JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 32 (2024) 179–185
Received: 20 May 2024
Accepted: 10 June 2024

Comparative analysis of CNN models for handwritten digit recognition

Analiza porównawcza modeli CNN do rozpoznawania cyfr pisma odręcznego

Krystyna Lidia Banaszewska*, Małgorzata Plechawska-Wójcik

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract
The paper discusses the subject of convolutional neural networks used for handwritten digit classification. The purpose of the research is to evaluate the accuracy, performance, training, and classification time of three OCR networks (VGG-16, VGG-19 and AlexNet) and compare them with each other while selecting the most optimal one. The popular MNIST dataset of 70,000 images was used for the study. For each model, a preliminary study was conducted to determine the optimal parameters in the form of the number of input data and number of training epochs. The result of the work indicates that, despite the longer training and classification time, the AlexNet model achieved the highest precision, recall, and F1-score, indicating its ability to effectively classify images.

Keywords: convolutional neural networks; handwriting classification

Streszczenie
W artykule podejmowana jest tematyka konwolucyjnych sieci neuronowych wykorzystywanych do klasyfikacji cyfr pisma odręcznego. Celem badań jest ocena dokładności, wydajności oraz czasu treningu i klasyfikacji trzech sieci OCR (VGG-16, VGG-19 i AlexNet) oraz porównanie ich między sobą wybierając przy tym ten najbardziej optymalny. Do badań wykorzystano popularny zestaw danych *MNIST* składający się z 70 000 obrazów. Dla każdego modelu przeprowadzono wstępne badanie w celu określenia optymalnej liczby danych wejściowych oraz liczbę epok treningowych. Wyniki badań wskazują, że pomimo dłuższego czasu treningu i klasyfikacji, model AlexNet osiągnął najwyższe wyniki w zakresie precyzji, czułości oraz F1-score, co wskazuje na jego zdolność do skutecznej klasyfikacji obrazów.

Słowa kluczowe: konwolucyjne sieci neuronowe; klasyfikacja pisma odręcznego

*Corresponding author
Email address: s100787@pollub.edu.pl (K. L. Banaszewska)

1. Wstęp

Rozpoznanie wzorców i przetwarzanie obrazów to kluczowe obszary badań sztucznej inteligencji, nabierające znaczenia wraz z rozwojem technologii. Rozpoznawanie pisma ręcznego znajduje zastosowanie w różnych sektorach, od digitalizacji zasobów dziedzictwa po rekonstrukcję i tworzenie muzyki. Obecnie na świecie realizowane są liczne projekty badawcze dotyczące rozpoznawania tekstów w starożytnych językach. Na świecie używa się ponad 7 tysięcy języków, a szacuje się, że w historii ludzkości było ich około 31 tysięcy, z czego połowa zniknęła w ostatnich 500 latach. Algorytmy uczenia maszynowego z powodzeniem przetłumaczyły starożytne tabliczki z Knossos. Od 2018 roku na Uniwersytecie w Toronto trwają prace nad automatycznym tłumaczeniem 69 tysięcy tekstów mezopotamskich z XXI wieku p.n.e., z których odczytano tylko 10%. W Japonii trwają próby opracowania narzędzia do rozpoznawania zapomnianych stylów pisma kuzushiji, co jest trudne z powodu różnorodności stylów i problemów z separacją znaków [1].

Mimo postępów, rozpoznawanie pisma odręcznego wciąż stawia wyzwania, ze względu na jego zmienność i zależność od nawyków pisania. Modele konwolucyjnych sieci neuronowych, takie jak VGG-16, VGG-19 i AlexNet, zrewolucjonizowały analizę obrazów, zapewniając fundamenty do zrozumienia wpływu głębokości sieci na skuteczność przetwarzania obrazów. Model AlexNet zdobył popularność, wygrywając konkurs *ImageNet* w 2012 roku, podczas gdy VGG-16 i VGG-19 wykazały, że zwiększanie głębokości sieci poprawia dokładność rozpoznawania [2, 3, 4].

