N = \sum_{j=1}^k n_j$ , mamy  $k = 12$ ;  $n_1 = n_3 = n_5 = n_7 = n_9 = n_{11} = 53$ ;  $n_2 = n_4 = n_6 = n_8 = n_{10} = n_{12} = 48$ , zatem uzyskujemy:  $N = 6 \cdot 53 + 6 \cdot 48 = 606$ ;
- Obliczmy sumy rang  $R_1, \dots, R_{12}$ . Wyniki są przedstawione w tab. 5.16.

Tabela 5.16: Sumy rang, porównanie (P3)

Zbiór:	IMAGINE	Dresden
Algorytm	Suma rang	Suma rang
CompaRe	$R_1 = 24844,5$	$R_2 = 23339,0$
Lukás	$R_3 = 26203,5$	$R_4 = 24137,5$
Bondi	$R_5 = 11448,0$	$R_6 = 17079,5$
Tuama	$R_7 = 6749,5$	$R_8 = 5725,0$
Mandelli	$R_9 = 8890,0$	$R_{10} = 14983,5$
Kirchner	$R_{11} = 6321,0$	$R_{12} = 6019,0$

Podstawiamy dane do wzoru:

$$\begin{aligned}
 H &= \frac{12}{606(606+1)} \cdot \left( \frac{24844,5^2}{53} + \frac{23339,0^2}{48} + \frac{26203,5^2}{53} + \frac{24137,0^2}{48} + \frac{11448,5^2}{53} + \frac{17079,5^2}{48} + \right. \\
 &\quad \left. + \frac{6749,5^2}{53} + \frac{13906,0^2}{48} + \frac{8890,0^2}{53} + \frac{14983,0^2}{48} + \frac{6321,0^2}{53} + \frac{6019,0^2}{48} \right) - 3(606+1) = \\
 &= \frac{2}{61307} \cdot 69202629 - 1821 = 436,58
 \end{aligned}$$

Wartość krytyczna statystyki  $\chi^2$  z  $12 - 1 = 11$  stopniami swobody wynosi  $H_c = 19,675$ . W związku z tym, że

$$H > H_c,$$

powinniśmy odrzucić hipotezę zerową o statystycznej równości median rozważanych danych.  $\square$

**Wniosek 5.6.** *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson na zbiorach IMAGINE oraz Dresden Image Database na mocy testu ANOVA Kruskala-Wallis.*

Przeprowadźmy analizę POST-HOC (tab. 5.17).

Tabela 5.17: Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P3)

Algorytm (wyniki dla IMAGINE)	Algorytm (wyniki dla Dresden)	$p$
CompaRe	CompaRe	1
Lukás	Lukás	1
Bondi	Bondi	0,54
Tuama	Tuama	0,01
Mandelli	Mandelli	0,01
Kirchner	Kirchner	1

Wyniki zawarte w tab. 5.17 wskazują, że dla metody Tuama oraz Mandelli  $p$  nie przekracza poziomu istotności  $\alpha$ . Oznacza to, że istnieje statystyczna różnica w jakości klasyfikacji dla tych metod. W pozostałych przypadkach wartość  $p$  wyraźnie przekracza zadany poziom istotności, dzięki czemu nie ma podstaw, by odrzucić hipotezę zerową.

**Wniosek 5.7.** *Na mocy analizy POST-HOC:*

- (1) *Nie istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmów CompaRe, Lukás, Bondi oraz Kirchner & Johnson na zbiorach IMAGINE oraz Dresden Image Database;*
- (2) *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmów Tuama oraz Mandelli na zbiorach IMAGINE oraz Dresden Image Database.*

### Porównanie (P4)

**Hipoteza 5.5.** *Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás pomiędzy zbiorami IMAGINE oraz Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Obliczmy statystykę  $H$  dla porównań (P4). Wyznaczymy niezbędne dane:

- Dla  $N = \sum_{j=1}^k n_j$ , mamy  $k = 4$ ;  $n_1 = n_3 = 53$ ;  $n_2 = n_4 = 48$ , a więc uzyskujemy:  $N = 2 \cdot 53 + 2 \cdot 48 = 202$ ;
- Obliczmy sumy rang  $R_1, \dots, R_4$ . Wyniki są przedstawione w tab. 5.18.

Tabela 5.18: Sumy rang, porównanie (P4)

Zbiór:	IMAGINE	Dresden
Algorytm	Suma rang	Suma rang
Vignetting-CT	$R_1 = 3075,0$	$R_2 = 2076,0$
Lukás	$R_3 = 8886,5$	$R_4 = 6465,5$

Podstawmy do wzoru 5.9:

$$\begin{aligned}
 H &= \frac{12}{202(202+1)} \cdot \left( \frac{3075,0^2}{53} + \frac{2076,0^2}{48} + \frac{8886,5^2}{53} + \frac{6465,5^2}{48} \right) - 3(202+1) = \\
 &= \frac{6}{20503} \cdot 2629082 - 609 = 160,37
 \end{aligned}$$

Wartość krytyczna statystyki  $\chi^2$  z  $4 - 1 = 3$  stopniami swobody wynosi  $H_c = 7,815$ . W związku z tym, że

$$H > H_c,$$

powinniśmy odrzucić hipotezę zerową o statystycznej równości median rozważanych danych.  $\square$

**Wniosek 5.8.** *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP algorytmów Vignetting-CT oraz Lukás na zbiorach IMAGINE oraz Dresden Image Database na mocy testu ANOVA Kruskala-Wallis.*

Skorzystajmy z analizy POST-HOC (tab 5.19):

Tabela 5.19: Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P4)

Algorytm (wyniki dla IMAGINE)	Algorytm (wyniki dla Dresden)	$p$
Vignetting-CT	Vignetting-CT	1
Lukás	Lukás	0,03

Porównanie (P4) przeprowadzone dla algorytmów Vignetting-CT oraz Lukás wskazuje, że algorytm Vignetting-CT zarówno na zbiorze IMAGINE, jak i Dresden osiąga podobne statystycznie rezultaty. Różnice obserwujemy jednak w przypadku algorytmu Lukás, gdzie  $p < \alpha$ , co oznacza przyjęcie hipotezy alternatywnej o braku równości median.

**Wniosek 5.9.** *Na mocy analizy POST-HOC:*

- (1) *Nie istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmu Vignetting-CT na zbiorach IMAGINE oraz Dresden Image Database;*
- (2) *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmu Lukás na zbiorach IMAGINE oraz Dresden Image Database.*

**Podsumowanie** W niniejszym podrozdziale wykazaliśmy statystycznie, że na ogół nie ma istotnych różnic w klasyfikacji algorytmów identyfikacji (zarówno proponowanych, jak i literaturowych) na proponowanym zbiorze zdjęć IMAGINE oraz bazie Dresden. Przeprowadziliśmy łącznie cztery grupy porównań (P1)-(P4) obejmujące 15 bezpośrednich porównań między, z których 12 nie wykazało podstaw do odrzucenia hipotezy o równości median. Oznacza to, że rozważane algorytmy osiągają statystycznie podobne rezultaty klasyfikacyjne na proponowanym zbiorze IMAGINE, a także na istniejącej bazie Dresden Image Database. Potwierdza to przydatność zbioru IMAGINE.

### 5.3.2 Algorytm MSE-DSI

W niniejszym badaniu porównamy statystycznie wartości TP algorytmu MSE-DSI z algorytmami z literatury: Lukás, Bondi oraz Tuama (dla aparatów cyfrowych) oraz Khanna (dla skanerów płaskich). Tym razem porównania przeprowadzimy odnosząc wyniki poszczególnych algorytmów bezpośrednio do siebie, a nie porównując wyniki określonego algorytmu na dwóch zbiorach danych, jak miało to miejsce w poprzednim podrozdziale. Przykładowo, porównamy algorytm MSE-DSI bezpośrednio z algorytmem Lukás, Bondi oraz Tuama (aparaty cyfrowe) oraz Khanna (skanery płaskie). Porównania dla aparatów cyfrowych przeprowadzimy dwa razy: tym razem hipotezy zweryfikujemy osobno dla zbioru IMAGINE oraz dla zbioru Dresden Image Database, w przeciwieństwie do poprzedniego podrozdziału.

#### Weryfikacja wyników klasyfikacji aparatów cyfrowych

##### Testy normalności rozkładu

**Hipoteza 5.6.** *Wartości wskaźników TP reprezentujących wyniki klasyfikacji algorytmów MSE-DSI, Lukás, Bondi oraz Tuama spełniają hipotezę zerową  $\mathcal{H}_0$  o zgodności z rozkładem normalnym, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku zgodności z rozkładem normalnym, zarówno dla zbioru IMAGINE, jak i Dresden.*

Tabela 5.20: Wyniki testów normalności rozkładu: algorytm MSE-DSI kontra literatura (dla zbiorów IMAGINE oraz Dresden)

Dla IMAGINE	Liczność	Statystyka $W$	Dla Dresden	Liczność	Statystyka $W$
MSE-DSI	53	0,96	MSE-DSI	48	0,96
Lukás	53	0,79	Lukás	48	0,94
Bondi	53	0,90	Bondi	48	0,94
Tuama	53	0,79	Tuama	48	0,86

*Dowód.* W tab. 5.20 przedstawiono wyniki testów normalności rozkładu Shapiro-Wilka.

Wyniki wskazują, że jedynie w przypadku algorytmu MSE-DSI spełniona jest normalność rozkładu, gdyż wartość obliczonej statystyki testowej przekracza wartość krytyczną  $W_c = 0,947$ . W przypadku algorytmów Lukás, Bondi oraz Tuama (zarówno dla zbioru IMAGINE oraz Dresden) nie ma podstaw do przyjęcia hipotezy zerowej o normalności rozkładów, gdyż obliczona wartość  $W$  jest mniejsza od wartości krytycznej. Stąd do dalszej analizy użyjemy testu nieparametrycznego.  $\square$

**Wniosek 5.10.** *Na mocy testu Shapiro-Wilka:*

- (1) *Wartości wskaźników TP reprezentujących wyniki algorytmu MSE-DSI dla zbiorów zdjęć IMAGINE oraz Dresden są zgodne z rozkładem normalnym;*
- (2) *Wartości wskaźników TP reprezentujących wyniki pozostałych algorytmów na obu zbiorach zdjęć nie są zgodne z rozkładem normalnym.*

**Test ANOVA Kruskala-Wallisa** W związku z tym, że rozważane dane w przeważającej liczbie przypadków nie są zgodne z rozkładem normalnym, korzystamy z testu ANOVA Kruskala-Wallisa.

**Hipoteza 5.7.** *Zweryfikujemy następujące hipotezy:*

- (1) *Mediany  $\theta$  wskaźników TP reprezentujących wyniki algorytmów MSE-DSI, Lukás, Bondi oraz Tuama na zbiorze IMAGINE spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości;*
- (2) *Mediany  $\theta$  wskaźników TP reprezentujących wyniki algorytmów MSE-DSI, Lukás, Bondi oraz Tuama na zbiorze Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Obliczmy statystykę  $H$ . W tym celu, wyznaczmy niezbędne dane, by móc je podstawić do wzoru 5.9. Wartości są następujące:

- Dla zbioru IMAGINE:  $N_I = \sum_{j=1}^k n_j$ , mamy  $k = 4$ ;  $n_1 = n_2 = n_3 = n_4 = 53$ , a więc uzyskujemy:  $N_I = \sum_{j=1}^4 53 = 4 \cdot 53 = 212$ ;
- Dla zbioru Dresden:  $N_D = \sum_{j=1}^k n_j$ , mamy  $k = 4$ ;  $n_1 = n_2 = n_3 = n_4 = 48$ , a więc uzyskujemy:  $N_D = \sum_{j=1}^4 48 = 4 \cdot 48 = 192$ ;
- Wyznamy sumy rang  $R_1, \dots, R_4$ . Wyniki są przedstawione w tab. 5.21.

Tabela 5.21: Sumy rang, algorytm MSE-DSI kontra literatura

Zbiór:	IMAGINE	Dresden
Algorytm	Suma rang	Suma rang
MSE-DSI	$R_1 = 1431,0$	$R_1 = 1176,0$
Lukás	$R_2 = 6198,0$	$R_2 = 4652,0$
Bondi	$R_3 = 7060,0$	$R_3 = 6386,0$
Tuama	$R_4 = 7889,0$	$R_4 = 6315,0$

Obliczmy statystykę  $H_I$  dla zbioru IMAGINE:

$$\begin{aligned}
 H_I &= \frac{12}{212(212+1)} \cdot \left( \frac{1431,0^2}{53} + \frac{6198,0^2}{53} + \frac{7060,0^2}{53} + \frac{7889,0^2}{53} \right) - 3(212+1) = \\
 &= \frac{1}{3763} \cdot 2878168 - 639 = 125,86
 \end{aligned}$$

Obliczmy statystykę  $H_D$  dla zbioru Dresden:

$$\begin{aligned}
 H_D &= \frac{12}{192(192+1)} \cdot \left( \frac{1176,0^2}{48} + \frac{4652,0^2}{48} + \frac{6386,0^2}{48} + \frac{6315,0^2}{48} \right) - 3(192+1) = \\
 &= \frac{1}{3088} \cdot 2160089 - 579 = 120,51
 \end{aligned}$$

Wartość krytyczna statystyki  $\chi^2$  z  $4 - 1 = 3$  stopniami swobody wynosi  $H_c = 7,815$ . W związku z tym, że

$$H_I > H_c,$$

oraz

$$H_D > H_c,$$

powinniśmy odrzucić hipotezę zerową o statystycznej równości median rozważanych danych, zarówno dla zbioru IMAGINE, jak i Dresden.  $\square$

**Wniosek 5.11.** *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów MSE-DSI, Lukás, Bondi oraz Tuama zarówno dla zbioru IMAGINE, jak i Dresden na mocy testu ANOVA Kruskala-Wallisa.*

Przeprowadźmy analizę POST-HOC (wyniki w tab. 5.22):

Tabela 5.22: Analiza POST-HOC: algorytm MSE-DSI kontra literatura (dla zbiorów IMAGINE oraz Dresden)

IMAGINE			Dresden		
		Wartość $p$			Wartość $p$
MSE-DSI	Lukás	0,01	MSE-DSI	Lukás	0,01
MSE-DSI	Bondi	0,01	MSE-DSI	Bondi	0,01
MSE-DSI	Tuama	0,01	MSE-DSI	Tuama	0,01

Analiza wyraźnie wskazuje, że proponowany algorytm osiąga statystyczną różnicę ze wszystkimi algorytmami z literatury dla obu zbiorów. W każdym przypadku wartość  $p$  jest mniejsza od przyjętego poziomu istotności  $\alpha$ , stąd nie powinniśmy przyjmować hipotezy zerowej o równości danych. Możemy ponadto przeanalizować wartości rang testu ANOVA Kruskala-Wallisa, co ułatwi nam precyzyjne określenie różnic w badanych danych. Wyniki przedstawia tab. 5.21. Uzyskane sumy rang jasno wskazują niższą skuteczność proponowanego algorytmu MSE-DSI w odniesieniu do metod z literatury. W ramach testu dla zbioru IMAGINE, metoda MSE-DSI osiąga sumę rang grupy 1431,0, podczas gdy dla algorytmów Lukás—Tuama jest to odpowiednio 6198,0; 7060,0 oraz 7889,0. Podobną sytuację obserwujemy przy użyciu bazy Dresden, gdzie proponowany algorytm także uzyskał najniższą wartość sumy rang (1176,0). Algorytmy literaturowe osiągają odpowiednio 4652,0; 6386,0 i 6315,0. Jeśli przyjmiemy, że najlepsze wyniki osiąga ten algorytm, dla którego suma rang jest największa, to w przypadku zbioru IMAGINE jest to metoda Tuama, natomiast dla bazy Dresden metoda Tuama nieznacznie ustępuje metodzie Bondi. Interpretacja wyników dla średnich rang grupy jest analogiczna.

**Wniosek 5.12.** *Na mocy analizy POST-HOC wartości TP proponowanego algorytmu MSE-DSI są statystycznie niższe w odniesieniu do algorytmów Lukás, Bondi oraz Tuama dla obu zbiorów zdjęć.*

**Podsumowanie** Uzyskane wyniki analizy statystycznej możemy zinterpretować następująco. Wartości TP uzyskane przez proponowany algorytm MSE-DSI są statystycznie różne od wyników uzyskanych przez cytowane algorytmy z literatury. Fakt ten został wykazany przez analizę POST-HOC.

### Weryfikacja wyników klasyfikacji skanerów płaskich

W niniejszym podrozdziale analizie poddamy wartości TP algorytmu MSE-DSI dla klasyfikacji skanerów płaskich oraz analogiczne wyniki, które uzyskał algorytm z literatury zaproponowany przez Khanna. Liczność wyników dla algorytmu MSE-DSI wynosi  $n_1 = 10$ , natomiast dla algorytmu Khanna,  $n_2 = 10$ .

### Testy normalności rozkładu

**Hipoteza 5.8.** *Wartości wskaźników TP reprezentujących wyniki klasyfikacji algorytmów MSE-DSI oraz Khanna spełniają hipotezę zerową  $\mathcal{H}_0$  o zgodności z rozkładem normalnym, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku zgodności z rozkładem normalnym, zarówno dla zbioru IMAGINE, jak i Dresden.*

*Dowód.* Wartość statystyki testowej  $W$  testu Shapiro-Wilka dla algorytmu MSE-DSI wynosi  $W = 0,85$ ; dla algorytmu Khanna  $W = 0,94$ . Wartość krytyczna Shapiro-Wilka dla prób o licznosci 10 wynosi  $W_c = 0,842$ . W związku z tym, że w przypadkach obu algorytmów obliczona wartość  $W > W_c$ , nie mamy podstaw do odrzucenia hipotezy zerowej o zgodności z rozkładem normalnym. Oznacza to, że do dalszej analizy możemy skorzystać z testów parametrycznych.  $\square$

**Wniosek 5.13.** *Wskaźniki TP reprezentujące wyniki algorytmów MSE-DSI oraz Khanna (dla skanerów płaskich) są zgodne z rozkładem normalnym na mocy testu Shapiro-Wilka.*

### Test Fishera-Snedecora

**Hipoteza 5.9.** *Wariancje  $\sigma^2$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów MSE-DSI oraz Khanna dla skanerów płaskich spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Za pomocą testu Fishera-Snedecora zweryfikujemy hipotezę o równości wariancji naszych danych. W przypadku algorytmu MSE-DSI,  $\sigma^2 = 431,34$ , natomiast dla algorytmu Khanna  $\sigma^2 = 12,54$ . Wyznamy statystykę testową:

$$F = \frac{431,34}{12,54} = 34,39$$

Wartość krytyczna rozkładu Fishera-Snedecora dla  $n_1 - 1 = 9$  oraz  $n_2 - 1 = 9$  stopni swobody (odczytana z tablic tego rozkładu) wynosi 3,18 i jest ona zauważalnie niższa od obliczonej statystyki  $F$ . Odrzucamy więc hipotezę zerową o równości wariancji. Stąd, do dalszej analizy skorzystamy z testu  $t$ -Studenta dla grup niezależnych z poprawką Cochran-Coxa.  $\square$



**Wniosek 5.14.** *Istnieje statystyczna różnica pomiędzy wariancjami  $\sigma^2$  wskaźników TP reprezentujących wyniki algorytmów MSE-DSI oraz Khanna (dla skanerów płaskich) na mocy testu Fishera-Snedecora.*

### Test $t$ -Studenta z poprawką Cochran-Coxa

**Hipoteza 5.10.** *Wartości średnie  $\bar{x}$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów MSE-DSI oraz Khanna dla skanerów płaskich spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Test  $t$ -Studenta dla grup niezależnych pozwala na weryfikację hipotezy o równości średnich arytmetycznych badanych danych. Ponieważ rozważane przez nas dane na mocy testu Fishera-Snedecora mają różne statystycznie wariancje, wykorzystamy poprawkę Cochran-Coxa. Przeprowadzimy test z dwustronnym obszarem krytycznym (ang. *two-tail test*), na co wskazuje postać hipotezy alternatywnej. Obliczmy wartość statystyki testowej  $t$ :

$$t = \frac{81,3 - 90,1}{\sqrt{\frac{431,34}{10} + \frac{12,54}{10}}} = \frac{-8,8}{6,66} = -1,32$$

Obliczmy liczbę stopni  $df$  swobody zgodnie z [131]:

$$df = \frac{\left(\frac{431,34}{10} + \frac{12,54}{10}\right)^2}{\left(\frac{431,34}{10}\right)^2 \frac{1}{10-1} + \left(\frac{12,54}{10}\right)^2 \frac{1}{10-1}} = \frac{1970,29}{206,73 + 0,17} = 9,52 \approx 10$$

Wartość krytyczna rozkładu  $t$ -Studenta na poziomie istotności  $\alpha = 0,05$  dla 10 stopni swobody wynosi 2,228. Przedział krytyczny  $C$  ma zatem postać:

$$C = (-\infty, -2,228] \cup [2,228, \infty)$$

W związku z tym, że  $t \in C$ , nie ma podstaw, by odrzucić hipotezę zerową o równości średnich arytmetycznych pomiędzy danymi.  $\square$

**Wniosek 5.15.** *Nie istnieje statystyczna różnica pomiędzy średnimi arytmetycznymi  $\bar{x}$  wskaźników TP reprezentujących wyniki algorytmów MSE-DSI oraz Khanna (dla skanerów płaskich) na mocy testu  $t$ -Studenta.*

**Podsumowanie** W niniejszym podrozdziale porównaliśmy statystycznie wartości TP algorytmów MSE-DSI oraz Khanna w identyfikacji skanerów płaskich. Za pomocą testu  $t$ -Studenta wykazaliśmy statystyczną nieistotność różnic pomiędzy średnimi arytmetycznymi rozważanych danych. Analiza więc wskazuje, że można uznać, iż proponowany algorytm MSE-DSI osiąga statystycznie porównywalną skuteczność klasyfikacji w odniesieniu do algorytmu Khanna.

### 5.3.3 Algorytm CompaRe

W niniejszym podrozdziale zweryfikujemy statystycznie wyniki klasyfikacji aparatów cyfrowych za pomocą algorytmu CompaRe, porównując je z algorytmami z literatury. Ocenie poddajemy wartości TP.

#### Eksperyment I

W ramach eksperymentu I porównamy wyniki algorytmu CompaRe z algorytmami kompaktowej reprezentacji cyfrowego odcisku palca zaproponowanymi przez Valsesia oraz Li.

#### Testy normalności rozkładu

**Hipoteza 5.11.** *Wartości wskaźników TP reprezentujących wyniki algorytmów CompaRe, Valsesia oraz Li spełniają hipotezę zerową  $\mathcal{H}_0$  o zgodności z rozkładem normalnym, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku zgodności z rozkładem normalnym, zarówno dla zbioru IMAGINE, jak i Dresden.*

*Dowód.* Wyniki testu Shapiro-Wilka dla algorytmów CompaRe, Valsesia oraz Li (zbiory IMAGINE oraz Dresden) prezentujemy w tab. 5.23.

Tabela 5.23: Wyniki testów normalności rozkładu: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden)

IMAGINE	Liczność	Statystyka $W$	Dresden	Liczność	Statystyka $W$
CompaRe	53	0,58	CompaRe	48	0,81
Valsesia	53	0,86	Valsesia	48	0,81
Li	53	0,84	Li	48	0,92

W żadnym z przypadków nie możemy przyjąć hipotezy zerowej o zgodności z rozkładem normalnym, gdyż wartość statystyki  $W$  jest mniejsza od wartości krytycznych  $W_c^{(53)} = 0,950$  oraz  $W_c^{(48)} = 0,947$ . Do dalszej analizy wykorzystamy więc test nieparametryczny.  $\square$

**Wniosek 5.16.** *Wskaźniki TP reprezentujące wyniki algorytmów CompaRe, Valsesia oraz Li dla zbiorów IMAGINE oraz Dresden nie są zgodne z rozkładem normalnym na mocy testu Shapiro-Wilka.*

**Test ANOVA Kruskala-Wallisa** W związku z tym, że rozważane dane nie są zgodne z rozkładem normalnym, korzystamy z testu ANOVA Kruskala-Wallisa.

**Hipoteza 5.12.** *Zweryfikujemy następujące hipotezy:*

- (1) *Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów CompaRe, Valsesia oraz Li na zbiorze IMAGINE spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości;*
- (2) *Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów CompaRe, Valsesia oraz Li na zbiorze Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Obliczymy statystykę  $H_I$  oraz  $H_D$ , odpowiednio dla zbiorów IMAGINE oraz Dresden. W tym celu, wyznaczmy niezbędne dane:

- Dla zbioru IMAGINE:  $N_I = \sum_{j=1}^k n_j$ , mamy  $k = 3$ ;  $n_1 = n_2 = n_3 = 53$ , a więc uzyskujemy:  $N_I = \sum_{j=1}^3 53 = 3 \cdot 53 = 159$ ;
- Dla zbioru Dresden:  $N_D = \sum_{j=1}^k n_j$ , mamy  $k = 3$ ;  $n_1 = n_2 = n_3 = 48$ , a więc uzyskujemy:  $N_D = \sum_{j=1}^3 48 = 3 \cdot 48 = 144$ ;
- Wyznamy sumy rang  $R_1, \dots, R_3$ . Wyniki są przedstawione w tab. 5.24.

Obliczymy statystykę  $H_I$  dla zbioru IMAGINE:

$$\begin{aligned} H_I &= \frac{12}{159(159+1)} \cdot \left( \frac{6953,0^2}{53} + \frac{2776,0^2}{53} + \frac{2991,0^2}{53} \right) - 3(159+1) = \\ &= \frac{1}{2120} \cdot 1226348 - 480 = 98,47 \end{aligned}$$

Obliczymy statystykę  $H_D$  dla zbioru Dresden:

$$\begin{aligned} H_D &= \frac{12}{144(144+1)} \cdot \left( \frac{5672,0^2}{48} + \frac{3021,0^2}{48} + \frac{1748,0^2}{48} \right) - 3(144+1) = \\ &= \frac{1}{1740} \cdot 924031 - 435 = 96,05 \end{aligned}$$

Tabela 5.24: Analiza rang: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden)

IMAGINE		
Algorytm	Suma rang grupy	Średnia rang grupy
CompaRe	$R_1 = 6953,0$	131,0
Valsesia	$R_2 = 2776,0$	52,0
Li	$R_3 = 2991,0$	56,0
Dresden		
Algorytm	Suma rang grupy	Średnia rang grupy
CompaRe	$R_1 = 5672,0$	118,0
Valsesia	$R_2 = 3021,0$	63,0
Li	$R_3 = 1748,0$	36,0

Wartość krytyczna statystyki  $\chi^2$  z  $3 - 1 = 2$  stopniami swobody wynosi  $H_c = 5,99$ .  
W związku z tym, że

$$H_I > H_c,$$

oraz

$$H_D > H_c,$$

odrzucaamy hipotezę zerową o statystycznej równości median rozważanych danych, zarówno dla zbioru IMAGINE, jak i Dresden.  $\square$

**Wniosek 5.17.** *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmów CompaRe, Valsesia oraz Li na zbiorach IMAGINE oraz Dresden na mocy testu ANOVA Kruskala-Wallis.*

Wyniki analizy POST-HOC prezentujemy w tab. 5.25.

Tabela 5.25: Analiza POST-HOC: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden)

IMAGINE			Dresden		
		Wartość $p$			Wartość $p$
CompaRe	Valsesia	0,01	CompaRe	Valsesia	0,01
CompaRe	Li	0,01	CompaRe	Li	0,01
Valsesia	Li	1	Valsesia	Li	0,01

Analiza wykazuje, że wartość prawdopodobieństwa  $p$  nie przekracza zadanego poziomu istotności w bezpośrednich porównaniach proponowanego algorytmu z propozycjami z literatury. Wynika z tego, że dane reprezentujące algorytm CompaRe różnią się istotnie z metodami Valsesia i Li. Brak statystycznych różnic możemy zauważyć jedynie w przypadku bezpośredniego porównania algorytmu Valsesia oraz Li dla zbioru zdjęć IMAGINE. Analizując sumy rang (tab. 5.24) zauważamy, że najwyższą wartość uzyskuje proponowany algorytm CompaRe, odpowiednio 6953 oraz 5672, zarówno dla zbiorów IMAGINE oraz Dresden. Algorytmy Valsesia oraz Li osiągnęły zauważalnie niższe wartości sumy rang (odpowiednio 2776 i 2991 dla zbioru IMAGINE oraz 3021 i 1748 dla Dresden). Analogicznie możemy zinterpretować wyniki średnich rang grupy. Na tej podstawie możemy ocenić, że najwyższą skuteczność uzyskał algorytm CompaRe i jest ona statystycznie istotna.

**Wniosek 5.18.** *Algorytm CompaRe osiąga wyższą statystycznie dokładność klasyfikacji porównując z algorytmami Valsesia oraz Li, zarówno dla zbioru IMAGINE, jak i Dresden.*

## **Eksperyment II**

W ramach eksperymentu II porównamy wyniki algorytmu CompaRe z „tradycyjnymi” algorytmami identyfikacji aparatów cyfrowych: Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson. Podkreśliśmy, że algorytmy z literatury nauczone zostały cyfrowymi odciskami palca wygenerowanymi przez algorytm CompaRe. Ze względu na duże podobieństwo, pomijamy prezentację wyników eksperymentów z propozycjami literaturowymi nauczonymi ich własnymi metodami generowania cyfrowych odcisków palca (można się jednak z nimi zapoznać w publikacji [21]).

## **Testy normalności rozkładu**

**Hipoteza 5.13.** *Wartości wskaźników TP reprezentujących wyniki klasyfikacji algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson spełniają hipotezę zerową  $\mathcal{H}_0$  o zgodności z rozkładem normalnym, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku zgodności z rozkładem normalnym, zarówno dla zbioru IMAGINE, jak i Dresden.*

*Dowód.* Wyniki testu Shapiro-Wilka dla algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson (zbiory IMAGINE oraz Dresden) prezentujemy w tab. 5.26.

W żadnym z przypadków nie możemy przyjąć  $\mathcal{H}_0$  o zgodności z rozkładem normalnym, gdyż wartość statystyki  $W$  nie przekracza wartości krytycznych, wynoszących odpowiednio  $W_c^{(53)} = 0,950$  oraz  $W_c^{(48)} = 0,947$ . W dalszej analizie skorzystamy więc z testu nieparametrycznego.  $\square$

Tabela 5.26: Wyniki testów normalności rozkładu: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden)

IMAGINE	Liczność	Statystyka $W$	Dresden	Liczność	Statystyka $W$
CompaRe	53	0,58	CompaRe	48	0,81
Lukás	53	0,80	Lukás	48	0,82
Bondi	53	0,88	Bondi	48	0,75
Tuama	53	0,89	Tuama	48	0,85
Mandelli	53	0,89	Mandelli	48	0,85
Kirchner	53	0,92	Kirchner	48	0,89

**Wniosek 5.19.** Wskaźniki  $TP$  reprezentujące wyniki algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson dla zbiorów IMAGINE oraz Dresden nie są zgodne z rozkładem normalnym na mocy testu Shapiro-Wilka.

**Test ANOVA Kruskala-Wallisa** Brak normalności rozkładu ponownie wskazuje konieczność użycia nieparametrycznego testu ANOVA Kruskala-Wallisa.

**Hipoteza 5.14.** Zweryfikujemy następujące hipotezy:

- (1) Mediany  $\theta$  wskaźników  $TP$  reprezentujących wyniki klasyfikacji algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson na zbiorze IMAGINE spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości;
- (2) Mediany  $\theta$  wskaźników  $TP$  reprezentujących wyniki klasyfikacji algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson na zbiorze Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.

*Dowód.* Obliczmy statystykę  $H_I$  oraz  $H_D$ , odpowiednio dla zbiorów IMAGINE oraz Dresden. Wyznaczmy niezbędne dane:

- Dla zbioru IMAGINE:  $N_I = \sum_{j=1}^k n_j$ , mamy  $k = 6$ ;  $n_1 = n_2 = n_3 = n_4 = n_5 = n_6 = 53$ , a więc uzyskujemy:  $N_I = \sum_{j=1}^6 53 = 6 \cdot 53 = 318$ ;
- Dla zbioru Dresden:  $N_D = \sum_{j=1}^k n_j$ , mamy  $k = 6$ ;  $n_1 = n_2 = n_3 = n_4 = n_5 = n_6 = 48$ , a więc uzyskujemy:  $N_D = \sum_{j=1}^6 48 = 6 \cdot 48 = 288$ ;
- Wyznaczmy sumy rang  $R_1, \dots, R_6$ . Wyniki są przedstawione w tab. 5.27.

Tabela 5.27: Analiza rang POST-HOC: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden)

IMAGINE		
Algorytm	Suma rang grupy	Średnia rang grupy
CompaRe	$R_1 = 13602,0$	257,0
Lukás	$R_2 = 14237,0$	269,0
Bondi	$R_3 = 7595,0$	143,0
Tuama	$R_4 = 4635,0$	87,0
Mandelli	$R_5 = 6081,0$	115,0
Kirchner	$R_6 = 4573,0$	86,0
Dresden		
Algorytm	Suma rang grupy	Średnia rang grupy
CompaRe	$R_1 = 10650,0$	222,0
Lukás	$R_2 = 11071,0$	231,0
Bondi	$R_3 = 7142,0$	149,0
Tuama	$R_4 = 5255,0$	109,0
Mandelli	$R_5 = 5864,0$	122,0
Kirchner	$R_6 = 1635,0$	34,0

Obliczmy statystykę  $H_I$  dla zbioru IMAGINE:

$$H_I = \frac{12}{318(318+1)} \cdot \left( \frac{13602,0^2}{53} + \frac{14237,0^2}{53} + \frac{7595,0^2}{53} + \frac{4635,0^2}{53} + \frac{6081,0^2}{53} + \frac{4573,0^2}{53} \right) - 3(318+1) = \frac{2}{16907} \cdot 9901221 - 957 = 214,26$$

Obliczmy statystykę  $H_D$  dla zbioru Dresden:

$$H_D = \frac{12}{218(218+1)} \cdot \left( \frac{10650,0^2}{48} + \frac{11071,0^2}{48} + \frac{7142,0^2}{48} + \frac{5255,0^2}{48} + \frac{5864,0^2}{48} + \frac{1635,0^2}{48} \right) - 3(218+1) = \frac{1}{6936} \cdot 7326509 - 867 = 189,30$$

Wartość krytyczna statystyki  $\chi^2$  z  $6 - 1 = 5$  stopniami swobody wynosi  $H_c = 11,07$ . W związku z tym, że

$$H_I > H_c,$$

oraz

$$H_D > H_c,$$

odrzucaamy hipotezę zerową o statystycznej równości median rozważanych danych, zarówno dla zbioru IMAGINE, jak i Dresden.  $\square$

**Wniosek 5.20.** *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki algorytmów CompaRe, Lukás, Bondi, Tuama, Mandelli oraz Kirchner & Johnson na zbiorach IMAGINE oraz Dresden na mocy testu ANOVA Kruskala-Wallisa.*

Do dalszej analizy wykorzystamy analizę POST-HOC. Wyniki przedstawione są w tab. 5.28.

Tabela 5.28: Analiza POST-HOC: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden)

IMAGINE			Dresden		
		Wartość $p$			Wartość $p$
CompaRe	Lukás	1	CompaRe	Lukás	1
CompaRe	Bondi	0,01	CompaRe	Bondi	0,01
CompaRe	Tuama	0,01	CompaRe	Tuama	0,01
CompaRe	Mandelli	0,01	CompaRe	Mandelli	0,01
CompaRe	Kirchner	0,01	CompaRe	Kirchner	0,01

Analiza wykazuje, że wartość prawdopodobieństwa  $p$  nie przekracza zadanego poziomu istotności w bezpośrednich porównaniach proponowanego algorytmu z niemal wszystkimi propozycjami z literatury, z wyjątkiem algorytmu Lukás. To oznacza, że dane reprezentujące algorytm CompaRe różnią się istotnie z metodami Bondi—Kirchner & Johnson, ale nie różnią się istotnie z algorytmem Lukás. Wartości rang możemy przeanalizować w tab. 5.27.

Uzyskane wyniki wskazują, że najwyższą sumę rang dla obu zbiorów zdjęć uzyskuje zarówno proponowany algorytm CompaRe, jak i algorytm Lukás. Algorytmy Bondi—Kirchner & Johnson uzyskują znacznie niższe wartości rang. Wartości sumy i średniej rang dla CompaRe oraz Lukás są bardzo zbliżone dla siebie (z niewielką przewagą dla algorytmu z literatury, choć nieistotną statystycznie). Pozostałe metody osiągają wyraźnie niższe wskaźniki, a najłabiej wypada metoda Kirchner & Johnson, dla której suma rang jest niemal trzykrotnie niższa od algorytmu CompaRe (IMAGINE) i aż sześciokrotnie w przypadku bazy Dresden. Tak więc przeprowadzona analiza wyraźnie wskazuje, że proponowany algorytm CompaRe osiąga taką samą statystycznie skuteczność klasyfikacji w odniesieniu do algorytmu Lukás oraz wyższą w odniesieniu do pozostałych metod.

**Wniosek 5.21.** *Na mocy analizy POST-HOC:*



- (1) Algorytm *CompaRe* osiąga statystycznie identyczną dokładność klasyfikacji w porównaniu do algorytmu *Lukás*, zarówno dla zbioru *IMAGINE*, jak i *Dresden*;
- (2) Algorytm *CompaRe* osiąga statystycznie wyższą dokładność klasyfikacji w porównaniu do algorytmów *Bondi*, *Tuama*, *Mandelli* oraz *Kirchner & Johnson*, zarówno dla zbioru *IMAGINE*, jak i *Dresden*.

**Podsumowanie** Przeprowadzona weryfikacja statystyczna algorytmu *CompaRe* w dwóch eksperymentach wykazała wyższą skuteczność klasyfikacji, odnosząc się do algorytmów kompaktowej reprezentacji cyfrowego odcisku palca (*Valsesia* oraz *Li*), a także porównując z algorytmami „standardowej” identyfikacji aparatów cyfrowych. Badania więc dowodzą skuteczności zaproponowanego algorytmu.

### 5.3.4 Algorytm Vignetting-CT

W niniejszym podrozdziale przeanalizujemy wartości TP uzyskane przez algorytm *Vignetting-CT* oraz *Lukás* na zbiorach *IMAGINE*, *Dresden* oraz zbiorze smartfonów.

#### Testy normalności rozkładu

**Hipoteza 5.15.** *Wartości wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás spełniają hipotezę zerową  $\mathcal{H}_0$  o zgodności z rozkładem normalnym, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku zgodności z rozkładem normalnym, zarówno dla zbioru IMAGINE, Dresden Image Database oraz zbioru smartfonów.*

*Dowód.* Wyniki testu Shapiro-Wilka są przedstawione w tab. 5.29.

Tabela 5.29: Wyniki testów normalności rozkładu: algorytm *Vignetting-CT* kontra *Lukás* (zbiory *IMAGINE*, *Dresden* oraz zb. smartfonów)

		Liczność	Statystyka $W$
IMAGINE	Vignetting-CT	53	0,94
	Lukás	53	0,77
Dresden	Vignetting-CT	48	0,94
	Lukás	48	0,94
Zb. smartfonów	Vignetting-CT	12	0,88
	Lukás	12	0,91

W przypadku zbiorów IMAGINE oraz Dresden wartości statystyki testowej  $W$  testu Shapiro-Wilka dla obu algorytmów nie przekraczają wartości krytycznych. W związku z tym powinniśmy odrzucić hipotezę zerową o zgodności z rozkładem normalnym.

W przypadku zbioru smartfonów wartość  $W$  dla algorytmu Vignetting-CT wynosi  $W = 0,88$ , natomiast dla Lukás  $W = 0,91$ . Dla grup o licznosci 12, wartość krytyczna testu wynosi  $W_c = 0,859$ . Obliczona wartość  $W$  jest większa, dlatego nie mamy podstaw do odrzucenia hipotezy zerowej, a dane możemy uznać za zgodne z rozkładem normalnym.  $\square$

**Wniosek 5.22.** *Na mocy testu Shapiro-Wilka:*

- (1) *Wskaźniki TP reprezentujące wyniki algorytmów Vignetting-CT oraz Lukás dla zbiorów zdjęć IMAGINE oraz Dresden nie są zgodne z rozkładem normalnym;*
- (2) *Wskaźniki TP reprezentujące wyniki algorytmów Vignetting-CT oraz Lukás dla zbioru smartfonów są zgodne z rozkładem normalnym.*

### Weryfikacja na zbiorach IMAGINE i Dresden

**Test U Manna-Whitneya** Jak pokazaliśmy w tab. 5.29, wyniki TP dla algorytmu Vignetting-CT oraz Lukás na zbiorach IMAGINE oraz Dresden są różne od rozkładu normalnego. Porównanie statystyczne tych danych powinniśmy więc przeprowadzić w oparciu o test nieparametryczny, dlatego proponujemy skorzystać z testu U Manna-Whitneya.

**Hipoteza 5.16.** *Zweryfikujemy następujące hipotezy:*

- (1) *Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás na zbiorze IMAGINE spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości;*
- (2) *Mediany  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás na zbiorze Dresden Image Database spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Wyznaczamy odpowiednie sumy rang, ich wartości przedstawiono w tab. 5.30. Obliczmy statystyki testowe  $U_I^{(1)}$ ;  $U_I^{(2)}$  (zbiór IMAGINE) oraz  $U_D^{(1)}$ ;  $U_D^{(2)}$  (zbiór Dresden):

$$U_I^{(1)} = 53 \cdot 53 + \frac{53(53 + 1)}{2} - 1431,0 = 2809;$$

$$U_I^{(2)} = 53 \cdot 53 + \frac{53(53 + 1)}{2} - 4240,0 = 0$$

oraz

$$U_D^{(1)} = 48 \cdot 48 + \frac{48(48 + 1)}{2} - 1176,0 = 2304;$$

$$U_D^{(2)} = 48 \cdot 48 + \frac{48(48 + 1)}{2} - 3480,0 = 0$$

Zgodnie z definicją, do analizy wybieramy mniejszą z wartości  $U_I^{(1)}$  i  $U_I^{(2)}$  oraz  $U_D^{(1)}$  i  $U_D^{(2)}$ . Ponieważ licznosc naszych danych przekracza 20 obserwacji, finalną statystykę testową  $Z_I$  (zbiór IMAGINE) oraz  $Z_D$  (zbiór Dresden) powinniśmy obliczyć ze wzoru 5.8. Podstawmy dane:

$$Z_I = \frac{0 - \frac{53 \cdot 53}{2}}{\sqrt{\frac{53 \cdot 53(53 + 53 + 1)}{12}}} = -8,87,$$

oraz

$$Z_D = \frac{0 - \frac{48 \cdot 48}{2}}{\sqrt{\frac{48 \cdot 48(48 + 48 + 1)}{12}}} = -8,44.$$

Wartość statystyki testowej w porównaniu algorytmów na zbiorze IMAGINE wynosi  $Z_I = -8,87$  oraz  $Z_D = -8,44$  dla bazy Dresden. Gdyby licznosci grup wynosiły  $n_1 = 20$  oraz  $n_2 = 40$ , wartość krytyczna  $c$  na poziomie istotności  $\alpha = 0,05$  wynosiłaby 274. W naszym przypadku  $n_1 = 53$  oraz  $n_2 = 48$  (trudno znaleźć wartość krytyczną  $c$  w dostępnych tablicach), więc można założyć, że w takim przypadku wartość krytyczna  $c$  będzie znacząco wyższa od 274. Stąd wynika, że  $Z_I < c$  oraz  $Z_D < c$ , a więc w obu przypadkach nie powinniśmy przyjmować hipotezy zerowej o równości median między testowanymi algorytmami. Aby ustalić „kierunek” różnic, możemy przeanalizować sumy oraz średnie rang. Wyniki analizy są przedstawione w tab. 5.30.