W niniejszym artykule przedstawiono analizę trzech modeli CNN, by zrozumieć ich zdolność do nauki i generalizacji oraz wpływ głębokości na wydajność. Skupienie się na modelach AlexNet i VGG pomaga zgłębić fundamenty i ewolucję technik rozpoznawania obrazów, co pozwala na badanie potencjalnych usprawnień w praktycznych zastosowaniach [3].

Celem badań jest analiza oraz porównanie głębokości i architektury modeli sieci neuronowych VGG-16, VGG-19 oraz AlexNet pod kątem skuteczności w rozpoznawaniu cyfr pisma ręcznego. Badanie skupia się na dokładności, czasie przetwarzania i wydajności tych modeli w różnych środowiskach operacyjnych. Eksperymentalna analiza modeli obejmuje ocenę ich zdolności adaptacyjnych w różnych scenariuszach oraz identyfikację potencjalnej optymalizacji architektury.

Analiza dotyczy trzech kluczowych aspektów:

- dokładności rozpoznawania wzorców w różnych warunkach obliczeniowych,
- czasu i zasobów wymaganych do przetwarzania i treningu sieci,

- adaptacji i skalowania modeli w aplikacjach praktycznych,

W artykule znajdują się odpowiedzi na następujące pytania:

1. Jak głębokość sieci neuronowych wpływa na dokładność i czas przetwarzania w zadaniu rozpoznawania cyfr pisma ręcznego?
2. Jakie są wymagania dotyczące pamięci obliczeniowej dla efektywnego treningu głębokich sieci neuronowych z perspektywy rozpoznawania pisma ręcznego?
3. Jakie są potencjalne strategie optymalizacji architektury sieci neuronowych w celu poprawy ich wydajności w praktycznych aplikacjach rozpoznawania pisma ręcznego?

2. Analiza stanu wiedzy

2.1. Konwolucyjne Sieci Neuronowe

W widzeniu komputerowym (ang. *Computer Vision)*, obrazy są reprezentowane, jako macierze liczbowe z wartościami zawierającymi się w przedziale od 0 do 255. Dwa podstawowe zadania sztucznej inteligencji w tym kontekście - klasyfikacji i regresji - wymagają precyzyjnej identyfikacji specyficznych cech. Modele obliczeniowe analizują wzorce w danych wejściowych, aby przyporządkować obiekty do odpowiednich klas [5].

Sieci konwolucyjne wykorzystują wielowarstwową architekturę, gdzie każda warstwa specjalizuje się w wykrywaniu innych atrybutów obrazu. Stopniowo rozpoznają one coraz bardziej złożone wzorce, umożliwiając efektywną klasyfikację obiektów poprzez hierarchiczne zwiększanie stopnia abstrakcji [2, 5].

Operacja splotu, zwana również konwolucją polega na zastosowaniu filtrów do obrazów wejściowych, gdzie każdy filtr oblicza ważoną sumę pikseli i wartości filtra, tworząc mapy cech z esencjonalną informacją przestrzenną. Każda warstwa konwolucyjna zawiera wiele filtrów, z których każdy tworzy osobną mapę cech, wykrywając wzorce w różnych regionach obrazu. Te mapy są następnie przetwarzane przez kolejne warstwy w celu ekstrakcji cech wyższego poziomu, umożliwiając skuteczną klasyfikację lub analizę [2, 5].

Parametry wypełnienia (ang. *padding*) i kroku (ang. *stride*) są kluczowe w operacjach splotu. Wypełnienie dodaje zera wokół obrazu wejściowego, co umożliwia kontrolowanie rozmiaru wyjściowej mapy cech. Krok określa, o ile pikseli przesuwa się filtr w każdym kierunku. Większy krok zmniejsza wymiar map cech, co redukuje liczbę parametrów i poprawia efektywność obliczeniową [5].