Tabela 5.30: Test U Manna-Whitneya: algorytm Vignetting-CT kontra Lukás (zbiory IMAGINE oraz Dresden)

		Suma rang grupy	Średnia rang grupy	Statystyka $U$
IMAGINE	Vignetting-CT	1431,0	27,0	2809
	Lukás	4240,0	80,0	
Dresden	Vignetting-CT	1176,0	25,0	2304
	Lukás	3480,0	73,0	

Zarówno na zbiorze IMAGINE, jak i Dresden, algorytm Vignetting-CT uzyskał zauważalnie niższe wartości sum i średnich rang. Suma rang w przypadku zbioru IMAGINE osiąga 1431,0

(4240,0 – algorytm Lukás) oraz dla Dresden 1176,0 (3480,0 – algorytm Lukás). Prowadzi to do wniosku, że proponowany algorytm Vignetting-CT osiąga statystycznie gorsze wyniki niż algorytm Lukás.  $\square$

**Wniosek 5.23.** *Na mocy testu U Manna-Whitneya:*

- (1) *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás na zbiorze IMAGINE;*
- (2) *Istnieje statystyczna różnica pomiędzy medianami  $\theta$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás na zbiorze Dresden Image Database.*

## Weryfikacja na zbiorze smartfonów

### Test Fishera-Snedecora

**Hipoteza 5.17.** *Wariancje  $\sigma^2$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás na zbiorze smartfonów spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Skorzystamy z testu Fishera-Snedecora, za pomocą którego zweryfikujemy hipotezę o równości wariancji danych reprezentujących wartości TP algorytmów Vignetting-CT oraz Lukás na zbiorze smartfonów. Dla algorytmu Vignetting-CT,  $\sigma^2 = 176,57$ , natomiast dla algorytmu Lukás  $\sigma^2 = 13,52$ . Obliczmy zatem statystykę testową:

$$F = \frac{176,57}{13,52} = 13,06$$

Wartość krytyczna rozkładu Fishera-Snedecora dla  $n_1 - 1 = 11$  oraz  $n_2 - 1 = 11$  stopni swobody wynosi 2,818, a więc jest ona mniejsza od obliczonej statystyki  $F$ . Odrzucamy więc hipotezę zerową o równości wariancji.  $\square$

**Wniosek 5.24.** *Istnieje statystyczna różnica pomiędzy wariancjami wskaźników TP reprezentującymi wyniki algorytmów Vignetting-CT oraz Lukás na mocy testu Fishera-Snedecora.*

### Test $t$ -Studenta z poprawką Cochrańa-Coxa

**Hipoteza 5.18.** *Wartości średnie  $\bar{x}$  wskaźników TP reprezentujących wyniki klasyfikacji algorytmów Vignetting-CT oraz Lukás na zbiorze smartfonów spełniają hipotezę zerową  $\mathcal{H}_0$  o ich równości, bądź hipotezę alternatywną  $\mathcal{H}_1$  o braku równości.*

*Dowód.* Wykorzystamy test  $t$ -Studenta z poprawką Cochran-Coxa, a przeprowadzony test będzie z dwustronnym obszarem krytycznym (wskazuje na to postać hipotezy alternatywnej). Obliczmy wartość statystyki  $t$ :

$$t = \frac{81,75 - 89,33}{\sqrt{\frac{176,57}{12} + \frac{13,52}{12}}} = \frac{-7,58}{3,98} = -1,90$$

Wyznamy liczbę stopni swobody  $df$ :

$$df = \frac{\left(\frac{176,57}{12} + \frac{13,52}{12}\right)^2}{\left(\frac{176,57}{12}\right)^2 \frac{1}{12-1} + \left(\frac{13,52}{12}\right)^2 \frac{1}{12-1}} = \frac{250,93}{19,68 + 0,12} = 12,67 \approx 13$$

Liczba stopni swobody wynosi  $df = 12,67 \approx 13$ . Odczytana z tablic statystycznych wartość krytyczna rozkładu  $t$ -Studenta na poziomie istotności  $\alpha = 0,05$  wynosi 2,16, przedział krytyczny  $C$  ma zatem postać:

$$C = (-\infty, -2,16] \cup [2,16, \infty)$$

Zatem  $t \in C$ . Z tego wynika, że nie ma podstaw, aby odrzucić hipotezę zerową o istotnej statystycznie różnicy średnich arytmetycznych.  $\square$

**Wniosek 5.25.** *Nie ma statystycznej różnicy pomiędzy średnimi wartościami wskaźników TP reprezentującymi wyniki algorytmów Vignetting-CT oraz Lukás na zbiorze smartfonów.*

**Podsumowanie** W niniejszym podrozdziale porównaliśmy statystycznie wartości TP algorytmów Vignetting-CT oraz Lukás. Za pomocą testu  $t$ -Studenta wykazaliśmy statystyczną nieistotność różnic pomiędzy średnimi arytmetycznymi wyników uzyskanych na zbiorze smartfonów. Test U Manna-Whitneya wykazał natomiast statystyczną różnicę wyników algorytmów na zbiorach IMAGINE oraz Dresden na niekorzyść proponowanej metody.

## 5.4 Podsumowanie

W niniejszym rozdziale zweryfikowaliśmy statystycznie proponowane algorytmy, odnosząc uzyskane wyniki do rezultatów z literatury. Analizie zostały poddane wskaźniki TP (True Positive), które są wyrażone na głównej przekątnej macierzy konfuzji. Uzyskane rezultaty zostały przeanalizowane na dwa sposoby: za pomocą statystyk opisowych oraz w oparciu o weryfikację hipotez statystycznych.

Celem statystyk opisowych było podsumowanie posiadanych danych za pomocą podstawowych wskaźników statystycznych, takich jak: średnia arytmetyczna, mediana, dominanta (moda), wariancja, odchylenie standardowe, rozstęp oraz wartość minimalna i maksymalna.

Celem weryfikacji hipotez statystycznych było ogólne porównanie wyników uzyskanych przez proponowane algorytmy z algorytmami z literatury. Wykorzystano szereg testów statystycznych, które objęły: testy normalności rozkładu metodą Shapiro-Wilka, analizę wariancji Fishera-Snedecora, analizę średnich (test  $t$ -Studenta) oraz analizę median (test U Manna-Whitneya – dla dwóch grup oraz ANOVA Kruskala-Wallisa dla większej niż dwie liczby grup).

Zarówno analiza statystyk opisowych, jak i weryfikacja hipotez doprowadziły do wniosków, że klasyfikacja za pomocą algorytmów MSE-DSI oraz Vignetting-CT osiąga nieco niższą dokładność w porównaniu do literatury. Wyniki uzyskane dla algorytmu CompaRe pokazały, że proponowana metoda osiąga nieznacznie wyższą dokładność klasyfikacji, porównując do algorytmów kompaktowej reprezentacji cyfrowego odcisku palca, a także taką samą dokładność w porównaniu do klasycznych algorytmów.

Ocenie został także poddany zbiór zdjęć IMAGINE, na którym testowane algorytmy (zarówno proponowane, jak i literaturowe) osiągnęły podobną skuteczność, odnosząc do zbioru Dresden Image Database. Możemy więc wywnioskować, że proponowany zbiór może spełniać swoją funkcję w zadaniach identyfikacji sensorów na równym poziomie, jak zbiór Dresden Image Database.

## Rozdział 6

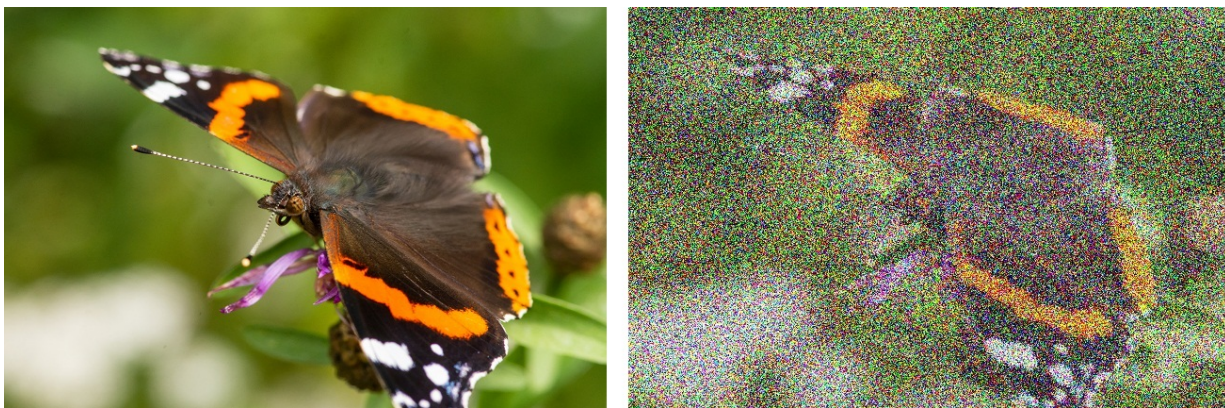
# Odporność algorytmów identyfikacji

W niniejszym rozdziale zostanie omówiona kwestia odporności algorytmów identyfikacji sensorów obrazujących. W szczególności, ocenie zostanie poddana możliwość identyfikacji aparatów cyfrowych na podstawie zdjęć, które zostały sztucznie „uszkodzone”, tj. celowo zaszumione (np. za pomocą zewnętrznego oprogramowania). Rozważane są następujące techniki: dodawanie losowych punktów do zdjęć (szum „pieprz i sól”), rozmycie zdjęcia metodą Gaussa, zaszumianie zdjęcia metodą Poissona oraz usuwanie najmniej znaczącego bitu (ang. *least significant bit*). Analizie zostanie poddany algorytm Lukás oraz konwolucyjna sieć neuronowa, której strukturę przedstawiliśmy w rozdz. 3 w podrozdz. 3.3.5. Przyjmijmy, że przez  $I$  będziemy oznaczać zdjęcie wejściowe, a przez  $I'$  rozumieć będziemy zdjęcie poddane opisywanym szumom. Ściślej, sprawdzimy, czy algorytm Lukás oraz konwolucyjna sieć neuronowa będą w stanie rozpoznać aparat cyfrowy na podstawie  $I'$ . W rozdziale oprócz opisów użytych technik zaprezentujemy jednocześnie wyniki eksperymentalne przeprowadzonej analizy. Eksperymenty zostały przeprowadzone z wykorzystaniem zbiorów zdjęć IMAGINE oraz Dresden Image Database. Pomijamy prezentację macierzy konfuzji dla algorytmu Lukás, na rzecz prezentacji wyników dla tego algorytmu w postaci analizy współczynnika korelacji  $\rho$ , którego procedura obliczania została opisana w rozdz. 4 w podrozdz. 4.2.3. Rezultaty z niniejszego rozdziału zostały opublikowane w pracach **D1** oraz **D2**.

## 6.1 Operacje na pikselach – techniki zaszumiania zdjęć

### 6.1.1 Losowe punkty na zdjęciach (szum „pieprz i sól”)

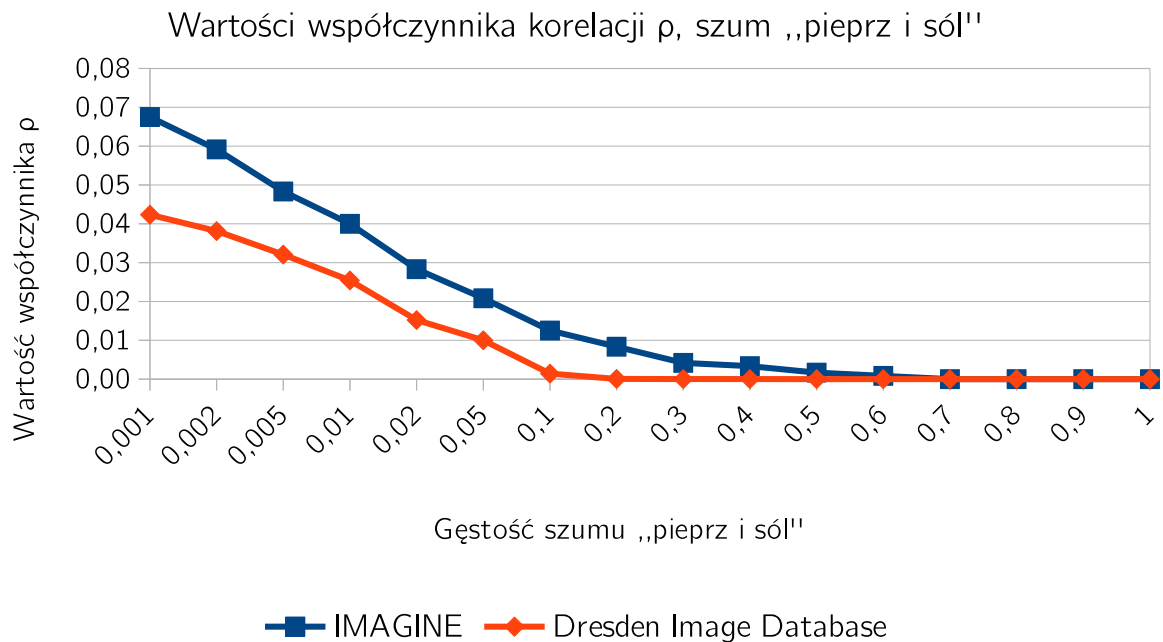
Proponujemy, aby w zdjęciu wejściowym  $I$  zamienić  $k$  pikseli w taki sposób, aby  $k/2$  pikseli otrzymało 0 kolor intensywności (zgodnie z modelem RGB), a pozostałe  $k/2$  pikseli 255 kolor intensywności w losowy sposób. Taki szum jest często w literaturze określany jako „pieprz i sól” [13, 141]. Przykład zdjęcia „normalnego” oraz odpowiadającego mu zdjęcia poddanego takiemu szumowi z  $k = 50\%$  jest przedstawiony na rys. 6.1. Aby poddać zdjęcia rozważanemu szumowi, skorzystaliśmy z implementacji funkcji `imnoise` w środowisku MATLAB z parametrem szumu  $k$  (który jest prawdopodobieństwem gęstości szumu) w zakresie od 0,001 do 1,0. Eksperymenty przeprowadzono dla zbioru Dresden Image Database oraz zbioru smartfonów. Średnie wartości  $\rho$  obliczone dla wszystkich aparatów można zobaczyć na rys. 6.2; a także w tab. 6.1 i 6.2. Tab. 6.3 przedstawia wyniki klasyfikacji dla sieci CNN (zbiór Dresden). Macierz konfuzji dla zbioru IMAGINE została przedstawiona w dodatku B w rozdz. B.4 (tab. B.24).



Rysunek 6.1: Dodanie losowych punktów do zdjęcia: zdjęcie oryginalne  $I$  (L); zdjęcie  $I'$ , na którym  $k = 50\%$  pikseli zostało zmienionych (P). Średnia intensywności pikseli w  $D = |I - I'|$  wynosi 55, odchylenie standardowe 42, a mediana 46. Aparat: Nikon D7200 (1) (zbiór IMAGINE)

Wyniki wskazują, że efektywne zaciemnienie cyfrowego odcisku palca w rozumieniu algorytmu Lukás jest możliwe, ale dopiero z prawdopodobieństwem  $k = 0,2$ . Wartości współczynnika korelacji są równe 0,00, ale jakość obrazu jest wyraźnie pogorszona, jak pokazano na rys. 6.1. Dla niższych wartości  $k$  identyfikacja nadal jest możliwa. W przypadku sieci CNN aparaty cyfrowe są rozpoznawane na podstawie zdjęć  $I'$  z dokładnością  $ACC = 99,0\%$ . Takie wyniki jasno





Rysunek 6.2: Średnie wartości współczynnika korelacji  $\rho$  (oś  $Y$ ) odnoszące się do wzrostu gęstości szumu pieprz i sól

więc wskazują, że nie można uznać efektywności rozważanej metody za satysfakcjonującą.

### 6.1.2 Rozmywanie zdjęć metodą Gaussa

Rozmycie Gaussa jest standardowym i często używanym filtrem, który jest oparty na rozkładzie normalnym z wariancją  $\sigma$  oraz średnią równą 0. Parametr  $\sigma$  określa siłę rozmycia [53].

Celem operacji jest splot jądra Gaussa z obrazem wejściowym  $I$  w celu wygenerowania zdjęcia wyjściowego  $I'$ . Filtr ten obniża kontrast na zdjęciach, dzięki czemu wyglądają one bardziej „miętko”. Przykład zdjęcia rozmytego metodą Gaussa z parametrem  $\sigma = 0,001$  przedstawiono na rys. 6.3. Eksperymenty przeprowadzono z parametrem  $\sigma$  równym  $\sigma = 0,001; 0,01; 0,1; 0,2; 0,3$ . Z rezultatów przedstawionych w tab. 6.4 oraz 6.5 wynika, że dla  $\sigma = 0,001$  identyfikacja **niektórych** aparatów w kontekście algorytmu Lukás nie jest możliwa. Zwiększanie wartości parametru  $\sigma$  nie powoduje „zmylenia” algorytmu Lukás w istotny sposób, a co więcej, jakość zdjęć nie jest już satysfakcjonująca. Aparaty, które nadal można zidentyfikować po zdjęciach rozmytych z parametrem  $\sigma = 0,3$  to: wszystkie modele Agfa, Canon IXUS 70, Kodak M1063,

Tabela 6.1: Wartości współczynnika korelacji  $\rho$  dla kolejnych gęstości  $k$  szumu pieprz i sól, zbiór IMAGINE

Nazwa urządzenia	$k =$							
	0,001	0,002	0,005	0,01	0,02	0,05	0,1	0,2 ÷ 1,0
Acer Liquid Jade S	0,08	0,06	0,03	0,02	0,01	0,01	0,00	0,00
Apple iPhone 5S	0,07	0,06	0,07	0,06	0,06	0,04	0,03	0,00
Canon SX270	0,05	0,05	0,05	0,04	0,01	0,01	0,00	0,00
DJI Spark	0,09	0,08	0,08	0,08	0,06	0,05	0,03	0,00
LG K10	0,08	0,07	0,06	0,04	0,02	0,02	0,00	0,00
Nikon D3100 (1)	0,08	0,07	0,04	0,03	0,02	0,00	0,00	0,00
Nikon D7200 (2)	0,06	0,05	0,04	0,03	0,02	0,01	0,00	0,00
Nikon P100	0,03	0,03	0,02	0,02	0,01	0,01	0,01	0,00
Samsung Galaxy Tab A	0,09	0,07	0,06	0,05	0,03	0,02	0,02	0,00
Samsung Galaxy S7	0,04	0,03	0,03	0,02	0,01	0,01	0,01	0,00
Yuneec Breeze 4K	0,07	0,07	0,04	0,03	0,04	0,02	0,01	0,00

Nikon D70, Nikon D200, Olympus 1050SW, Praktica DCZ5, Rollei RCP 7325XS oraz Samsung L74 (zbiór Dresden). Analogicznie w zbiorze IMAGINE, część urządzeń nadal jest możliwa do identyfikacji. W tab. 6.6 przedstawiono rezultaty dla sieci CNN (zbiór Dresden). Macierz konfuzji dla zbioru IMAGINE została przedstawiona w dodatku B w rozdz. B.4 (tab. B.25). Metoda rozmycia Gaussa jest zaimplementowana w środowisku MATLAB w funkcji `imnoise`.

Uzyskane wyniki jasno wskazują, że metoda rozmycia Gaussa nie może być uznana za efektywną w kontekście przeciwdziałania identyfikacji aparatów cyfrowych. Chociaż jakość obrazu jest lepsza niż w przypadku dodania szumu „pieprz i sól”, degradacja jest nadal zauważalna. Wartości współczynnika korelacji  $\rho$  dla większości wartości parametru rozmycia gaussowskiego  $\sigma$  nie są dostatecznie małe, aby można było założyć, że testowane zdjęcie nie pochodzi z badanego aparatu. Wprawdzie dla algorytmu Lukás przy  $\sigma = 0,001$  przy niektórych aparatach wartości  $\rho$  oscylują wokół zera, co wskazuje, że klasyfikator dał się „zmylić”, jednak w większości przypadków wartości  $\rho$  osiągają bądź przekraczają próg  $\rho = 0,01$ . Taka wartość jest wystarczająca, by uznać, że zdjęcie  $I'$  ma związek z danym aparatem. Analogicznie w przypadku sieci CNN; urządzenia są rozpoznawane z prawdopodobieństwem  $ACC = 99,0\%$ .