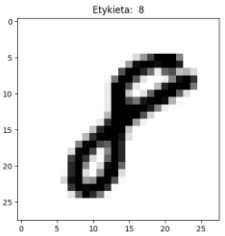
Dwie kluczowe funkcje współwystępują w każdej warstwie wraz z operacją konwolucji: funkcja aktywacji oraz operacja *max poolingu*. Aktywacja jest kluczowa dla modelowania złożonych wzorców. Funkcja ta wprowadza do równania nieliniowość, co usprawnia proces uczenia, zwiększając zdolność modelu do generalizacji danych. Najczęściej spotykaną funkcją aktywacji, użytą również w przypadku badanych sieci jest funkcja ReLU [2, 3]. *Max pooling* to proces redukcji wymiarowości, który przekazuje najbardziej wyraziste cechy, zmniejszając ilość danych i zwiększając odporność na zmiany położenia [2, 5].

Porzucenie (ang. *dropout)* to technika regularyzacyjna ograniczająca przeuczenie przez losowe wyłączanie neuronów podczas treningu, poprawiając zdolność modelu do generalizacji na nowe dane. Dzięki temu model jest bardziej odporny na specyficzne cechy zbioru treningowego i efektywniej działa na danych testowych i w praktyce [2].

W pełni połączone warstwy (ang. *fully connected layers)* analizują cechy i integrują informacje, doprowadzając do ostatecznej klasyfikacji lub predykcji. Warstwa wyjściowa dokonuje końcowej decyzji na podstawie przetworzonych danych, korzystając z odpowiednich funkcji aktywacji, takich jak *softmax* [2, 5].

2.2. Zbiór danych MNIST

Kluczowym elementem w badaniach nad rozpoznawaniem cyfr pisma ręcznego jest wybór odpowiedniej bazy danych. Do treningu i oceny wybranych modeli wykorzystano bazę *MNIST* (*ang. Modified National Institute of Standards and Technology*) [6], powszechnie stosowaną w uczeniu maszynowym i widzeniu komputerowym. *MNIST* składa się z 70 000 obrazów cyfr pisma ręcznego, z czego 60 000 przeznaczono do treningu, a 10 000 do testu. Każdy obraz, znormalizowany do rozmiaru 28x28 pikseli, przedstawia pojedynczą cyfrę od 0 do 9 w skali szarości (Rysunek 1) [6, 7].



Rysunek 1: Przykładowy obraz z zestawu *MNIST*.

Wykorzystanie *MNIST* umożliwia bezpośrednie porównanie z innymi badaniami i zapewnia solidną podstawę do oceny zdolności modeli do nauki i generalizacji. Normalizacja danych, polegająca na skalowaniu wartości pikseli od 0 do 1, oraz dostosowanie rozmiaru obrazów do wymagań sieci przy użyciu odpowiednich technik interpolacji są kluczowe przed przystąpieniem do trenowania modelu [6, 7].

2.3. Metryki porównawcze

Odpowiednie metryki porównawcze umożliwiają rzetelne określenie, które modele osiągają najlepsze wyniki pod kątem precyzji, szybkości, stabilności, a także innych istotnych aspektów. Wybór właściwych wskaźników oceny zależy od konkretnej aplikacji i kontekstu badawczego. Powszechnie stosowane miary, takie jak dokładność, czułość, precyzja, miara F1, czy też obliczenia związane z czasem treningu i klasyfikacji, pomagają w zrozumieniu mocnych i słabych stron poszczególnych modeli [8].

Macierz błędów (ang. *Confusion Matrix*), nazywana również macierzą pomyłek, to narzędzie używane

w analizie statystycznej do oceny wydajności klasyfikacji. Jest ona macierzą przedstawiającą liczbę przypadków, które zostały sklasyfikowane w odpowiedzi na test lub model, jako:

- Prawdziwie pozytywne, TP *(*ang. *True Positive)* - liczba przypadków, które zostały sklasyfikowane poprawnie.
- Prawdziwie negatywne, TF *(ang. True False)* - liczba przypadków poprawnie sklasyfikowanych, jako negatywne.
- Fałszywe pozytywne, FP *(*ang. *False Positive)* - liczba przypadków błędnie sklasyfikowanych, jako pozytywne.
- Fałszywie negatywne, FN *(*ang. *False Negative)* - liczba przypadków, które zostały błędnie sklasyfikowane, jako negatywne, podczas, gdy w rzeczywistości były one pozytywne [8, 9].