Tabela 6.2: Wartości współczynnika korelacji  $\rho$  dla kolejnych gęstości  $k$  szumu pieprz i sól, zbiór Dresden

Nazwa urządzenia	$k =$							
	0,001	0,002	0,005	0,01	0,02	0,05	0,1	0,2 ÷ 1,0
Agfa DC 733s	0,16	0,15	0,13	0,10	0,06	0,13	0,01	0,00
Agfa DC 830i	0,06	0,06	0,05	0,04	0,02	0,01	0,00	0,00
Agfa Sensor 505	0,07	0,06	0,05	0,03	0,02	0,00	0,00	0,00
Agfa Sensor 530s	0,06	0,06	0,05	0,04	0,03	0,02	0,01	0,00
Canon Ixus 55	0,03	0,02	0,02	0,02	0,01	0,00	0,00	0,00
Canon Ixus 70	0,05	0,04	0,04	0,03	0,02	0,01	0,00	0,00
Casio EX Z150	0,02	0,02	0,02	0,01	0,01	0,00	0,00	0,00
Kodak M1063	0,06	0,05	0,04	0,03	0,02	0,01	0,00	0,00
Nikon CoolPix S710	0,08	0,07	0,06	0,04	0,03	0,02	0,00	0,00
Nikon D70	0,05	0,04	0,03	0,03	0,02	0,00	0,00	0,00
Nikon D70s	0,04	0,04	0,03	0,02	0,02	0,00	0,00	0,00
Nikon D200	0,04	0,04	0,03	0,03	0,02	0,01	0,00	0,00
Olympus 1050SW	0,04	0,04	0,03	0,03	0,02	0,00	0,00	0,00
Praktica DCZ5	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Rollei RCP 7325XS	0,05	0,05	0,05	0,04	0,03	0,01	0,00	0,00
Samsung L74	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,00
Samsung NV15	0,01	0,01	0,01	0,01	0,00	0,01	0,00	0,00

### 6.1.3 Szum Poissona

Szum Poissona (zwany również szumem kwantowym) to szum zależny od sygnału, który być łatwo zauważony na obrazach. Piklele  $x$  są generowane zgodnie z rozkładem Poissona w taki sposób, że dla danego piksela wejściowego, piksel wyjściowy jest generowany z rozkładu Poissona o średniej równej wartości piksela wejściowego [49]. Przykładowy obraz demonstrujący szum Poissona można zobaczyć na rys. 6.4. Do eksperymentów wykorzystaliśmy implementację szumu Poissona w funkcji `imnoise` w oprogramowaniu MATLAB.

Wartości współczynnika korelacji dla algorytmu Lukás przedstawiono w tab. 6.7 oraz 6.8. W tab. 6.9 przedstawiono rezultaty dla sieci CNN (zbiór Dresden). Macierz konfuzji dla zbioru IMAGINE została przedstawiona w dodatku B w rozdz. B.4 (tab. B.26).

Analiza wskazuje, że zarówno algorytm Lukás, jak i sieć CNN bez większych problemów rozpoznają urządzenia na podstawie obrazów  $I'$ . Dla algorytmu Lukás wartości współczynnika

Tabela 6.3: Macierz konfuzji (zbiór Dresden), identyfikacja modelu, szum „pieprz i sól” (proponowana CNN,  $k = 0,2$ ), ACC = 99,0% (symbol \* oznacza wartości mniejsze niż 1,0%)

	Ag3	Ca1	Ca4	Ko1	Ni1	Ni2	Ni6	Ni8	N10	N11	Pr1	Ro2	Sa1
Ag3	99,0	*	*	*	*	*	*	*	*	*	*	*	*
Ca1	*	99,0	*	*	*	*	*	*	*	*	*	*	*
Ca4	*	*	99,0	*	*	*	*	*	*	*	*	*	*
Ko1	*	*	*	99,0	*	*	*	*	*	*	*	*	*
Ni1	*	*	*	*	99,0	*	*	*	*	*	*	*	*
Ni2	*	*	*	*	*	98,0	*	*	*	*	*	*	*
Ni6	*	*	*	*	*	*	99,0	*	*	*	*	*	*
Ni8	*	*	*	*	*	*	*	98,0	*	*	*	*	*
N10	*	*	*	*	*	*	*	*	99,0	*	*	*	*
N11	*	*	*	*	*	*	*	*	*	99,0	*	*	*
Pr1	*	*	*	*	*	*	*	*	*	*	99,0	*	*
Ro2	*	*	*	*	*	*	*	*	*	*	*	99,0	*
Sa1	*	*	*	*	*	*	*	*	*	*	*	*	99,0



Rysunek 6.3: Rozmywanie metodą Gaussa: zdjęcie oryginalne  $I$  (L); zdjęcie rozmyte metodą Gaussa  $I'$  (P),  $\sigma = 0,5$ . Średnia intensywności pikseli w  $D = |I - I'|$  wynosi 62, odchylenie standardowe 46, a mediana 53. Aparat: Nikon D7200 (1) (zbiór IMAGINE)

$\rho$  przekraczają próg, który uznajemy za wystarczający do powiązania aparatu ze zdjęciem; analogiczne wyniki obserwujemy w przypadku sieci CNN, dla której dokładność identyfikacji wynosi ACC = 99,0% przy obu badanych zbiorach.

Tabela 6.4: Wartości współczynnika  $\rho$  dla rozmycia Gaussa dla różnych wartości parametru  $\sigma$ , zbiór IMAGINE

Nazwa urządzenia	$\sigma =$				
	0,001	0,01	0,1	0,2	0,3
Acer Liquid Jade S	0,00	0,00	0,01	0,01	0,00
Canon SX270	0,00	0,00	0,00	0,01	0,00
DJI Spark	0,00	0,01	0,10	0,01	0,01
iPhone 5S	0,01	0,01	0,00	0,00	0,00
LG K10	0,00	0,01	0,00	0,01	0,00
Nikon D3100 (1)	0,01	0,01	0,01	0,01	0,01
Nikon D7200 (2)	0,00	0,01	0,01	0,00	0,01
Nikon P100	0,01	0,00	0,00	0,00	0,00
Samsung Galaxy Tab A	0,00	0,01	0,00	0,01	0,00
Samsung Galaxy S7	0,00	0,01	0,00	0,00	0,01
Yuneec Breeze 4K	0,00	0,01	0,01	0,00	0,01

Rysunek 6.4: Rozmywanie metodą Poissona: zdjęcie oryginalne  $I$  (L); zdjęcie rozmyte metodą Poissona  $I'$  (P). Średnia intensywności pikseli w  $D = |I - I'|$  wynosi 5, odchylenie standardowe 4, a mediana 3. Aparat: Canon PowerShot G9 X Mark II (1) (zbiór IMAGINE)

#### 6.1.4 Usuwanie najmniej znaczącego bitu

Kolejną strategią jest usunięcie z niektórych pikseli zdjęcia najmniej znaczących bitów (ang. *least significant bit* – LSB) z parametrem  $p$  określającym prawdopodobieństwo usunięcia. W zależności od liczby pikseli poddanych operacji usunięcia LSB, taka operacja może spowodować

Tabela 6.5: Wartości współczynnika  $\rho$  dla rozmycia Gaussa dla różnych wartości parametru  $\sigma$ , Dresden Image Database

Nazwa urządzenia	$\sigma =$				
	0,001	0,01	0,1	0,2	0,3
Agfa DC 733s	0,01	0,01	0,10	0,01	0,01
Agfa DC 830i	0,01	0,00	0,10	0,01	0,01
Agfa Sensor 505	0,00	0,00	0,00	0,01	0,01
Agfa Sensor 530s	0,01	0,00	0,01	0,01	0,01
Canon Ixus 55	0,00	0,00	0,01	0,01	0,00
Canon Ixus 70	0,01	0,00	0,01	0,01	0,01
Casio EX Z150	0,00	0,00	0,01	0,01	0,00
Kodak M1063	0,00	0,00	0,00	0,00	0,01
Nikon CoolPix S710	0,00	0,00	0,00	0,00	0,00
Nikon D70	0,00	0,00	0,00	0,01	0,01
Nikon D70s	0,00	0,00	0,01	0,01	0,00
Nikon D200	0,00	0,00	0,00	0,00	0,01
Olympus 1050SW	0,00	0,00	0,00	0,00	0,01
Praktica DCZ5	0,00	0,00	0,00	0,00	0,01
Rollei RCP 7325XS	0,00	0,01	0,01	0,00	0,01
Samsung L74	0,00	0,00	0,00	0,00	0,01
Samsung NV15	0,00	0,00	0,01	0,01	0,00

zmniejszenie zakresu tonalnego zdjęcia nawet o połowę. Przykład zdjęcia z usuniętym LSB dla wszystkich pikseli znajduje się na rys. 6.5.

Eksperymenty zostały przeprowadzone dla każdego piksela, z prawdopodobieństwem  $p$ , równym 0,001; 0,125; 0,25; 0,5 i 1,0. Wyniki wskazują, że usunięcie LSB nie zmniejsza w istotny sposób wartości korelacji  $\rho$  (rys. 6.6). W tab. 6.10 przedstawiono rezultaty dla sieci CNN (zbiór Dresden). Macierz konfuzji dla zbioru IMAGINE została przedstawiona w dodatku B w rozdz. B.4 (tab. B.27).

Analiza pokazuje, że usuwanie LSB nie jest optymalną strategią na usunięcie linku między zdjęciem a aparatem cyfrowym. Nawet dla  $p = 1,0$  nadal jest możliwe powiązanie zdjęcia  $I'$  z aparatem źródłowym. Wartości współczynnika korelacji  $\rho$  dla algorytmu Lukás pozostają na wysokim poziomie, a sama jakość zdjęć jest zauważalnie naruszona. Podobnie sytuacja przedstawia się dla sieci CNN, gdzie urządzenia są identyfikowane z dokładnością  $ACC = 99,0\%$ .

Tabela 6.6: Macierz konfuzji (zbiór Dresden), identyfikacja modelu, rozmycie Gaussa (propozycja CNN,  $\sigma = 0,1$ ), ACC = 99,0% (symbol \* oznacza wartości mniejsze niż 1,0%)

	Ag3	Ca1	Ca4	Ko1	Ni1	Ni2	Ni6	Ni8	N10	N11	Pr1	Ro2	Sa1
Ag3	99,0	*	*	*	*	*	*	*	*	*	*	*	*
Ca1	*	98,0	*	*	*	*	*	*	*	*	*	*	*
Ca4	*	*	99,0	*	*	*	*	*	*	*	*	*	*
Ko1	*	*	*	99,0	*	*	*	*	*	*	*	*	*
Ni1	*	*	*	*	99,0	*	*	*	*	*	*	*	*
Ni2	*	*	*	*	*	99,0	*	*	*	*	*	*	*
Ni6	*	*	*	*	*	*	99,0	*	*	*	*	*	*
Ni8	*	*	*	*	*	*	*	98,0	*	*	*	*	*
N10	*	*	*	*	*	*	*	*	98,0	*	*	*	*
N11	*	*	*	*	*	*	*	*	*	99,0	*	*	*
Pr1	*	*	*	*	*	*	*	*	*	*	99,0	*	*
Ro2	*	*	*	*	*	*	*	*	*	*	*	99,0	*
Sa1	*	*	*	*	*	*	*	*	*	*	*	*	99,0

Tabela 6.7: Wartości współczynnika  $\rho$  dla szumu Poissona, wybrane urządzenia ze zbioru IMA-GINE

Nazwa urządzenia	Wartość $\rho$
Acer Liquid Jade S	0,00
Canon SX270	0,00
DJI Spark	0,00
iPhone 5S	0,01
LG K10	0,00
Nikon D3100 (1)	0,01
Nikon D7200 (2)	0,00
Nikon P100	0,01
Samsung Galaxy Tab A	0,00
Samsung Galaxy S7	0,00
Yuneec Breeze 4K	0,00

### 6.1.5 Numeryczna analiza deformacji

Przeanalizowaliśmy podobieństwo między obrazami  $\mathbf{I}$  oraz  $\mathbf{I}'$ . Przyjmijmy, że  $|\mathbf{I} - \mathbf{I}'|$  jest macierzą *residuum* zawierającą wartości bezwzględne różnic pomiędzy intensywnościami pikseli w  $\mathbf{I}$  oraz  $\mathbf{I}'$ . Obliczyliśmy takie residua dla obrazów poddanych zaszumieniu „pieprz i sól” oraz rozmytych metodą Gaussa. Intuicyjnie, zwiększając parametr gęstości szumu, pogarsza się jakość zdjęć.

Tabela 6.8: Wartości współczynnika  $\rho$  dla szumu Poissona, wybrane urządzenia z bazy Dresden Image Database

Nazwa urządzenia	Wartość $\rho$
Agfa DC 733s	0,01
Agfa DC 830i	0,01
Agfa Sensor505	0,00
Agfa Sensor530s	0,01
Canon Ixus 55	0,01
Canon Ixus 70 (1)	0,01
Casio EX Z150 (1)	0,00
Kodak M1063 (1)	0,00
Nikon CoolPix S710 (2)	0,00
Nikon D70 (1)	0,01
Nikon D70s (2)	0,02
Nikon D200 (1)	0,00
Olympus 1050SW (0)	0,03
Praktica DCZ5 (0)	0,01
Rollei RCP 7325XS (0)	0,01
Samsung L74 (1)	0,05
Samsung NV15 (1)	0,04

Minimalne prawdopodobieństwo zaciemnienia cyfrowego odcisku palca dla algorytmu Lukás wynosi 0,2 (zbiór Dresden) oraz 0,5 (zbiór smartfonów), jednakże jakość zdjęć jest wyraźnie pogorszona. Jednocześnie, współczynnik korelacji  $\rho$  spada wówczas do 0,00. W przypadku rozmycia Gaussa, dla większości aparatów zaciemnienie cyfrowego odcisku palca może być osiągnięte dla parametru rozmycia  $\sigma = 0,001$  przy jednoczesnej relatywnie dobrej jakości obrazu  $I'$ . Szczegóły o wpływie zwiększania parametrów szumów na jakość zdjęć przedstawione są na rys. 6.7, i 6.8 oraz 6.9 i 6.10. Dodajmy, że im bliższe wartości intensywności pikseli w  $|I - I'|$  zeru, tym lepsza jest jakość obrazu  $I'$ .

Prezentowane na wykresach wyniki jasno wskazują, że w przypadku obu rozważanych szumów jakość zdjęć ulega znaczącemu pogorszeniu dla parametru, odpowiednio  $k \geq 0,1$  oraz  $\sigma \geq 0,1$ . Pomijamy prezentację wyników dla zdjęć poddanych usunięciu najmniej znaczącego bitu (LSB), gdyż jakość uzyskiwanych obrazów  $I'$  była dalece niezadowolająca, a sama identyfikacja aparatu nadal była możliwa nawet dla  $p = 1,0$ .

Przypomnijmy jednak, że w zasadzie żadna z prezentowanych technik zaszumiania obrazów



Tabela 6.9: Macierz konfuzji (zbiór Dresden), identyfikacja modelu, szum Poissona (proponowana CNN), ACC = 99,0% (symbol \* oznacza wartości mniejsze niż 1,0%)

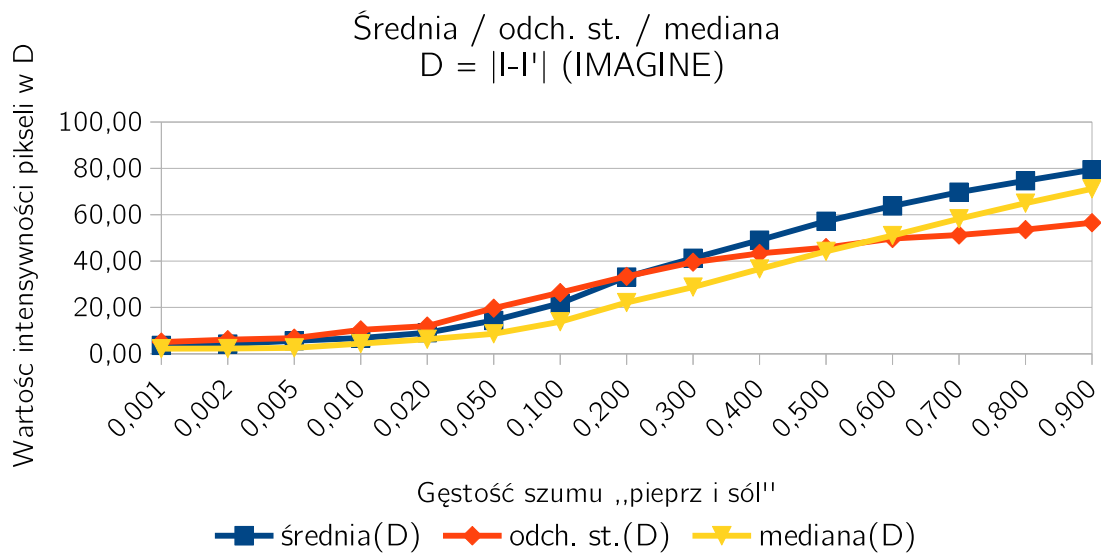
	Ag3	Ca1	Ca4	Ko1	Ni1	Ni2	Ni6	Ni8	N10	N11	Pr1	Ro2	Sa1
Ag3	99,0	*	*	*	*	*	*	*	*	*	*	*	*
Ca1	*	99,0	*	*	*	*	*	*	*	*	*	*	*
Ca4	*	*	99,0	*	*	*	*	*	*	*	*	*	*
Ko1	*	*	*	99,0	*	*	*	*	*	*	*	*	*
Ni1	*	*	*	*	99,0	*	*	*	*	*	*	*	*
Ni2	*	*	*	*	*	98,0	*	*	*	*	*	*	*
Ni6	*	*	*	*	*	*	98,0	*	*	*	*	*	*
Ni8	*	*	*	*	*	*	*	98,0	*	*	*	*	*
N10	*	*	*	*	*	*	*	*	99,0	*	*	*	*
N11	*	*	*	*	*	*	*	*	*	99,0	*	*	*
Pr1	*	*	*	*	*	*	*	*	*	*	99,0	*	*
Ro2	*	*	*	*	*	*	*	*	*	*	*	98,0	*
Sa1	*	*	*	*	*	*	*	*	*	*	*	*	99,0



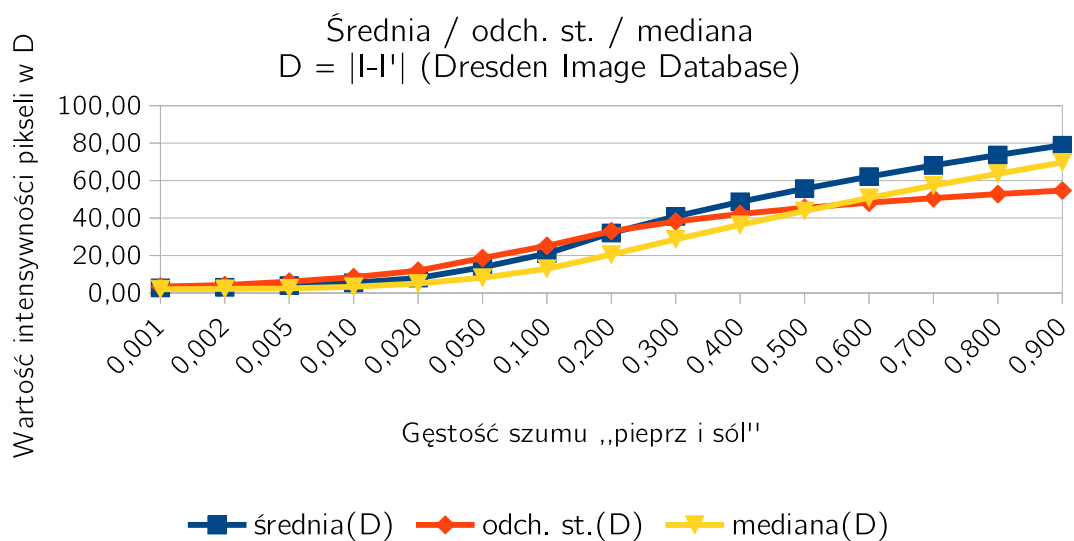
Rysunek 6.5: Usuwanie LSB: zdjęcie oryginalne **I** (L); LSB usunięte ze wszystkich pikseli **I'** (P). Średnia intensywności pikseli w  $\mathbf{D} = |\mathbf{I} - \mathbf{I}'|$  wynosi 41, odchylenie standardowe 33, a mediana 36. Aparat: Canon PowerShot G9 X Mark II (1) (zbiór IMAGINE)

nie stanowi rzetelnej „zapory” przeciwko identyfikacji aparatów, gdyż w wielu przypadkach zarówno algorytm Lukás, jak i konwolucyjna sieć neuronowa nadal były w stanie poprawnie zidentyfikować aparat na podstawie obrazu **I'**.