Analiza macierzy błędów pozwala ocenić zdolność modelu do generalizacji oraz zidentyfikować specyficzne problemy z klasyfikacją. Na jej podstawie można obliczyć dodatkowe metryki, takie jak dokładność, precyzja, czułość i metryka F1.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{1}$$

gdzie *accuracy* to dokładność oznaczająca procent obrazów, które zostały sklasyfikowane poprawnie [8].

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

gdzie *precision* to precyzja, która ocenia, jak wiele z pozytywnych prognoz modelu jest rzeczywiście poprawnych [8].

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

gdzie *recall* to czułość, metryka określająca zdolność modelu do wykrycia wszystkich pozytywnych przypadków [8].

$$F1_{score} = \frac{2\ (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4}$$

gdzie $F1_{score}$ to miara łącząca zarówno precyzję, jak i czułość w jedną wartość. Ocenia równowagę między dokładnością i zdolnością modelu do wykrywania pozytywnych przypadków [9].

Następną możliwą metryką oceny wydajności klasyfikatorów, pokazująca związek między czułością a specyficznością dla poszczególnej klasy jest krzywa ROC. Im bardziej krzywa oddalona jest od linii losowej i zbliża się do wartości 1,0 dla *True Positive Rate*, tym lepiej działa model. Powierzchnia pod krzywą, znana, jako AUC (*Area Under Curve),* reprezentuje prawdopodobieństwo, że klasyfikator oceni losowo wybraną pozytywną instancję wyżej niż losowo wybraną negatywną [8, 9].

2.4. Wymagania sprzętowe

Trenowanie złożonych modeli sztucznej inteligencji wymaga znacznych zasobów sprzętowych z powodu wysokiej złożoności obliczeniowej problemu. Wymagana jest duża ilość pamięci RAM, aby zarządzać danymi, przechowywać wagi oraz aktywacje pośrednie podczas uczenia. Ponadto, intensywne operacje konwolucji i *pooling'u* czynią GPU kluczowym elementem, dzięki równoległemu przetwarzaniu danych, co znacząco skraca czas trenowania. Wykorzystanie zasobów z *Google Colab* w postaci karty graficznej *Tesla T4 GPU* z 16GB pamięci podręcznej zapewnia elastyczność, dostępność, oraz wydajność w trenowaniu złożonych modeli i zarządzaniu dużymi zbiorami danych. Porównanie wydajności GPU i CPU pozwala na wybór optymalnej konfiguracji obliczeniowej w zależności od wymagań projektu.

3. Opis wybranych modeli CNN

Wybór modeli VGG-16, VGG-19 oraz AlexNet, jako przedmiotu analizy porównawczej w kontekście rozpoznawania pisma odręcznego opiera się na złożonym zestawie czynników, które wykraczają poza ich bezpośrednią skuteczność w danych zadaniach. Te modele, uznawane za kamienie milowe w dziedzinie uczenia głębokiego, zapewniają fundament do eksploracji wpływu głębokości sieci i specyfiki architektury na zdolność do przetwarzania obrazów [2].

3.1. Budowa modelu VGG-16

VGG-16 to model konwolucyjnej sieci neuronowej opracowany przez zespół *Visual Graphics Group* na Uniwersytecie Oksfordzkim, przedstawiony w publikacji „*Very Deep Convolutional Networks for Large-Scale Image Recognition*" autorstwa Karen Simanyan i Andrew Zisserman [2].

Model ten składa się z 13 warstw splotu z filtrami o rozmiarze 3x3 ze skokiem równym 1 oraz warstw max pooling 2x2 ze skokiem równym 2. Na końcu sieci znajdują się 3 w pełni połączone warstwy. Pierwsze dwie mają po 4096 kanałów, a ostatnia – wyjściowa – ma tyle kanałów ile klas (w przypadku *MNIST*, jest to 10 klas) [2, 10].

3.2. Budowa modelu VGG-19

Architektura sieci VGG-19 jest ewolucją modelu VGG-16. Główną różnicą między siostrzanymi sieciami jest liczba warstw z wagami. Architektura sieci VGG-19 składa się z 19 warstw, z czego: 16 warstw to warstwy konwolucyjne i 3 w pełni połączone. W tym przypadku również warstwy konwolucyjne są układane sekwencyjnie, z warstwami *max-pooling* rozdzielającymi niektóre z nich. Większa głębokość sieci prowadzi do wzrostu liczby parametrów, co może zwiększać ryzyko przeuczenia oraz wymagać więcej zasobów obliczeniowych i pamięci podczas treningu i wnioskowania [10].

3.3. Budowa modelu AlexNet

AlexNet to konwolucyjna sieć neuronowa opracowana przez Alexa Krizhevsky'ego, Ilya Sutskevera i Geoffreya Hintona. Model zajął pierwsze miejsce w konkursie *ImageNet LSVRC-2012*, z przewagą 10,8% przy współczynniku błędów 15,5%, co podkreśliło potencjał głębokich

sieci neuronowych w przetwarzaniu złożonych zadań wizualnych. Architektura AlexNet zawiera 8 warstw z wagami: 5 konwolucyjnych i 3 w pełni połączone, z aktywacją ReLU na końcu każdej z nich (oprócz ostatniej). *Dropout* zastosowano w dwóch pierwszych w pełni połączonych warstwach. Ostatnia warstwa zwraca wyniki za pomocą funkcji *softmax*, klasyfikując je w 1000 kategorii [3].

4. Scenariusze badawcze

Niniejszy podpunkt opisuje badania wykonane na omawianych modelach, koncentrujące się na czterech kluczowych obszarach oceny: zbieżności modelu, dokładności klasyfikatorów, czasie treningu i klasyfikacji oraz ocenie wydajności. Każdy obszar został zaprojektowany tak, by odpowiedzieć na konkretne pytania badawcze, tworząc pełen obraz możliwości i ograniczeń analizowanych modeli.

Dane przed wdrożeniem do sieci znormalizowano do odpowiedniego rozmiaru (32x32 pikseli dla modeli VGG i 224x224 pikseli dla modelu AlexNet). Następnie obrazy przekonwertowano na 3 kanały, jako że wszystkie badane modele działają na obrazach RGB. Na końcu nastąpiła normalizacja danych do zakresu od 0 do 1. Etykiety zostały zakodowane w systemie *one-hot* dla precyzyjnej klasyfikacji. Dane podzielono na zestaw treningowy i testowy w stosunku 80/20, aby zapewnić optymalne warunki do nauki modeli i rzetelne testowanie. W każdym eksperymencie zastosowano ustalone ziarno losowości, aby wyniki były spójne i porównywalne między iteracjami.

4.1. Badania wstępne

W ramach pierwszego scenariusza badawczego analizowano optymalną liczbę danych wejściowych niezbędnych do efektywnego działania modeli głębokiego uczenia. Przeprowadzono po siedem eksperymentów dla każdego z modeli, trenując je na zbiorach danych od 10 000 do 70 000 elementów, zwiększając liczbę, co 10 000. Parametry oceny obejmowały dokładność i stratę uzyskaną na zbiorach treningowych i walidacyjnych.

Przedmiotem drugiego scenariusza jest identyfikacja optymalnej liczby epok, potrzebnych do osiągnięcia maksymalnej dokładności przez modele, przy jednoczesnym minimalizowaniu ryzyka nadmiernego dopasowania. Modele wytrenowane zostały na ustalonej liczbie danych wejściowych z poprzedniego scenariusza, na 50 epokach.

Na podstawie tych danych określono wartości parametrów wejściowych, przy których model zachowuje kompromis między dokładnością klasyfikatora a zużyciem zasobów komputerowych oraz ryzykiem nadmiernego dopasowania.