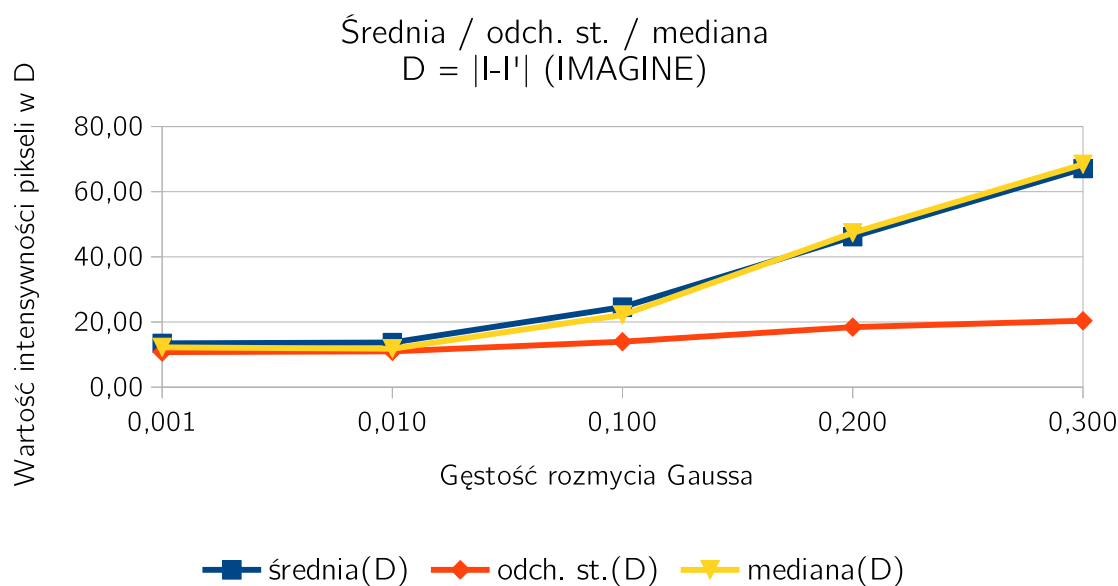




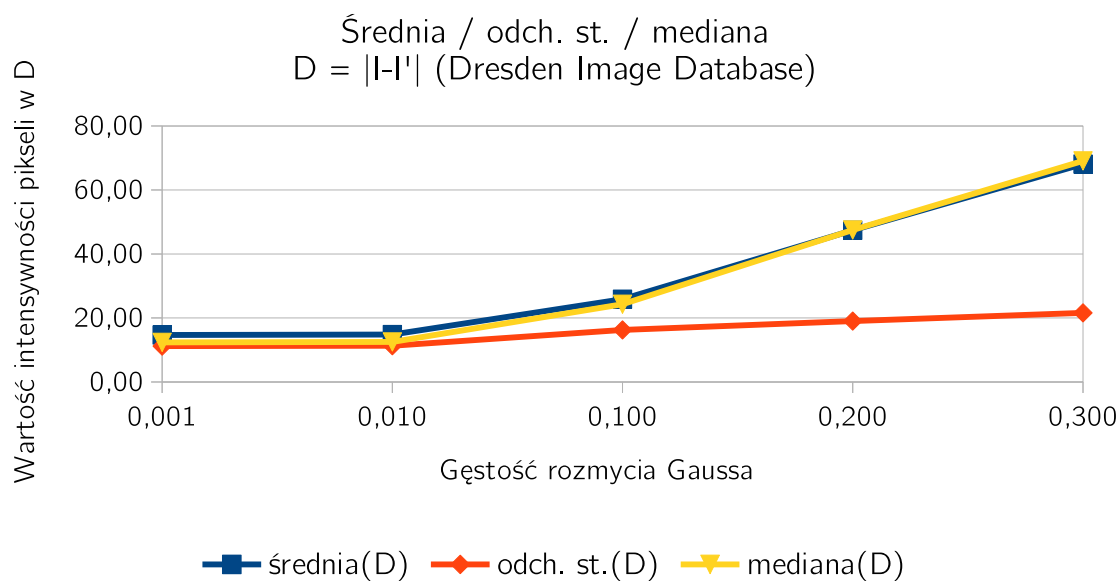
Rysunek 6.7: Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w  $D = |I - I'|$  dla szumu „pieprz i sól”, zbiór IMAGINE



Rysunek 6.8: Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w  $D = |I - I'|$  dla szumu „pieprz i sól”, zbiór Dresden



Rysunek 6.9: Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w  $D = |I - I'|$  dla rozmycia Gaussa, zbiór IMAGINE



Rysunek 6.10: Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w  $D = |I - I'|$  dla rozmycia Gaussa, zbiór Dresden

### 6.1.6 Podsumowanie

Wyniki wskazują, że rozważane proste techniki zaszumiania zdjęć wykorzystujące szum „pieprz i sól”, rozmycie Gaussa, szum Poissona oraz usunięcie najmniej znaczącego bitu (LSB) nie są wystarczające do rzetelnego zaciemnienia cyfrowego odcisku palca. Wprawdzie w niektórych przypadkach metody rozmycia Gaussa algorytm Lukás nie rozpoznawał poprawnie aparatów, to jednak w znacznej liczbie przypadków taka identyfikacja nadal jest możliwa. Aby całkowicie zaciemnić cyfrowy odcisk palca, należy poddać zdjęcia  $I$  szumom z bardzo dużymi wartościami parametrów zaszumienia.

## 6.2 Algorytm przeciwdziałania identyfikacji

### 6.2.1 Metoda przeciwdziałania identyfikacji w rozumieniu algorytmu Lukás

Rozważmy prostą procedurę, przedstawioną jako algorytm 7, która wykorzystuje algorytm Lanczosa ponownego próbkowania obrazu.

---

**Algorytm 7** Algorytm przeciwdziałania identyfikacji wykorzystujący próbkowanie Lanczosa

---

**Wejście:** Obraz (zdjęcie)  $I$  w formacie RGB o wymiarach  $M \times N$  pikseli;

**Wyjście:** Kadrowany obraz (zdjęcie)  $I'$  w formacie RGB o wymiarach  $M \times N$  pikseli.

- 1: Przytnij zdjęcie  $I$  o określoną liczbę pikseli
  - 2: Wygeneruj zdjęcie  $I'$ , rozszerzając kadrowane zdjęcie  $I$  do pierwotnych wymiarów  $M \times N$  wykorzystując algorytm Lanczos
  - 3: **return**  $I'$
- 

Zauważmy, że końcowe wymiary zdjęcia  $I'$  muszą być takie same, jak w przypadku zdjęcia  $I$ . Jest to istotne nie tylko z punktu widzenia analizy algorytmów wykorzystujących PRNU, które zazwyczaj wymagają operowania na zdjęciach o tych samych wymiarach<sup>1</sup>, ale również z praktycznego punktu widzenia. Przykładowo, adwersarz może sprawdzić, czy wymiary zdjęcia  $I'$  są zgodne z komercyjnie dostępnymi aparatami, a jeśli nie, to może to wskazywać, iż zdjęcie było poddane manipulacji.

Proponujemy wykorzystać algorytm Lanczosa ponownego próbkowania (ang. *Lanczos re-sampling*) do przeprowadzenia operacji kadrowania zdjęcia  $I$  oraz rozszerzenia zdjęcia  $I'$  do

---

<sup>1</sup>Przykładowo, algorytm Lukás wymaga do klasyfikacji obliczenia współczynnika korelacji między macierzami, co nie jest możliwe w przypadku macierzy o różnych wymiarach.

pierwotnego rozmiaru  $M \times N$  [137]. Jądro Lanczosa zostało wybrane ze względu na to, że nie powoduje ono pogorszenia jakości zdjęcia dla obserwatora, jak również nie wprowadza niepożądanych artefaktów do PRNU, które mogłyby mieć wpływ na jego jakość. Idea algorytmu ponownego próbkowania Lanczosa jest powszechnie znana, dlatego szczegóły tego algorytmu przypominamy w dodatku C do niniejszej pracy.

Kadrowanie i rozszerzenie obrazu  $I$  do pierwotnego rozmiaru jest proste i szybkie. Co więcej, przycięcie obrazu o niewielką liczbę pikseli nie doprowadza do wyraźnego obniżenia jego jakości, czy też do zakłócenia geometrii obiektów na zdjęciu. Oczywiście można jednak zauważyć, że takie zdjęcie jest nieznacznie powiększone w porównaniu do oryginału. Przykładowe zdjęcie można zobaczyć na rys. 6.11.

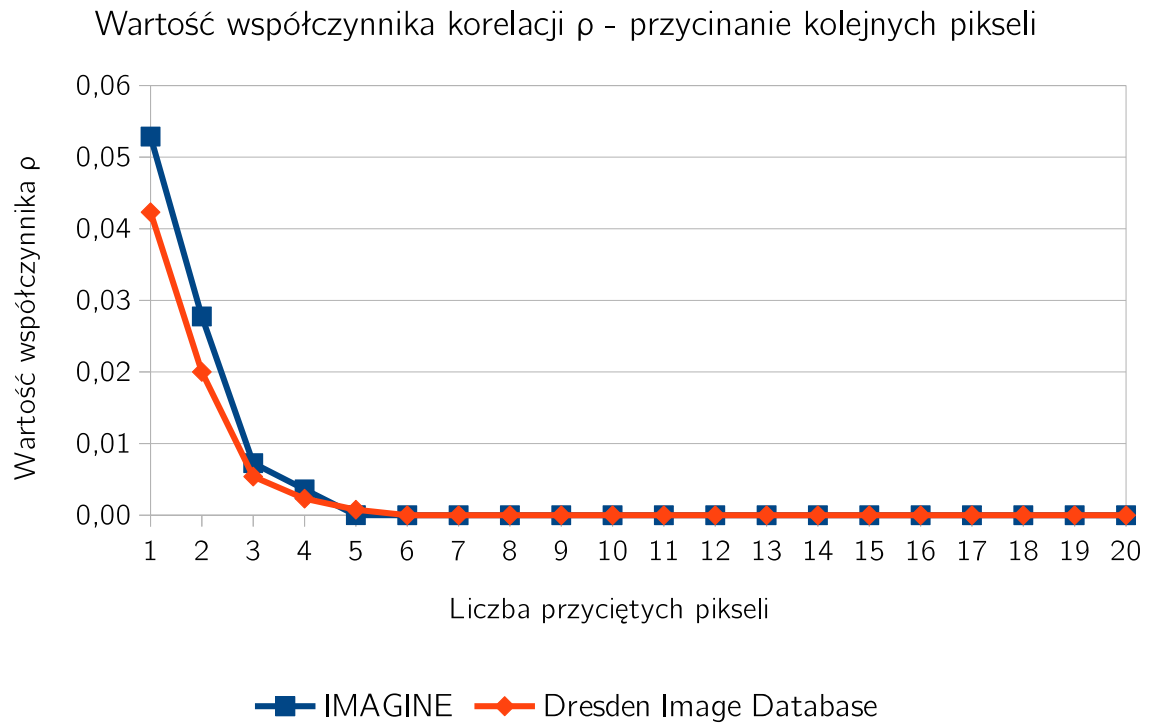


Rysunek 6.11: Po lewej: oryginalne zdjęcie  $I$  o wymiarach  $2592 \times 1944$  pikseli, w środku: zdjęcie  $I'$  przycięte łącznie o 20 pikseli (5 pikseli wzdłuż każdej krawędzi), po prawej: zdjęcie  $I'$  kadrowane i rozszerzone do oryginalnego rozmiaru. Aparat: Agfa Sensor 505

## 6.2.2 Wyniki eksperymentów

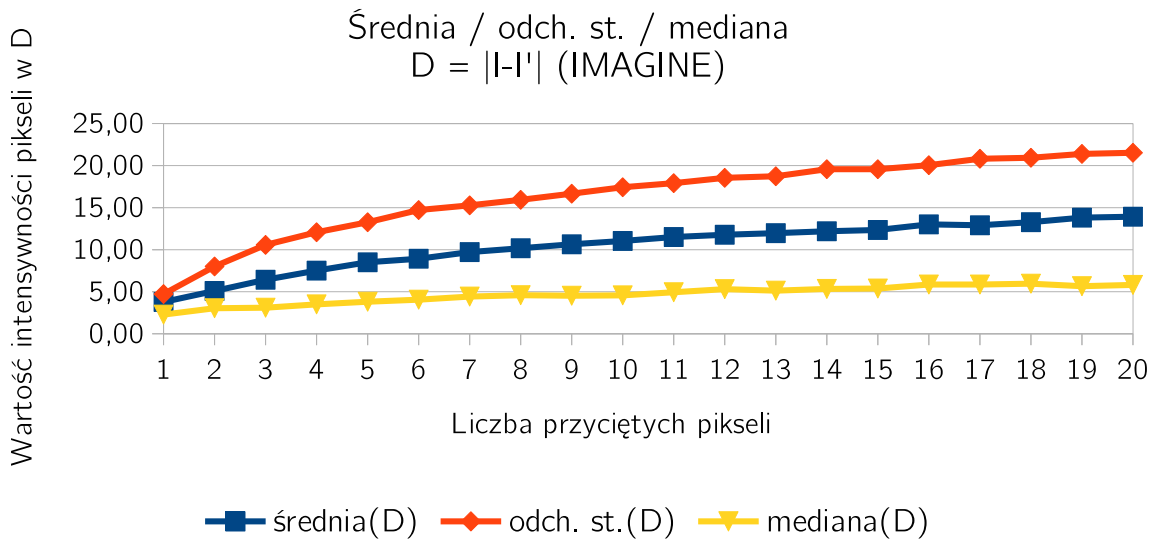
Przeprowadzono 20 eksperymentów, w ramach których usunięto symetrycznie piksele od 1 do 20 wzdłuż każdej krawędzi zdjęcia; następnie rozszerzono w ten sposób uzyskany obraz do pierwotnych wymiarów za pomocą algorytmu Lanczosa. Taka operacja oczywiście wiąże się z nieznaczną utratą informacji na zdjęciu, jednak nie dochodzi do znaczącej degradacji jakości. „Przycinając” kolejne piksele, wartości współczynnika korelacji  $\rho$  stopniowo maleją. Eksperymenty wykazały, że przycięcie każdej krawędzi zdjęcia już o 6 pikseli uniemożliwia identyfikację oryginalnego aparatu, ponieważ wartość  $\rho$  spada do 0,00. Średnie wartości współczynnika  $\rho$  z urządzeń ze wszystkich zbiorów zdjęć pokazano na rys. 6.12. Ogólne porównanie jakości obrazów pokazano na rys. 6.13 oraz 6.14.

Zaprezentowana strategia kadrowania i rozszerzania obrazu jest szybka i niemal niezauważalna dla obserwatora ze względu na niewielką utratę oryginalnych informacji na zdjęciu. Stanowi jednak bardzo skuteczną ochronę przed możliwością identyfikacji źródłowego aparatu.

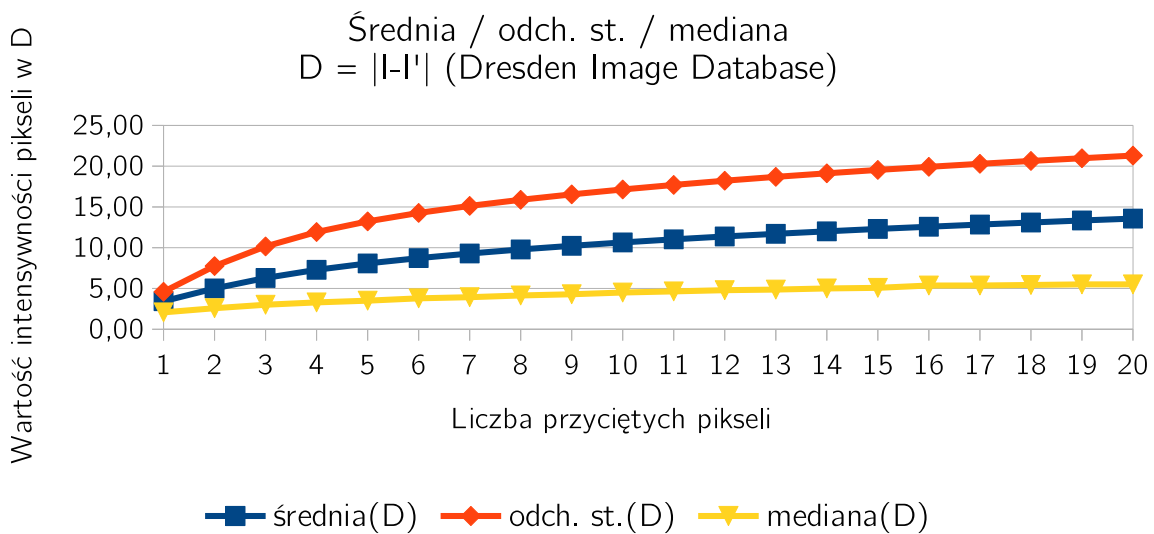


Rysunek 6.12: Wartości współczynnika  $\rho$  po kadrowaniu kolejnych pikseli (średnia dla wszystkich aparatów, zbiory IMAGINE i Dresden)

Ponadto, jest bardzo łatwa w realizacji, zarówno w przypadku „ręcznej” implementacji, jak również poprzez wykorzystanie zewnętrznego oprogramowania.



Rysunek 6.13: Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w  $D = |I - I'|$  w eksperymentach kadrowania, zbiór IMAGINE



Rysunek 6.14: Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w  $D = |I - I'|$  w eksperymentach kadrowania, zbiór Dresden



## 6.3 Podsumowanie

W niniejszym rozdziale przedyskutowano techniki zaszumiania zdjęć w celu zbadania odporności algorytmu Lukás oraz konwolucyjnej sieci neuronowej. Zaproponowano zaszumianie zdjęć za pomocą szumu „pieprz i sól”, rozmywanie zdjęć metodą Gaussa oraz Poissona, a także usuwanie najmniej znaczącego bitu (LSB). Wyniki eksperymentalne wykazały nieprzydatność wymienionych technik do skutecznego przeciwdziałania identyfikacji zarówno dla algorytmu Lukás, jak i konwolucyjnych sieci neuronowych.

Zaproponowano również prosty algorytm skutecznego przeciwdziałania identyfikacji sensorów obrazujących, wykorzystujący metodę filtrowania i ponownego próbkowania Lanczosa. Eksperymenty wykazały, że algorytm Lukás nie jest w stanie identyfikować aparatów cyfrowych na podstawie zdjęć, które zostały przycięte o 6 pikseli wzdłuż każdej krawędzi, a następnie rozszerzone za pomocą algorytmu Lanczosa do pierwotnych wymiarów.

# Rozdział 7

## Podsumowanie

W niniejszej rozprawie poruszono temat identyfikacji sensorów obrazujących (rozumianych głównie jako aparaty cyfrowe) w oparciu o analizę obrazów, które zostały za ich pomocą wygenerowane. Przeanalizowano dotychczasowe algorytmy w zakresie identyfikacji w aspekcie indywidualnego źródła (ISCI), jak również zaproponowano rozwiązania autorskie.

Zaproponowano zbiór zdjęć IMAGINE, który może posłużyć do testowania algorytmów identyfikacji w aspekcie ISCI. Zawiera on reprezentatywną liczbę nieprzetworzonych zdjęć z szeregu nowoczesnych urządzeń. Zbiór jest udostępniony na witrynie internetowej, skąd może zostać łatwo pobrany za pomocą opracowanego i udostępnionego skryptu.

Algorytm MSE-DSI umożliwia szybką identyfikację aparatów cyfrowych na podstawie niewielkich wycinków zdjęć z pojedynczego kanału kolorystycznego. Pozwala to na znaczącą redukcję liczby operacji koniecznych do nauczenia algorytmu, dzięki czemu przetwarzanie zdjęć jest zauważalnie szybsze w odniesieniu do algorytmów z literatury. Ponadto, przewagą algorytmu jest reprezentowanie cyfrowych odcisków palca w postaci liczbowej zamiast macierzowej, co znacząco ułatwia ich przechowywanie w przestrzeni dyskowej komputera.

Algorytm CompaRe umożliwia identyfikację na podstawie kompaktowej formy cyfrowego odcisku palca. W przeciwieństwie do algorytmu MSE-DSI, kompaktowy cyfrowy odcisk palca jest reprezentowany jako macierz, jednak o wymiarach znacząco zredukowanych względem rozmiaru zdjęć wejściowych. Algorytm zapewnia bardzo wysoką dokładność klasyfikacji, która na ogół jest porównywalna z istniejącymi algorytmami, a nawet w niektórych przypadkach wyższa. Co więcej, algorytm nie wymaga odszumiania zdjęć, co jest powszechną techniką. Zastosowanie algorytmu CompaRe pozwala przyspieszyć identyfikację w oparciu o istniejące algorytmy.

Algorytmy Vignetting-CT oraz Distortion-CT można wykorzystać do bardzo szybkiej identyfikacji aparatów w oparciu o zdjęcia tych samych kadrów, wykonane w tych samych warunkach

oświetleniowych. Możliwość praktycznego zastosowania tych algorytmów może być więc ograniczona, jednak mogą być one przydatne do szybkiej analizy aparatów cyfrowych pochodzących z naukowych zbiorów danych, takich jak na przykład baza Dresden Image Database.

Analiza odporności algorytmu Lukás oraz konwolucyjnych sieci neuronowych wykazała nieprzydatność prostych technik manipulacji obrazem w celu przeciwdziałania identyfikacji. Wprowadzenie do zdjęć szumu typu „pieprz i sól”, rozmywanie metodą Gaussa, Poissona, a także usuwanie najmniej znaczącego bitu nie są wystarczającymi metodami, które uniemożliwiłyby identyfikację aparatu cyfrowego na podstawie w ten sposób przekształconych zdjęć. Można jednak powstrzymać algorytm Lukás przed identyfikacją za pomocą algorytmu wykorzystującego metodę filtrowania i ponownego próbkowania Lanczosa. Proponowana metoda zakłada przycięcie obrazu o 6 pikseli wzdłuż każdej krawędzi, a następnie rozszerzenie obrazu do pierwotnych wymiarów. Algorytm jest szybki i jednocześnie bardzo skuteczny.

Skuteczność proponowanych metod została dowiedziona eksperymentalnie z reprezentatywną liczbą około 14 000 zdjęć, pochodzących z ponad stu urządzeń. Przeanalizowano czas potrzebny na nauczenie algorytmów, a uzyskane wyniki jednoznacznie potwierdziły przewagę proponowanych algorytmów nad metodami z literatury w tym zakresie. Rezultaty eksperymentów zostały także poddane gruntownej analizie statystycznej, która potwierdziła efektywność proponowanych metod.

Podsumowując, jako oryginalny wkład niniejszej rozprawy można wskazać:

- (1) Opracowanie zbioru zdjęć IMAGINE, zawierającego reprezentatywną liczbę zdjęć pochodzącą z szeregu nowoczesnych urządzeń obrazujących;
- (2) Opracowanie algorytmów MSE-DSI, CompaRe, Vignetting-CT oraz Distortion-CT, realizujących zadanie identyfikacji sensorów obrazujących na podstawie zdjęć. Algorytmy wykazują się przede wszystkim dużo krótszym czasem potrzebnym na nauczenie, w odniesieniu do algorytmów z literatury;
- (3) Zbadanie odporności algorytmu Lukás oraz konwolucyjnych sieci neuronowych w ramach identyfikacji na podstawie zdjęć poddanych różnorodnym operacjom zaszumienia;
- (4) Opracowanie metody przeciwdziałania identyfikacji w rozumieniu algorytmu Lukás, wykorzystującej próbkowanie Lanczosa.