4.2. Badania właściwe

Badania wstępne pozwoliły określić optymalną liczbę danych wejściowych oraz ilość epok treningowych dla każdego z modeli. Na podstawie tych danych ponownie wytrenowano sieci i skupiono się na pomiarze czasu, ocenie ich dokładności i wydajności.

W pierwszej kolejności przeprowadzono badanie mające na celu zmierzenie czasu treningu i klasyfikacji każdego z modeli. Celem tego scenariusza jest porównanie modeli pod kątem ich efektywności obliczeniowej, co ma kluczowe znaczenie przy wyborze modelu dla aplikacji wymagających szybkiej odpowiedzi.

Następne badanie skupia się na skonstruowaniu macierzy błędów reprezentujące wyniki klasyfikacji dokonanej przez konkretny model. Wizualna reprezentacja klasyfikacji modelu ma na celu wskazanie obszarów, w których model popełnia najwięcej błędów. Badanie to umożliwi bardziej szczegółową analizę działania konkretnego klasyfikatora i pozwoli na wyciągnięcie precyzyjniejszych wniosków.

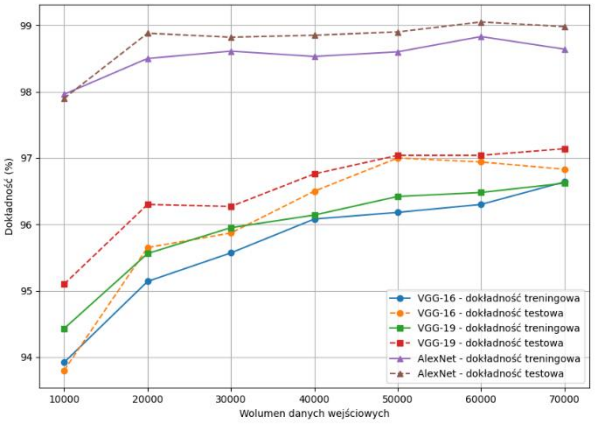
Na podstawie parametrów klasyfikacyjnych z macierzy pomyłek obliczone zostaną dodatkowo metryki w postaci precyzji, czułości i miary F1. Metryki te będą pomocne przy porównaniu modeli i określeniu, który z nich sprawdził się w tym zadaniu najlepiej.

Ostatnie badanie skupia się na analizie ROC i AUC. Krzywe ROC umożliwią oceną wydajności klasyfikatorów poprzez analizę zależności pomiędzy czułością a specyficznością. Wartość AUC reprezentująca pole pod krzywą ROC, zostanie obliczona dla każdej z klas, dostarczając zwięzłego wskaźnika jakości klasyfikatora, umożliwiając bezpośrednie porównanie wydajności wybranych modeli.

5. Wyniki badań

5.1. Analiza wolumenu danych wejściowych

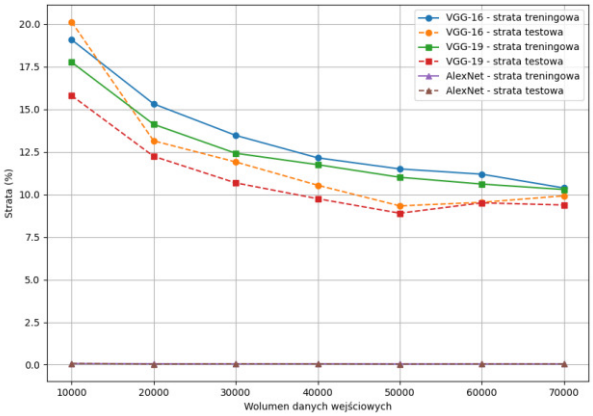
Analiza pierwszego scenariusza badawczego wskazuje na poprawę dokładności (Rysunek 2) i zmniejszenie straty (Rysunek 3) modeli wraz ze wzrostem ilości danych wejściowych dla wszystkich badanych modeli. W przypadku modeli VGG powyżej wartości 50 000 obrazów wejściowych, korzyści stają się mniej znaczące. Identyczna sytuacja ma miejsce powyżej wartości 30 000



dla modelu AlexNet. Przyjęcie 50 000 obrazów dla modeli VGG oraz 30 000 dla modelu AlexNet, jako optymalnej wielkości zbioru danych umożliwia uzyskanie wysokiej dokładności przy jednoczesnej minimalizacji straty, co wskazuje na zasadność ograniczenia dalszego

Rysunek 2: Wykres zbiorowy dokładności treningowych i testowych badanych modeli w funkcji liczby danych wejściowych.