Ponadto, istotnym wkładem jest przeprowadzenie reprezentatywnych eksperymentów zarówno na proponowanym zbiorze IMAGINE, jak i na znanej z literatury bazie zdjęć Dresden Image Database, które potwierdziły skuteczność proponowanych metod.

Pomimo znacznego zainteresowania badanym tematem w literaturze, wiele problemów pozostaje nadal nierozwiązanych. Najbardziej podstawowym wydaje się opracowanie algorytmu identyfikacji sensorów w czasie rzeczywistym. Dotychczasowe rozwiązania, oparte głównie na odsumianiu, nie są wystarczające w tym zakresie. Wprawdzie jak wynika z literatury, dokładność identyfikacji jest niemal doskonała (co również pokazaliśmy w niniejszej rozprawie), jednak czas uczenia algorytmów jest nieakceptowalnie długi. Dlatego więc opracowanie nowych algorytmów, znacząco redukujących czas potrzebny do nauki algorytmów, wydaje się naturalnym kierunkiem.

Obecnie standardem w dziedzinie widzenia komputerowego jest uczenie głębokie, do którego zaliczamy również konwolucyjne sieci neuronowe. Mogą one z powodzeniem rozwiązywać zadania identyfikacji sensorów, co jest doskonale widoczne od lat w literaturze. Niemniej można zaobserwować, że nieco mniejszym zainteresowaniem w zakresie identyfikacji sensorów cieszą się konwolucyjne autoenkodery. Dlatego ciekawym kierunkiem wydaje się opracowanie konwolucyjnego autoenkodera, który mógłby posłużyć do wygenerowania kompaktowej reprezentacji cyfrowego odcisku palca. Rozważana architektura zakładałaby wykorzystanie jedynie „pierwszej części” konwolucyjnego autoenkodera (bez dekodera). Takie rozwiązanie mogłoby pobrane na wejście zdjęcie I „sprowadzić” do rozmiaru wektora w ramach tzw. warstwy ukrytej (ang. *latent space*). Wygenerowany w ten sposób wektor mógłby zostać wykorzystany do identyfikacji aparatu i służyć jako cyfrowy odcisk palca. Być może byłaby to łatwa i niezawodna metoda generowania cyfrowych odcisków palca w formie kompaktowej.

## Dodatek A

# Algorytm Lukás – wartości współczynnika korelacji $\rho$

W tab. A.1 oraz A.2 prezentujemy przykładowe wartości współczynnika korelacji  $\rho$  w odniesieniu do PRNU z aparatów Canon EOS RP i Nikon CoolPix S710 (3) (odpowiednio zbiór IMAGINE oraz Dresden). W przypadku zdjęć wykonanych tymi urządzeniami, widać wyraźnie większe wartości współczynnika  $\rho$ .

Tabela A.1: Przykładowe wartości współczynnika  $\rho$ , zbiór IMAGINE

PRNU	„Nowe” zdjęcie	Wartość $\rho$
Canon EOS RP PRNU	Canon 1D X Mark II 8.jpg	0,0056
Canon EOS RP PRNU	Canon 5D Mark IV 8.jpg	0,0029
Canon EOS RP PRNU	Canon 5D Mark IV 14.jpg	0,0008
Canon EOS RP PRNU	Canon 6D Mark II 4.jpg	0,035
Canon EOS RP PRNU	Canon 6D Mark II 6.jpg	0,0056
Canon EOS RP PRNU	Canon 90D 3.jpg	0,0033
Canon EOS RP PRNU	Canon EOS R 13.jpg	0,0015
Canon EOS RP PRNU	Canon EOS RP 3.jpg	0,5555
Canon EOS RP PRNU	Canon EOS RP 4.jpg	0,7061
Canon EOS RP PRNU	Canon EOS RP 5.jpg	0,6701
Canon EOS RP PRNU	Canon EOS RP 6.jpg	0,626
Canon EOS RP PRNU	Canon EOS RP 7.jpg	0,5875
Canon EOS RP PRNU	Canon EOS RP 8.jpg	0,6058
Canon EOS RP PRNU	Canon EOS RP 9.jpg	0,6644

Kontynuacja na następnej stronie

Tabela A.1 – Kontynuacja z poprzedniej strony

Canon EOS RP PRNU	Canon EOS RP 10.jpg	0,6293
Canon EOS RP PRNU	Canon EOS RP 11.jpg	0,6091
Canon EOS RP PRNU	Canon EOS RP 12.jpg	0,6141
Canon EOS RP PRNU	Canon EOS RP 13.jpg	0,6901
Canon EOS RP PRNU	Canon M100 22.jpg	0,0015
Canon EOS RP PRNU	Canon M100 26.jpg	0,0048
Canon EOS RP PRNU	Fujifilm X T200 11.jpg	0,0099
Canon EOS RP PRNU	Fujifilm X T200 13.jpg	0,0083
Canon EOS RP PRNU	Nikon D5 16.jpg	0,0041
Canon EOS RP PRNU	Nikon D5 17.jpg	0,0075
Canon EOS RP PRNU	Nikon D6 10.jpg	0,0043
Canon EOS RP PRNU	Nikon D500 28.jpg	0,0162
Canon EOS RP PRNU	Nikon D500 29.jpg	0,0058
Canon EOS RP PRNU	Nikon D500 30.jpg	0,0062
Canon EOS RP PRNU	Nikon D610 4.jpg	0,0087
Canon EOS RP PRNU	Nikon D610 5.jpg	0,0094
Canon EOS RP PRNU	Nikon D850 16.jpg	0,0074
Canon EOS RP PRNU	Nikon D3100 1 16.jpg	0,0097
Canon EOS RP PRNU	Nikon D3100 1 17.jpg	0,0046
Canon EOS RP PRNU	Nikon D7200 1 8.jpg	0,0067
Canon EOS RP PRNU	Nikon D7200 1 11.jpg	0,0068
Canon EOS RP PRNU	Nikon D7200 1 13.jpg	0,0031
Canon EOS RP PRNU	Nikon Z7 35.jpg	0,0003
Canon EOS RP PRNU	Sony A9 10.jpg	0,0037
Canon EOS RP PRNU	Sony A9 15.jpg	0,0041
Canon EOS RP PRNU	Sony A9 PRNU	0,0056

Tabela A.2: Przykładowe wartości współczynnika  $\rho$ , zbiór Dresden Image Database

PRNU	„Nowe” zdjęcie	Wartość $\rho$
Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (0) 3.jpg	0,0081
Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (0) 5.jpg	0,0057
Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (0) 6.jpg	0,008
Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (0) 7.jpg	0,002

Kontynuacja na następnej stronie

Tabela A.2 – Kontynuacja z poprzedniej strony

Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (0) 8.jpg	0,0009
Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (0) 9.jpg	0,0019
Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (0) 10.jpg	0,0017
Nikon CoolPix S710 (3) PRNU	Canon Ixus 70 (1) 2.jpg	0,0011
Nikon CoolPix S710 (3) PRNU	Kodak M1063 (2) 5.jpg	0,0013
Nikon CoolPix S710 (3) PRNU	Kodak M1063 (2) 6.jpg	0,0078
Nikon CoolPix S710 (3) PRNU	Kodak M1063 (2) 7.jpg	0,0058
Nikon CoolPix S710 (3) PRNU	Kodak M1063 (2) 8.jpg	0,007
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 1.jpg	0,0241
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 2.jpg	0,0175
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 3.jpg	0,0228
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 4.jpg	0,3022
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 5.jpg	0,26
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 6.jpg	0,8636
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 7.jpg	0,1022
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 8.jpg	0,1157
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 9.jpg	0,2033
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 10.jpg	0,005
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 11.jpg	0,2977
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 12.jpg	0,3239
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 13.jpg	0,8898
Nikon CoolPix S710 (3) PRNU	Nikon CoolPix S710 (3) 14.jpg	0,0179
Nikon CoolPix S710 (3) PRNU	Samsung NV15 (2) 1.jpg	0,0082
Nikon CoolPix S710 (3) PRNU	Samsung NV15 (2) 2.jpg	0,006
Nikon CoolPix S710 (3) PRNU	Samsung NV15 (2) 3.jpg	0,0078

## Dodatek B

### Wybrane macierze konfuzji

Dla przejrzystości, w niniejszym rozdziale prezentujemy macierze konfuzji przedstawiające wyniki klasyfikacji dla algorytmów MSE-DSI oraz CompaRe.

#### B.1 Algorytm MSE-DSI

Tabela B.1: Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Lukás), ACC = 95,0% (symbol \* oznacza wartości mniejsze niż 5,0%)

	Ca1	Ca2	C17	C19	Fu1	Ni2	Ni3	Ni5	Ni9	N19	N21	So1	So2
Ca1	96,0	*	*	*	*	*	*	*	*	*	*	*	*
Ca2	*	96,0	*	*	*	*	*	*	*	*	*	*	*
C17	*	*	96,0	*	*	*	*	*	*	*	*	*	*
C19	*	*	*	94,0	*	*	*	*	*	*	*	*	*
Fu1	*	*	*	*	94,0	*	*	*	*	*	*	*	*
Ni2	*	*	*	*	*	95,0	*	*	*	*	*	*	*
Ni3	*	*	*	*	*	*	95,0	*	*	*	*	*	*
Ni5	*	*	*	*	*	*	*	95,0	*	*	*	*	*
Ni9	*	*	*	*	*	*	*	*	96,0	*	*	*	*
N19	*	*	*	*	*	*	*	*	*	95,0	*	*	*
N21	*	*	*	*	*	*	*	*	*	*	96,0	*	*
So1	*	*	*	*	*	*	*	*	*	*	*	95,0	*
So2	*	*	*	*	*	*	*	*	*	*	*	*	95,0



































## Dodatek C

# Algorytm Lanczosa próbkowania obrazu

Próbkowanie obrazu w oparciu o algorytm Lanczosa (ang. *Lanczos resampling*) jest często używane jako część przetwarzania wstępnego w celu dostosowania stosunku wysokości do szerokości lub po prostu rozmiaru obrazu. Zarówno próbkowanie obrazu w górę (ang. *upsampling*), jak i w dół (ang. *downsampling*) skutkuje redukcją oryginalnych informacji (np. uśrednianie wartości intensywności), a także wprowadzeniem pewnych „nowych informacji” (artefaktów), specyficznych dla używanej metody. Wytworzone w ten sposób artefakty niekoniecznie muszą być niepożądanym zjawiskiem, ponieważ artefakty powstałe w wyniku zastosowania algorytmu ponownego próbkowania Lanczosa [146] powodują poprawę postrzeganej ostrości obrazu, pomimo dodania szumu, który może utrudniać prawidłową analizę natężenia lub gradientu [146].

W niniejszej pracy wykorzystujemy występowanie artefaktów podczas ponownego próbkowania obrazu w celu zachowania prywatności i ukrycia cech charakterystycznych dla aparatu cyfrowego.

Próbkowanie Lanczosa jest przybliżeniem filtra sinc (łac. *sinus cardinalis* – funkcja interpolująca). Mianowicie jądro, które jest konwolucją z wartościami intensywności obrazu  $I$ , jest zdefiniowane jako funkcja

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \quad (\text{C.1})$$

pomnożona przez okno Lanczosa o szerokości  $a$ . Dla dodatniego  $a$  jądro  $L(x)$  ma ekstrema lokalne  $2a - 1$  oraz globalne dla  $x = 0$ . Poniżej przypominamy wzór opisujący jądro:

$$L(x) = \begin{cases} 1 & \text{dla } x = 0 \\ \frac{a \sin \pi x \sin \frac{\pi x}{a}}{\pi^2 x^2} & \text{dla } -a \leq x < a \\ 0 & \text{w pozostałych przypadkach} \end{cases} \quad (\text{C.2})$$

Podczas ponownego próbkowania próbki sygnału są interpolowane przez splot z jądrem, co



daje w wyniku [29]:

$$\mathbf{S}(x) = \sum_{i=\lfloor x \rfloor - a + 1}^{\lfloor x \rfloor + a} s_i \mathbf{L}(x - i) \quad (\text{C.3})$$

Ze względu na to, że wartości  $a$  wynoszące 2 lub 3 są najczęściej używane zarówno w przetwarzaniu sygnału, jak i komercyjnych edytorach obrazów, a  $a = 2$  skutkuje najniższym postrzeganym pogorszeniem wizualnym oryginalnego obrazu, w eksperymentach zdecydowaliśmy się użyć właśnie  $a = 2$ . Zauważmy, że ze względu na to, że maksimum jądra jest równe 1, wartość intensywności pikseli nigdy nie może wzrosnąć powyżej 255 dla kanału 8-bitowego, jednak dla  $|x| = 1,29671$  istnieją lokalne minima na poziomie około  $-0,086$ , co może sprowadzić wynik do ujemnej intensywności 22, stąd końcowa wartość  $\mathbf{S}(x)$  powinna zostać znormalizowana do zakresu  $[0, 255]$ .

# Spis rysunków

1.1	Schemat odszumiania obrazów i klasyfikacja nowego obrazu w oparciu o algorytm Lukás [104] . . . . .	7
1.2	Klasyfikacja szumów występujących w sensorach obrazujących. Szum generowany przez sensor składa się z deterministycznego szumu wzorcowego (ang. <i>pattern noise</i> ) oraz niedeterministycznego szumu śrutowego (ang. <i>shot noise</i> ). Szum wzorcowy natomiast tworzą stały szum wzorcowy (ang. <i>fixed pattern noise</i> – FPN) oraz szum niejednorodności odpowiedzi fotoelektrycznej (ang. <i>photo-response nonuniformity noise</i> – PRNU) [15] . . . . .	9
2.1	Przykładowe zdjęcia pochodzące ze zbioru IMAGINE . . . . .	24
2.2	Strona główna zbioru danych IMAGINE . . . . .	25
3.1	Przykładowy obraz <b>I</b> (po lewej) i jego residuum <b>S</b> (po prawej) obliczone jako bezwzględna różnica $\mathbf{S} =  \mathbf{I} - F(\mathbf{I}^{(R)}) $ . Średnia wartość intensywności pikseli w <b>S</b> wynosi 0,6, wartości intensywności w narożnikach <b>S</b> wynoszą: $\hat{s}_{a_1} = 0,36$ , $\hat{s}_{a_2} = 0,49$ , $\hat{s}_{a_3} = 0,34$ , $\hat{s}_{a_4} = 0,54$ . . . . .	42
3.2	Zdjęcie obiektu z widoczną dystorsją beczkową ( $k = -0,012$ ), niektóre piksele obarczone dystorsją są zaznaczone kolorem czerwonym (wyznaczono za pomocą oprogramowania Hugin [5]) . . . . .	47
3.3	Zdjęcie, na którym dystorsja nie występuje . . . . .	47
4.1	Dystorsja na podstawie zdjęć tego samego kadru z różnych urządzeń (przykładowe linie obarczone dystorsją oznaczono kolorem czerwonym). Zdjęcie po lewej: Huawei P8, parametr dystorsji $k = -0,6782$ ; zdjęcie po prawej: Samsung Galaxy S6, parametr dystorsji $k = -0,2$ . . . . .	68

4.2	Dystorsja na podstawie zdjęć tego samego kadru z różnych urządzeń (przykładowe linie obarczone dystorsją oznaczono kolorem czerwonym). Zdjęcie po lewej: Nikon CoolPix S710 (2), parametr dystorsji $k = -0,10323$ ; zdjęcie po prawej: Nikon CoolPix S710 (1), parametr dystorsji $k = -0,01799$ . . . . .	68
4.3	Porównanie czasu uczenia algorytmów, identyfikacja aparatów cyfrowych (zbiór IMAGINE) . . . . .	72
4.4	Porównanie czasu uczenia algorytmów, identyfikacja aparatów cyfrowych (zbiór Dresden Image Database) . . . . .	73
4.5	Porównanie czasu uczenia algorytmów, identyfikacja skanerów płaskich . . . . .	74
4.6	Czas potrzebny na wygenerowanie cyfrowych odcisków palca . . . . .	75
4.7	Czas potrzebny na uczenie sieci konwolucyjnych dla pojedynczej epoki (wyniki uśrednione dla zbiorów IMAGINE oraz Dresden) . . . . .	76
4.8	Czas potrzebny na uczenie algorytmu Lukás (wyniki uśrednione dla obu zbiorów zdjęć) . . . . .	77
4.9	Porównanie rozmiarów cyfrowych odcisków palca zapisanych jako pliki *.mat . . . . .	78
4.10	Porównanie czasów uczenia algorytmów . . . . .	80
4.11	Dokładność treningu (50 epok) – porównanie funkcji aktywacji ReLU, Leaky ReLU oraz SELU – proponowana sieć CNN . . . . .	82
4.12	Dokładność treningu (50 epok) – porównanie funkcji aktywacji ReLU, Leaky ReLU oraz SELU – sieć Bondi . . . . .	82
4.13	Dokładność treningu (50 epok) – porównanie funkcji aktywacji ReLU, Leaky ReLU oraz SELU – sieć Tuama . . . . .	83
5.1	Procedura weryfikacji statystycznej uzyskanych wyników . . . . .	89
6.1	Dodanie losowych punktów do zdjęcia: zdjęcie oryginalne $I$ (L); zdjęcie $I'$ , na którym $k = 50\%$ pikseli zostało zmienionych (P). Średnia intensywności pikseli w $D =  I - I' $ wynosi 55, odchylenie standardowe 42, a mediana 46. Aparat: Nikon D7200 (1) (zbiór IMAGINE) . . . . .	128
6.2	Średnie wartości współczynnika korelacji $\rho$ (oś Y) odnoszące się do wzrostu gęstości szumu pieprz i sól . . . . .	129
6.3	Rozmywanie metodą Gaussa: zdjęcie oryginalne $I$ (L); zdjęcie rozmyte metodą Gaussa $I'$ (P), $\sigma = 0,5$ . Średnia intensywności pikseli w $D =  I - I' $ wynosi 62, odchylenie standardowe 46, a mediana 53. Aparat: Nikon D7200 (1) (zbiór IMAGINE) . . . . .	132

6.4	Rozmywanie metodą Poissona: zdjęcie oryginalne $I$ (L); zdjęcie rozmyte metodą Poissona $I'$ (P). Średnia intensywności pikseli w $D =  I - I' $ wynosi 5, odchylenie standardowe 4, a mediana 3. Aparat: Canon PowerShot G9 X Mark II (1) (zbiór IMAGINE) . . . . .	133
6.5	Usuwanie LSB: zdjęcie oryginalne $I$ (L); LSB usunięte ze wszystkich pikseli $I'$ (P). Średnia intensywności pikseli w $D =  I - I' $ wynosi 41, odchylenie standardowe 33, a mediana 36. Aparat: Canon PowerShot G9 X Mark II (1) (zbiór IMAGINE) . . . . .	137
6.6	Wpływ zwiększania parametru $p$ na współczynnik $\rho$ . . . . .	138
6.7	Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w $D =  I - I' $ dla szumu „pieprz i sól”, zbiór IMAGINE . . . . .	139
6.8	Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w $D =  I - I' $ dla szumu „pieprz i sól”, zbiór Dresden . . . . .	139
6.9	Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w $D =  I - I' $ dla rozmycia Gaussa, zbiór IMAGINE . . . . .	140
6.10	Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w $D =  I - I' $ dla rozmycia Gaussa, zbiór Dresden . . . . .	140
6.11	Po lewej: oryginalne zdjęcie $I$ o wymiarach $2592 \times 1944$ pikseli, w środku: zdjęcie $I'$ przycięte łącznie o 20 pikseli (5 pikseli wzdłuż każdej krawędzi), po prawej: zdjęcie $I'$ kadrowane i rozszerzone do oryginalnego rozmiaru. Aparat: Agfa Sensor 505 . . . . .	142
6.12	Wartości współczynnika $\rho$ po kadrowaniu kolejnych pikseli (średnia dla wszystkich aparatów, zbiory IMAGINE i Dresden) . . . . .	143
6.13	Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w $D =  I - I' $ w eksperymentach kadrowania, zbiór IMAGINE . . . . .	144
6.14	Uśredniona różnica, odchylenie standardowe i mediana intensywności pikseli w $D =  I - I' $ w eksperymentach kadrowania, zbiór Dresden . . . . .	144

# Spis tabel

2.1	Rozmiary sensorów (niektóre sensory oznaczone symbolami FF oraz FE mają wymiary nieznacznie inne niż wymienione w tabeli) . . . . .	21
2.2	Wykorzystane urządzenia w zbiorze IMAGINE (MILC – aparat bezlusterkowy, DSLR – lustrzanka cyfrowa). Symbol * oznacza brak oficjalnej informacji od producenta o fizycznym rozmiarze sensora (można podejrzewać, że jest on podobny do rozmiaru 1/4,0"). Rozdzielczość zdjęć zdefiniowana jest w pikselach i oznacza maksymalny rozmiar zdjęcia, jaki można uzyskać z określonego aparatu	22
3.1	Przykładowe wartości MSE-DSI (po obliczeniu średniej dla 45 urządzeń) dla niektórych urządzeń ze zbioru IMAGINE . . . . .	36
3.2	Przykładowe aparaty wraz z wymiarami obrazów (w pikselach) . . . . .	39
3.3	Przykładowe wartości intensywności pikseli dla aparatu Nikon D70s (1) w residuum <b>S</b> . Im wartość jest bliższa 0, tym ciemniejszy jest obszar zdjęcia . . . . .	43
3.4	Wymiary obrazów (w pikselach), którymi uczone są algorytmy Lukás oraz Vignetting-CT . . . . .	44
4.1	Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm MSE-DSI), ACC = 73,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	58
4.2	Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (sieć konwolucyjna Bondi), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	58
4.3	Poprawność klasyfikacji ACC [%] obliczona na podstawie klasyfikacji wszystkich urządzeń . . . . .	59
4.4	Macierz konfuzji, identyfikacja modelu (algorytm MSE-DSI), ACC = 82,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	60
4.5	Macierz konfuzji, identyfikacja modelu (algorytm Khanna), ACC = 91,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	61

4.6	Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm CompaRe), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	63
4.7	[Eksperyment I] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Valsesia), ACC = 93,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	63
4.8	[Eksperyment II] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Lukás), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	64
4.9	Dokładność ACC [%] obliczona na podstawie klasyfikacji wszystkich urządzeń. ACC <sup>1</sup> oznacza dokładność algorytmów uczonych cyfrowymi odciskami palca wygenerowanymi przez algorytm CompaRe; ACC <sup>2</sup> oznacza dokładność algorytmów za pomocą cyfrowych odcisków palca wygenerowanych własnymi procedurami . . . . .	65
4.10	Macierz konfuzji (zbiór smartfonów), identyfikacja modelu (algorytm Vignetting-CT), ACC = 82,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	66
4.11	Macierz konfuzji (zbiór smartfonów), identyfikacja modelu (algorytm Lukás), ACC = 89,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	66
4.12	Dokładność ACC [%] obliczona na podstawie klasyfikacji wszystkich urządzeń . . . . .	67
4.13	Wartości parametru dystorsji $k$ na podstawie zdjęć tego samego kadru (zbiór smartfonów, zdjęcia wykonane na ekwiwalencie ogniskowej 28mm) . . . . .	69
4.14	Wartości parametru dystorsji $k$ na podstawie zdjęć tego samego kadru (zbiór Dresden). Wykorzystano trzy egzemplarze aparatu Canon IXUS 70 oraz dwa egzemplarze Nikon D70 (zdjęcia wykonane na ekwiwalencie ogniskowej 27mm) . . . . .	70
4.15	Czas przetwarzania zdjęć . . . . .	71
4.16	Czas przetwarzania (w sekundach) 290 skanów . . . . .	71
4.17	Średni czas (w sekundach) potrzebny na wygenerowanie pojedynczego cyfrowego odcisku palca (zbiory IMAGINE i Dresden) . . . . .	74
4.18	Czas (w godzinach) potrzebny na nauczenie pojedynczej epoki . . . . .	75
4.19	Czas przetwarzania zdjęć (w minutach) . . . . .	79
4.20	Czas przetwarzania zdjęć (w minutach) . . . . .	81
5.1	Statystyki opisowe: algorytm MSE-DSI kontra literatura (zbiór IMAGINE) . . . . .	94
5.2	Statystyki opisowe: algorytm MSE-DSI kontra literatura (zbiór Dresden Image Database) . . . . .	95
5.3	Statystyki opisowe: algorytm MSE-DSI kontra Khanna (identyfikacja skanerów płaskich) . . . . .	95
5.4	Statystyki opisowe: algorytm CompaRe kontra algorytmy kompaktowej reprezentacji (zbiór IMAGINE) . . . . .	96

5.5	Statystyki opisowe: algorytm CompaRe kontra algorytmy kompaktowej reprezentacji (zbiór Dresden) . . . . .	96
5.6	Statystyki opisowe: algorytm CompaRe kontra pozostałe algorytmy (zbiór Dresden) . . . . .	96
5.7	Statystyki opisowe: algorytm CompaRe kontra pozostałe algorytmy (zbiór Dresden) . . . . .	96
5.8	Statystyki opisowe: algorytm Vignetting-CT kontra pozostałe algorytmy (zbiór IMAGINE) . . . . .	97
5.9	Statystyki opisowe: algorytm Vignetting-CT kontra pozostałe algorytmy (zbiór Dresden) . . . . .	98
5.10	Statystyki opisowe: algorytm Vignetting-CT kontra pozostałe algorytmy (zbiór smartfonów) . . . . .	98
5.11	Testy normalności rozkładu: zbiór IMAGINE kontra Dresden . . . . .	100
5.12	Sumy rang, porównanie (P1) . . . . .	102
5.13	Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P1) . . . . .	102
5.14	Sumy rang, porównanie (P2) . . . . .	103
5.15	Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P2) . . . . .	104
5.16	Sumy rang, porównanie (P3) . . . . .	105
5.17	Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P3) . . . . .	106
5.18	Sumy rang, porównanie (P4) . . . . .	107
5.19	Analiza POST-HOC: zbiór IMAGINE kontra Dresden, (P4) . . . . .	107
5.20	Wyniki testów normalności rozkładu: algorytm MSE-DSI kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	109
5.21	Sumy rang, algorytm MSE-DSI kontra literatura . . . . .	110
5.22	Analiza POST-HOC: algorytm MSE-DSI kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	111
5.23	Wyniki testów normalności rozkładu: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	114
5.24	Analiza rang: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	116
5.25	Analiza POST-HOC: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	116
5.26	Wyniki testów normalności rozkładu: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	118

5.27	Analiza rang POST-HOC: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	119
5.28	Analiza POST-HOC: algorytm CompaRe kontra literatura (dla zbiorów IMAGINE oraz Dresden) . . . . .	120
5.29	Wyniki testów normalności rozkładu: algorytm Vignetting-CT kontra Lukás (zbiory IMAGINE, Dresden oraz zb. smartfonów) . . . . .	121
5.30	Test U Manna-Whitneya: algorytm Vignetting-CT kontra Lukás (zbiory IMAGINE oraz Dresden) . . . . .	123
6.1	Wartości współczynnika korelacji $\rho$ dla kolejnych gęstości $k$ szumu pieprz i sól, zbiór IMAGINE . . . . .	130
6.2	Wartości współczynnika korelacji $\rho$ dla kolejnych gęstości $k$ szumu pieprz i sól, zbiór Dresden . . . . .	131
6.3	Macierz konfuzji (zbiór Dresden), identyfikacja modelu, szum „pieprz i sól” (proponowana CNN, $k = 0,2$ ), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	132
6.4	Wartości współczynnika $\rho$ dla rozmycia Gaussa dla różnych wartości parametru $\sigma$ , zbiór IMAGINE . . . . .	133
6.5	Wartości współczynnika $\rho$ dla rozmycia Gaussa dla różnych wartości parametru $\sigma$ , Dresden Image Database . . . . .	134
6.6	Macierz konfuzji (zbiór Dresden), identyfikacja modelu, rozmycie Gaussa (proponowana CNN, $\sigma = 0,1$ ), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	135
6.7	Wartości współczynnika $\rho$ dla szumu Poissona, wybrane urządzenia ze zbioru IMAGINE . . . . .	135
6.8	Wartości współczynnika $\rho$ dla szumu Poissona, wybrane urządzenia z bazy Dresden Image Database . . . . .	136
6.9	Macierz konfuzji (zbiór Dresden), identyfikacja modelu, szum Poissona (proponowana CNN), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	137
6.10	Macierz konfuzji (zbiór Dresden), identyfikacja modelu, usuwanie LSB (proponowana CNN, $p = 0,5$ ), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	138
A.1	Przykładowe wartości współczynnika $\rho$ , zbiór IMAGINE . . . . .	149
A.2	Przykładowe wartości współczynnika $\rho$ , zbiór Dresden Image Database . . . . .	150



B.1	Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Lukás), ACC = 95,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	152
B.2	Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (sieć konwolucyjna Tuama), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	153
B.3	Macierz konfuzji (Dresden Image Database), identyfikacja modelu (algorytm MSE-DSI), ACC = 75,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	153
B.4	Macierz konfuzji (Dresden Image Database), identyfikacja modelu (algorytm Lukás), ACC = 94,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	154
B.5	Macierz konfuzji (Dresden Image Database), identyfikacja modelu (sieć konwolucyjna Bondi), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	154
B.6	Macierz konfuzji (Dresden Image Database), identyfikacja modelu (sieć konwolucyjna Tuama), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	155
B.7	[Eksperyment I] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Li), ACC = 93,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	156
B.8	[Eksperyment II] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (metoda Bondi), ACC = 93,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	156
B.9	[Eksperyment II] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (metoda Tuama), ACC = 92,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	157
B.10	[Eksperyment II] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (metoda Mandelli), ACC = 92,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	157
B.11	[Eksperyment II] Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (metoda Kirchner & Johnson), ACC = 92,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	158
B.12	Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (algorytm CompaRe), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	158
B.13	[Eksperyment I] Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (algorytm Valsesia), ACC = 94,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	159
B.14	[Eksperyment I] Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (algorytm Li), ACC = 93,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	159
B.15	[Eksperyment II] Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (algorytm Lukás), ACC = 96,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	160

B.16 [Eksperyment II] Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (metoda Bondi), ACC = 95,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	160
B.17 [Eksperyment II] Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (metoda Tuama), ACC = 94,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	161
B.18 [Eksperyment II] Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (metoda Mandelli), ACC = 94,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	161
B.19 [Eksperyment II] Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (metoda Kirchner & Johnson), ACC = 92,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	162
B.20 Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Vignetting-CT), ACC = 66,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	163
B.21 Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu (algorytm Lukás), ACC = 95,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	163
B.22 Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (algorytm Vignetting-CT), ACC = 64,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	164
B.23 Macierz konfuzji (zbiór Dresden Image Database), identyfikacja modelu (algorytm Lukás), ACC = 94,0% (symbol * oznacza wartości mniejsze niż 5,0%) . . . . .	164
B.24 Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu, szum „pieprz i sól” (proponowana CNN, $k = 0,2$ ), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	165
B.25 Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu, rozmycie Gaussa (proponowana CNN, $\sigma = 0,1$ ), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	166
B.26 Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu, szum Poissona (proponowana CNN), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	166
B.27 Macierz konfuzji (zbiór IMAGINE), identyfikacja modelu, usuwanie LSB (proponowana CNN, $p = 0,5$ ), ACC = 99,0% (symbol * oznacza wartości mniejsze niż 1,0%) . . . . .	167

# Spis algorytmów

1	Algorytm Lukás [104]	30
2	Algorytm Valsesia [148]	32
3	Algorytm MSE-DSI	34
4	Algorytm CompaRe	38
5	Algorytm Vignetting-CT	41
6	Algorytm Distortion-CT	45
7	Algorytm przeciwdziałania identyfikacji wykorzystujący próbkowanie Lanczosa	141

# Bibliografia

- [1] MICHE database (download link), <http://biplab.unisa.it/MICHE/database/>, 2019. Online; ostatni dostęp 1 grudnia 2019.
- [2] Bernacki J., Scherer R., IMAGINE (2022), 2022.
- [3] Edmund Optics, Distortion (online), <https://www.edmundoptics.eu/knowledge-center/application-notes/imaging/distortion/>, 2022. Online; ostatni dostęp 24 września 2022.
- [4] Test Canon EOS RP – serwis optyczne.pl, [https://www.optyczne.pl/423.9-Test\\_aparatu-Canon\\_EOS\\_RP\\_Podsumowanie.html](https://www.optyczne.pl/423.9-Test_aparatu-Canon_EOS_RP_Podsumowanie.html), 2022. Online; ostatni dostęp 21 sierpnia 2022.
- [5] Hugin photo stitcher, <http://hugin.sourceforge.net/>.
- [6] Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.
- [7] RGB Adobe. Color image encoding, version 2005-05, 2005. *Adobe Systems Incorporated*, <https://www.adobe.com/digitalimag/pdfs/AdobeRGB1998.pdf>, 1998.
- [8] Akshay Agarwal, Rohit Keshari, Manya Wadhwa, Mansi Vijh, Chandani Parmar, Richa Singh, and Mayank Vatsa. Iris sensor identification in multi-camera environment. *Information Fusion*, 45:333–345, 2019.
- [9] Akshay Agarwal, Richa Singh, and Mayank Vatsa. Fingerprint sensor classification via mélange of handcrafted features. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pages 3001–3006, 2016.

- [10] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–5, 2020.
- [11] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internet—srgb. In *Color and imaging conference*, volume 1996, pages 238–245. Society for Imaging Science and Technology, 1996.
- [12] Deepak Nag Ayyala, Santu Ghosh, and Daniel F. Linder. Covariance matrix testing in high dimension using random projections. *Comput. Stat.*, 37(3):1111–1141, 2022.
- [13] Jamil Azzeh, Bilal Zahran, and Ziad Alqadi. Salt and pepper noise: Effects and removal. *JOIV: International Journal on Informatics Visualization*, 2(4):252–256, 2018.
- [14] Teun Baar, Wiger van Houten, and Zeno J. M. H. Geradts. Camera identification by grouping images from database, based on shared noise patterns. *CoRR*, abs/1207.2641, 2012.
- [15] Guru Swaroop Bennabhaktula, Derrick Timmerman, Enrique Alegre, and George Azzopardi. Source camera device identification from videos. *SN Comput. Sci.*, 3(4):316, 2022.
- [16] Jarosław Bernacki and Rafał Scherer. Digital forensics: a fast algorithm for a digital sensor identification. *Journal of Information and Telecommunication*, 0(0):1–21, 2022.
- [17] Jaroslaw Bernacki. Digital camera identification based on analysis of optical defects. *Multim. Tools Appl.*, 79(3-4):2945–2963, 2020.
- [18] Jaroslaw Bernacki. A survey on digital camera identification methods. *Digit. Investig.*, 34:300983, 2020.
- [19] Jaroslaw Bernacki. On robustness of camera identification algorithms. *Multim. Tools Appl.*, 80(1):921–942, 2021.
- [20] Jaroslaw Bernacki. Robustness of digital camera identification with convolutional neural networks. *Multim. Tools Appl.*, 80(19):29657–29673, 2021.
- [21] Jaroslaw Bernacki. Digital camera identification by fingerprint’s compact representation. *Multim. Tools Appl.*, 81(15):21641–21674, 2022.

- [22] Jarosław Bernacki, Kelton A. P. Costa, and Rafał Scherer. Individual source camera identification with convolutional neural networks. In Edward Szczerbicki, Krystian Wojtkiewicz, Sinh Van Nguyen, Marcin Pietranik, and Marek Krótkiewicz, editors, *Recent Challenges in Intelligent Information and Database Systems*, pages 45–55, Singapore, 2022. Springer Nature Singapore.
- [23] Jaroslaw Bernacki, Marek Klonowski, and Piotr Syga. Some remarks about tracing digital cameras - faster method and usable countermeasure. In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT, Madrid, Spain, July 24-26, 2017*, pages 343–350, 2017.
- [24] Jaroslaw Bernacki and Rafal Scherer. Fast imaging sensor identification. In Ngoc Thanh Nguyen, Lazaros Iliadis, Ilias Maglogiannis, and Bogdan Trawinski, editors, *Computational Collective Intelligence - 13th International Conference, ICCCI 2021, Rhodes, Greece, September 29 - October 1, 2021, Proceedings*, volume 12876 of *Lecture Notes in Computer Science*, pages 572–584. Springer, 2021.
- [25] Gajanan K. Birajdar and Vijay H. Mankar. Digital image forgery detection using passive techniques: A survey. *Digital Investigation*, 10(3):226–245, 2013.
- [26] Luca Bondi, Luca Baroffio, David Guera, Paolo Bestagini, Edward J. Delp, and Stefano Tubaro. First steps toward camera model identification with convolutional neural networks. *IEEE Signal Process. Lett.*, 24(3):259–263, 2017.
- [27] Morton B. Brown and Alan B. Forsythe. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367, 1974.
- [28] Faisal Bukhari and Matthew N. Dailey. Automatic radial distortion estimation from a single image. *J. Math. Imaging Vis.*, 45(1):31–45, 2013.
- [29] Wilhelm Burger and Mark J. Burge. *Principles of Digital Image Processing: Core Algorithms*. Springer Publishing Company, Incorporated, 1 edition, 2009.
- [30] Roberto Caldelli, Irene Amerini, and Andrea Novi. An analysis on attacker actions in fingerprint-copy attack in source camera identification. In *2011 IEEE International Workshop on Information Forensics and Security, WIFS 2011, Iguacu Falls, Brazil, November 29 - December 2, 2011*, pages 1–6, 2011.

- [31] Glenn H. Chapman, Rahul Thomas, Rohit Thomas, Zahava Koren, and Israel Koren. Enhanced correction methods for high density hot pixel defects in digital imagers. In Ralf Widenhorn and Antoine Dupret, editors, *Image Sensors and Imaging Systems 2015*, volume 9403, page 94030T. International Society for Optics and Photonics, SPIE, 2015.
- [32] Glenn H. Chapman, Rohit Thomas, Zahava Koren, and Israel Koren. Empirical formula for rates of hot pixel defects based on pixel size, sensor area, and ISO. In Ralf Widenhorn and Antoine Dupret, editors, *Sensors, Cameras, and Systems for Industrial and Scientific Applications XIV*, volume 8659, page 86590C. International Society for Optics and Photonics, SPIE, 2013.
- [33] Chen Chen and Matthew C. Stamm. Camera model identification framework using an ensemble of demosaicing features. In *2015 IEEE International Workshop on Information Forensics and Security, WIFS 2015, Roma, Italy, November 16-19, 2015*, pages 1–6, 2015.
- [34] Jiansheng Chen, Xiangui Kang, Ye Liu, and Z. J. Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal Process. Lett.*, 22(11):1849–1853, 2015.
- [35] Mo Chen, Jessica J. Fridrich, Miroslav Goljan, and Jan Lukás. Source digital camcorder identification using sensor photo response non-uniformity. In Edward J. Delp III and Ping Wah Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, USA, January 28, 2007*, volume 6505 of *SPIE Proceedings*, page 65051G. SPIE, 2007.
- [36] Sijia Chen, Yingxue Zhang, Yiming Li, Zhenzhong Chen, and Zhou Wang. Spherical structural similarity index for objective omnidirectional video quality assessment. In *2018 IEEE International Conference on Multimedia and Expo, ICME 2018, San Diego, CA, USA, July 23-27, 2018*, pages 1–6. IEEE Computer Society, 2018.
- [37] Yunshu Chen, Yue Huang, and Xinghao Ding. Camera model identification with residual neural network. In *2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017*, pages 4337–4341, 2017.
- [38] Kai San Choi, Edmund Y. Lam, and Kenneth K. Y. Wong. Source camera identification using footprints from lens aberration. In *Digital Photography II, San Jose, CA, USA, January 15, 2006*, page 60690J, 2006.

- [39] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.
- [40] Curtis A Corum. Dark frame subtraction: Uspto, 2000. Type: Other Patent.
- [41] Alexandre Ashade Lassance Cunha, Melissa Carvalho Costa, and Marco Aurélio Cavalcanti Pacheco. Sentiment analysis of youtube video comments using deep neural networks. In Leszek Rutkowski, Rafal Scherer, Marcin Korytkowski, Witold Pedrycz, Ryszard Tadeusiewicz, and Jacek M. Zurada, editors, *Artificial Intelligence and Soft Computing - 18th International Conference, ICAISC 2019, Zakopane, Poland, June 16-20, 2019, Proceedings, Part I*, volume 11508 of *Lecture Notes in Computer Science*, pages 561–570. Springer, 2019.
- [42] Maria De Marsico, Michele Nappi, Daniel Riccio, and Harry Wechsler. Mobile iris challenge evaluation (miche)-i, biometric iris dataset and protocols. *Pattern Recognition Letters*, 57:17–23, 2015.
- [43] Jason P. de Villiers, F. Wilhelm Leuschner, and Ronelle Geldenhuys. Centi-pixel accurate real-time inverse distortion correction. In John T. Wen, Dalibor Hodko, Yukitoshi Otani, Jonathan Kofman, and Okyay Kaynak, editors, *Optomechatronic Technologies 2008*, volume 7266, pages 320 – 327. International Society for Optics and Photonics, SPIE, 2008.
- [44] Luca Debiasi and Andreas Uhl. Comparison of PRNU enhancement techniques to generate PRNU fingerprints for biometric source sensor attribution. In *4th International Conference on Biometrics and Forensics, IWBF 2016, Limassol, Cyprus, March 3-4, 2016*, pages 1–6, 2016.
- [45] Zhonghai Deng, Arjan Gijssenij, and Jingyuan Zhang. Source camera identification using auto-white balance approximation. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 57–64, 2011.
- [46] Renuka. G. Deshpande, Lata L Ragha, and Satyendra Kumar Sharma. Video Quality Assessment through PSNR Estimation for Different Compression Standards. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(3):918–924, September 2018.



- [47] A. E. Dirik, H. T. Sencar, and N. Memon. Source camera identification based on sensor dust characteristics. In *2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, pages 1–6, 2007.
- [48] Ahmet Emir Dirik, Husrev T. Sencar, and Nasir D. Memon. Digital single lens reflex camera identification from traces of sensor dust. *IEEE Trans. Information Forensics and Security*, 3(3):539–552, 2008.
- [49] Matlab documentation. *Add noise to image*, ostatni dostęp: 30 grudnia 2020, <https://uk.mathworks.com/help/images/ref/imnoise.html>.
- [50] Olive Jean Dunn. Multiple comparisons using rank sums. *Technometrics*, 6(3):241–252, 1964.
- [51] Erwin Erwin. Improving retinal image quality using the contrast stretching, histogram equalization, and clahe methods with median filters. *International Journal of Image, Graphics and Signal Processing*, 12:30–41, 04 2020.
- [52] Weili Fang, Botao Zhong, Neng Zhao, Peter E. D. Love, Hanbin Luo, Jiayue Xue, and Shuangjie Xu. A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network. *Adv. Eng. Informatics*, 39:170–177, 2019.
- [53] Jan Flusser, Sajad Farokhi, Cyril Höschl IV, Tomás Suk, Barbara Zitová, and Matteo Pedone. Recognition of images degraded by gaussian blur. *IEEE Trans. Image Processing*, 25(2):790–806, 2016.
- [54] David Freire-Obregón, Fabio Narducci, Silvio Barra, and Modesto Castrillón Santana. Deep learning for source camera identification on mobile devices. *CoRR*, abs/1710.01257, 2017.
- [55] David Freire-Obregón, Fabio Narducci, Silvio Barra, and Modesto Castrillón Santana. Deep learning for source camera identification on mobile devices. *Pattern Recognition Letters*, 126:86–91, 2019.
- [56] Jessica Fridrich. Sensor defects in digital image forensic. *Digital Image Forensics: There is More to a Picture than Meets the Eye*, pages 179–218, 11 2013.
- [57] Jessica J. Fridrich. Digital image forensics. *IEEE Signal Process. Mag.*, 26(2):26–37, 2009.

- [58] Jessica J. Fridrich and Miroslav Goljan. Determining approximate age of digital images using sensor defects. In *Media Forensics and Security III, San Francisco Airport, CA, USA, January 24-26, 2011, Proceedings*, page 788006, 2011.
- [59] Chiara Galdi, Michele Nappi, and Jean-Luc Dugelay. Combining hardwaremetry and biometry for human authentication via smartphones. In *Image Analysis and Processing - ICIAP 2015 - 18th International Conference, Genoa, Italy, September 7-11, 2015, Proceedings, Part II*, pages 406–416, 2015.
- [60] Chiara Galdi, Michele Nappi, and Jean-Luc Dugelay. Multimodal authentication on smartphones: Combining iris and sensor recognition for a double check of user identity. *Pattern Recognition Letters*, 82:144–153, 2016.
- [61] Zeno J. Geradts, Jurrien Bijhold, Martijn Kieft, Kenji Kurosawa, Kenro Kuroki, and Naoki Saitoh. Methods for identification of images acquired with digital cameras. In Simon K. Bramble, Edward M. Carapezza, Lenny I. Rudin, Lenny I. Rudin, and Simon K. Bramble, editors, *Enabling Technologies for Law Enforcement and Security*, volume 4232, pages 505 – 512. International Society for Optics and Photonics, SPIE, 2001.
- [62] Thomas Gloe. Feature-based forensic camera model identification. *Trans. Data Hiding Multim. Secur.*, 8:42–62, 2012.
- [63] Thomas Gloe and Rainer Böhme. The ‘Dresden Image Database’ for benchmarking digital image forensics. In *Proceedings of the 25th Symposium On Applied Computing (ACM SAC 2010)*, volume 2, pages 1585–1591, 2010.
- [64] Miroslav Goljan. Digital camera identification from images - estimating false acceptance probability. In *Digital Watermarking, 7th International Workshop, IWDW 2008, Busan, Korea, November 10-12, 2008. Selected Papers*, pages 454–468, 2008.
- [65] Miroslav Goljan, Mo Chen, Pedro Comesaña, and Jessica J. Fridrich. Effect of compression on sensor-fingerprint based camera identification. In *Media Watermarking, Security, and Forensics 2016, San Francisco, California, USA, February 14-18, 2016*, pages 1–10, 2016.
- [66] Miroslav Goljan and Jessica J. Fridrich. Estimation of lens distortion correction from single images. In *Media Watermarking, Security, and Forensics 2014, San Francisco, CA, USA, February 2, 2014, Proceedings*, page 90280N, 2014.

- [67] Miroslav Goljan, Jessica J. Fridrich, and Mo Chen. Sensor noise camera identification: countering counter-forensics. In Nasir D. Memon, Jana Dittmann, Adnan M. Alattar, and Edward J. Delp, editors, *Media Forensics and Security II, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18-20, 2010, Proceedings*, volume 7541 of *SPIE Proceedings*, page 75410S. SPIE, 2010.
- [68] Miroslav Goljan, Jessica J. Fridrich, and Mo Chen. Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Trans. Information Forensics and Security*, 6(1):227–236, 2011.
- [69] Miroslav Goljan, Jessica J. Fridrich, and Tomas Filler. Large scale test of sensor fingerprint camera identification. In Edward J. Delp, Jana Dittmann, Nasir D. Memon, and Ping Wah Wong, editors, *Media Forensics and Security I, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 19-21, 2009, Proceedings*, volume 7254 of *SPIE Proceedings*, page 72540I. SPIE, 2009.
- [70] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [71] William Sealy Gosset. The probable error of a mean. *Biometrika*, 6(1):1–25, March 1908. Originally published under the pseudonym “Student”.
- [72] Rafal Grycuk and Rafal Scherer. Solar image hashing by intermediate descriptor and autoencoder. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8. IEEE, 2021.
- [73] Bhupendra Gupta and Mayank Tiwari. Improving source camera identification performance using dct based image frequency components dependent sensor pattern noise extraction method. *Digital Investigation*, 24:121 – 127, 2018.
- [74] Benjamin Hadwiger and Christian Riess. The forchheim image database for camera identification in the wild. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges - Virtual Event, January 10-15, 2021, Proceedings, Part VI*, volume 12666 of *Lecture Notes in Computer Science*, pages 500–515. Springer, 2020.

- [75] Gerald C. Holst. *CCD arrays, cameras, and displays / Gerald C. Holst*. JCD Pub. ; SPIE Optical Engineering Press Winter Park, FL : Bellingham, Wash, 1996.
- [76] Hossein Hosseini, Baicen Xiao, Mayoore Jaiswal, and Radha Poovendran. On the limitation of convolutional neural networks in recognizing negative images. In Xuewen Chen, Bo Luo, Feng Luo, Vasile Palade, and M. Arif Wani, editors, *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*, pages 352–358. IEEE, 2017.
- [77] Morteza Darvish Morshedi Hosseini and Miroslav Goljan. Camera identification from HDR images. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec 2019, Paris, France, July 3-5, 2019*, pages 69–76, 2019.
- [78] James R Janesick, Tom Elliott, Stewart Collins, Morley M Blouke, and Jack Freeman. Scientific charge-coupled devices. *Optical Engineering*, 26(8):692–714, 1987.
- [79] Xiang Jiang, Shikui Wei, Ruizhen Zhao, Yao Zhao, and Xindong Wu. Camera fingerprint: A new perspective for identifying user’s identity. *CoRR*, abs/1610.07728, 2016.
- [80] Thibault Julliard, Vincent Nozick, and Hugues Talbot. Image noise and digital image forensics. In *Digital-Forensics and Watermarking - 14th International Workshop, IWDW 2015, Tokyo, Japan, October 7-10, 2015, Revised Selected Papers*, pages 3–17, 2015.
- [81] Nathan D. Kalka, Nick Bartlow, Bojan Cukic, and Arun Ross. A preliminary study on identifying sensors from iris images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2015, Boston, MA, USA, June 7-12, 2015*, pages 50–56, 2015.
- [82] Xiangui Kang, Yinxiang Li, Zhenhua Qu, and Jiwu Huang. Enhancing source camera identification performance with a camera reference phase sensor pattern noise. *IEEE Trans. Information Forensics and Security*, 7(2):393–402, 2012.
- [83] Nitin Khanna, George T.-C. Chiu, Jan P. Allebach, and Edward J. Delp. Forensic techniques for classifying scanner, computer generated and digital camera images. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, pages 1653–1656, 2008.

- [84] Nitin Khanna, Aravind K. Mikkilineni, George T.-C. Chiu, Jan P. Allebach, and Edward J. Delp. Scanner identification using sensor pattern noise. In *Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, USA, January 28, 2007*, page 65051K, 2007.
- [85] Matthias Kirchner and Cameron Johnson. SPN-CNN: boosting sensor-based source camera attribution with deep learning. *CoRR*, abs/2002.02927, 2020.
- [86] K. Kitami, M. Murai, N. Sugaya, O. Kikuchi, and S. Ohno. New technique for acquiring dead pixel free and fine inspection image of advanced lsi package with rough surface using scanning acoustic tomograph. In *Proceedings of the 21th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pages 340–343, June 2014.
- [87] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *CoRR*, abs/1706.02515, 2017.
- [88] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley, Reading, Mass., third edition, 1997.
- [89] A. Kordecki, H. Palus, and A. Bal. Fast vignetting reduction method for digital still camera. In *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 1145–1150, Aug 2015.
- [90] WH Kruskal and WA Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, pages 583–621, 1952.
- [91] William H. Kruskal. A Nonparametric test for the Several Sample Problem. *The Annals of Mathematical Statistics*, 23(4):525 – 540, 1952.
- [92] Kolmogorov A. L. Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari*, 4:83–91, 1933.
- [93] T. Van Lanh, K. S. Chong, S. Emmanuel, and M. S. Kankanhalli. A survey on digital camera image forensic methods. In *2007 IEEE International Conference on Multimedia and Expo*, pages 16–19, July 2007.
- [94] S. Lawrence, C.L. Giles, Ah Chung Tsoi, and A.D. Back. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

- [95] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [96] Jenny Leung, Glenn H. Chapman, Yong H. Choi, Rohit Thomas, Israel Koren, and Zahava Koren. Analyzing the impact of ISO on digital imager defects with an automatic defect trace algorithm. In Erik Bodegom and Valérie Nguyen, editors, *Sensors, Cameras, and Systems for Industrial/Scientific Applications XI*, volume 7536, page 75360F. International Society for Optics and Photonics, SPIE, 2010.
- [97] Haoliang Li, Shiqi Wang, and Alex C. Kot. Image recapture detection with convolutional and recurrent neural networks. In *Media Watermarking, Security, and Forensics 2017, Burlingame, CA, USA, 29 January 2017 - 2 February 2017*, pages 87–91, 2017.
- [98] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. Medical image classification with convolutional neural network. In *2014 13th International Conference on Control Automation Robotics and Vision (ICARCV)*, pages 844–848, 2014.
- [99] Ruizhe Li, Chang-Tsun Li, and Yu Guan. Inference of a compact representation of sensor fingerprint for source camera identification. *Pattern Recognition*, 74:556–567, 2018.
- [100] Xiaoyu Li, Bo Zhang, Pedro V. Sander, and Jing Liao. Blind geometric distortion correction on images through deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4855–4864. Computer Vision Foundation / IEEE, 2019.
- [101] Xinghua Li, Huanfeng Shen, Liangpei Zhang, Hongyan Zhang, and Qiangqiang Yuan. Dead pixel completion of aqua MODIS band 6 using a robust m-estimator multiregression. *IEEE Geosci. Remote Sensing Lett.*, 11(4):768–772, 2014.
- [102] Hubert W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, Jun. 1967.
- [103] Hubert W. Lilliefors. On the kolmogorov-smirnov test for the exponential distribution with mean unknown. *Journal of the American Statistical Association*, 64(325):387–389, Mar. 1969.
- [104] Jan Lukás, Jessica J. Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. Information Forensics and Security*, 1(2):205–214, 2006.

- [105] Jan Lukás, Jessica J. Fridrich, and Miroslav Goljan. Camera fingerprint matlab implementation, <http://dde.binghamton.edu/download/> (ostatni dostęp: 27 października 2021). 2016.
- [106] Hanbin Luo, Chaohua Xiong, Weili Fang, Peter E.D. Love, Bowen Zhang, and Xi Ouyang. Convolutional neural networks: Computer vision-based workforce activity assessment in construction. *Automation in Construction*, 94:282–289, 2018.
- [107] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognit.*, 91:216–231, 2019.
- [108] Sara Mandelli, Davide Cozzolino, Paolo Bestagini, Luisa Verdoliva, and Stefano Tubaro. Cnn-based fast source device identification. *IEEE Signal Process. Lett.*, 27:1285–1289, 2020.
- [109] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947.
- [110] Francesco Marra, Diego Gagnaniello, and Luisa Verdoliva. On the vulnerability of deep learning to adversarial attacks for camera model identification. *Sig. Proc.: Image Comm.*, 65:240–248, 2018.
- [111] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Evaluation of residual-based local features for camera model identification. In *New Trends in Image Analysis and Processing - ICIAP 2015 Workshops - ICIAP 2015 International Workshops: BioFor, CTMR, RHEUMA, ISCA, MADiMa, SBMI, and QoEM, Genoa, Italy, September 7-8, 2015, Proceedings*, pages 11–18, 2015.
- [112] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Correlation clustering for prnu-based blind image source identification. In *IEEE International Workshop on Information Forensics and Security, WIFS 2016, Abu Dhabi, United Arab Emirates, December 4-7, 2016*, pages 1–6, 2016.
- [113] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. A deep learning approach for iris sensor model identification. *Pattern Recognit. Lett.*, 113:46–53, 2018.

- [114] Christiaan Meij and Zeno Geradts. Source camera identification using photo response non-uniformity on whatsapp. *Digital Investigation*, 24:142 – 154, 2018.
- [115] Arjan Mieremet. Camera-identification and common-source identification: The correlation values of mismatches. *Forensic Science International*, 301:46 – 54, 2019.
- [116] A.M. Mood, F.A. Graybill, and D.C. Boes. *Introduction to the Theory of Statistics*. International Student edition. McGraw-Hill, 1973.
- [117] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582, 2016.
- [118] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress, 2010.
- [119] Jakub Nowak, Taras Holotyak, Marcin Korytkowski, Rafal Scherer, and Slava Voloshynovskiy. Fingerprinting of URL logs: Continuous user authentication from behavioural patterns. In Valeria V. Krzhizhanovskaya, Gábor Závodszy, Michael Harold Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira, editors, *Computational Science - ICCS 2020 - 20th International Conference, Amsterdam, The Netherlands, June 3-5, 2020, Proceedings, Part IV*, volume 12140 of *Lecture Notes in Computer Science*, pages 184–195. Springer, 2020.
- [120] Jakub Nowak, Marcin Korytkowski, Marcin Wozniak, and Rafal Scherer. Url-based phishing attack detection by convolutional neural networks. *Aust. J. Intell. Inf. Process. Syst.*, 15(2):60–67, 2019.
- [121] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 372–387, 2016.
- [122] Dawid Polap, Marcin Wozniak, Marcin Korytkowski, and Rafal Scherer. Encoder-decoder based CNN structure for microscopic image identification. In Haiqin Yang, Kitsuchart Pasupa, Andrew Chi-Sing Leung, James T. Kwok, Jonathan H. Chan, and Irwin King,



- editors, *Neural Information Processing - 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23-27, 2020, Proceedings, Part I*, volume 12532 of *Lecture Notes in Computer Science*, pages 301–312. Springer, 2020.
- [123] B. Prescott and G.F. McLean. Line-based correction of radial lens distortion. *Graphical Models and Image Processing*, 59(1):39–47, 1997.
- [124] Nanmaran R and S.Hari Priya. Design and development of decorrelation stretch technique for enhancing the quality of satellite images with improved MSE and UIQI in comparison with wiener filter. *ECS Transactions*, 107(1):13279–13288, apr 2022.
- [125] Chaima Ben Rabah, Gouenou Coatrieux, and Riadh Abdelfattah. Automatic source scanner identification using 1d convolutional neural network. *Multim. Tools Appl.*, 81(16):22789–22806, 2022.
- [126] Abdul Muntakim Rafi, Uday Kamal, Rakibul Hoque, Abid Abrar, Sowmitra Das, Robert Laganière, and Md. Kamrul Hasan. Application of densenet in camera model identification and post-processing detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 19–28, 2019.
- [127] Patrick Royston. Approximating the shapiro-wilk w-test for non-normality. *Statistics and Computing*, 2(3):117–119, 1992.
- [128] Frank Sandon. Experimental designs. by w. g. cochran and g. m. cox. 2nd ed. pp. xiv, 611. 82s. 1957. (john wiley and sons, new york; chapman and hall, london). *The Mathematical Gazette*, 42(342):334–334, 1958.
- [129] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 143–152. IEEE Computer Society, 2006.
- [130] Riccardo Satta. Sensor pattern noise matching based on reliability map for source camera identification. In *VISAPP 2015 - Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Volume 1, Berlin, Germany, 11-14 March, 2015*, pages 222–226, 2015.
- [131] F. E. Satterthwaite. An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6):110–114, 1946.

- [132] Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.
- [133] Omar Al Shaya, Pengpeng Yang, Rongrong Ni, Yao Zhao, and Alessandro Piva. A new dataset for source identification of high dynamic range images. *Sensors*, 18(11):3801, 2018.
- [134] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya, and Alessandro Piva. VISION: a video and image dataset for source identification. *EURASIP J. Information Security*, 2017:15, 2017.
- [135] Martin Steinebach, Huajian Liu, Peishuai Fan, and Stefan Katzenbeisser. Cell phone camera ballistics: attacks and countermeasures. In Reiner Creutzburg and David Akopian, editors, *Multimedia on Mobile Devices 2010*, volume 7542, page 75420B. International Society for Optics and Photonics, SPIE, 2010.
- [136] Yagiz Sutcu, Sevinc Bayram, Husrev T. Sencar, and Nasir D. Memon. Improvements on sensor noise based source camera identification. In *ICME*, pages 24–27. IEEE, 2007.
- [137] Piotr Syga, Cezary Sieluzycski, Patrycja Krzyzanowska-Berkowska, and D. Robert Iskander. Fully automated detection of lamina cribrosa in optical coherence tomography: Framework and illustrative examples. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju Island, South Korea, July 11-15, 2017*, pages 608–611. IEEE, 2017.
- [138] Thanh Hai Thai, Florent Reiraint, and Rémi Cogranne. Camera model identification based on the generalized noise model in natural images. *Digital Signal Processing*, 48:285–297, 2016.
- [139] H. Tian, Y. Xiao, G. Cao, J. Ding, and B. Ou. Robust watermarking of mobile video resistant against barrel distortion. *China Communications*, 13(9):131–138, Sep. 2016.
- [140] Mayank Tiwari and Bhupendra Gupta. Image features dependant correlation-weighting function for efficient prnu based source camera identification. *Forensic Science International*, 285:111 – 120, 2018.
- [141] Kenny Kal Vin Toh and Nor Ashidi Mat Isa. Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. *IEEE Signal Processing Letters*, 17(3):281–284, 2010.

- [142] Yoichi Tomioka and Hitoshi Kitazawa. Digital camera identification based on the clustered pattern noise of image sensors. In *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, ICME 2011, 11-15 July, 2011, Barcelona, Catalonia, Spain*, pages 1–4, 2011.
- [143] Min-Jen Tsai, Chen-Sheng Wang, Jung Liu, and Jin-Sheng Yin. Using decision fusion of feature selection in digital forensics for camera source model identification. *Computer Standards & Interfaces*, 34(3):292–304, 2012.
- [144] Amel Tuama, Frédéric Comby, and Marc Chaumont. Camera model identification based machine learning approach with high order statistics features. In *24th European Signal Processing Conference, EUSIPCO 2016, Budapest, Hungary, August 29 - September 2, 2016*, pages 1183–1187, 2016.
- [145] Amel Tuama, Frédéric Comby, and Marc Chaumont. Camera model identification with the use of deep convolutional neural networks. In *IEEE International Workshop on Information Forensics and Security, WIFS 2016, Abu Dhabi, United Arab Emirates, December 4-7, 2016*, pages 1–6. IEEE, 2016.
- [146] Ken Turkowski. Graphics gems. chapter Filters for Common Resampling Tasks, pages 147–165. Academic Press Professional, Inc., San Diego, CA, USA, 1990.
- [147] Andreas Uhl and Yvonne Höller. Iris-sensor authentication using camera PRNU fingerprints. In *5th IAPR International Conference on Biometrics, ICB 2012, New Delhi, India, March 29 - April 1, 2012*, pages 230–237, 2012.
- [148] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli. Compressed fingerprint matching and camera identification via random projections. *IEEE Transactions on Information Forensics and Security*, 10(7):1472–1485, July 2015.
- [149] Fei Wang and Ping Li. Efficient nonnegative matrix factorization with random projections. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 281–292. SIAM, 2010.
- [150] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, 2019.

- [151] Yan Wang, Qindong Sun, Dongzhu Rong, Shancang Li, and Li Da Xu. Image source identification using convolutional neural networks in iot environment. *Wirel. Commun. Mob. Comput.*, 2021:5804665:1–5804665:12, 2021.
- [152] Ruyue Xin, Jiang Zhang, and Yitong Shao. Complex network classification with convolutional neural network. *CoRR*, abs/1802.00539, 2018.
- [153] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015.
- [154] Pengpeng Yang, Rongrong Ni, and Yao Zhao. Recapture image forensics based on laplacian convolutional neural networks. In *Digital Forensics and Watermarking - 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers*, pages 119–128, 2016.
- [155] Pengpeng Yang, Rongrong Ni, Yao Zhao, and Wei Zhao. Source camera identification based on content-adaptive fusion residual networks. *Pattern Recognition Letters*, 119:195–204, 2019.
- [156] Hongwei Yao, Tong Qiao, Ming Xu, and Ning Zheng. Robust multi-classifier for camera model identification based on convolution neural network. *IEEE Access*, 6:24973–24982, 2018.
- [157] Changhui You, Hong Zheng, Zhongyuan Guo, Tianyu Wang, and Xiongbin Wu. Multiscale content-independent feature fusion network for source camera identification. *Applied Sciences*, 11(15), 2021.
- [158] Hui Zeng, Jiansheng Chen, Xiangui Kang, and Wenjun Zeng. Removing camera fingerprint to disguise photograph source. In *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, pages 1687–1691, 2015.
- [159] Botao Zhong, Xuejiao Xing, Peter E. D. Love, Xu Wang, and Hanbin Luo. Convolutional neural network: Deep learning-based classification of building quality problems. *Adv. Eng. Informatics*, 40:46–57, 2019.