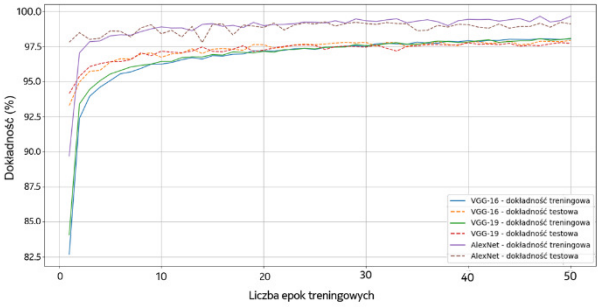
zwiększania wielkości zbioru w kontekście efektywności wykorzystania zasobów obliczeniowych.



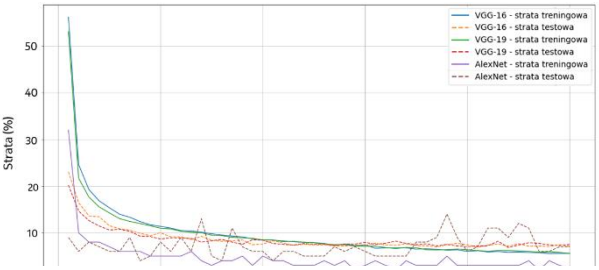
Rysunek 3: Wykres zbiorowy strat treningowych i testowych badanych modeli w funkcji liczby danych wejściowych.

5.2. Analiza zbieżności modelu

Na podstawie analizy zbieżności modeli wynika, że modele VGG osiągają stabilizację zarówno w dokładności (Rysunek 4) jak i w minimalnych stratach (Rysunek 5) po około 10 epokach.



Rysunek 4: Wykres zbiorowy dokładności treningowych i testowych badanych modeli w funkcji liczby epok.



Rysunek 5: Wykres zbiorowy strat treningowych i testowych badanych modeli w funkcji liczby epok.

Model AlexNet wymaga więcej epok, aby osiągnąć oczekiwaną stabilizację. Optymalne wyniki dla tej sieci zauważalne są po około 15 epokach. Trenowanie modeli przez więcej niż 10 epok dla modeli VGG oraz 15 epok dla modelu AlexNet, nie przyniesie znaczących korzyści, a jedynie zwiększy czas obliczeniowy. Dlatego, przyjęcie 10 epok dla modeli VGG-16 oraz VGG-19 i 15 dla modelu AlexNet jest wyborem optymalnym.

5.3. Pomiar czasu treningu i klasyfikacji

Wyniki pierwszego badania właściwego dotyczące pomiaru czasu treningu i klasyfikacji zostały przedstawione w tabeli 1. W przypadku trenowania modeli na karcie graficznej T4 GPU z dużą ilością pamięci RAM, modele VGG wykazują znacząco krótsze czasy treningu w porównaniu do treningu na CPU. Dla modelu VGG-16, czas treningu na GPU wynosi ok. 78 sekund, podczas gdy na CPU jest to aż 4247 sekund, co wskazuje na ponad 54-krotnie szybszy trening na GPU. Model AlexNet, będący bardziej złożonym modelem, również pokazuje znacząco lepsze wyniki na GPU. Średni czas treningu AlexNet na GPU wynosi 1209 sekund, natomiast na CPU jest to aż 54 030 sekund, co oznacza, że trening na CPU jest około 45 razy dłuższy niż na GPU.

Tabela 1: Średnie czasy treningu i klasyfikacji dla VGG-16, VGG-19 oraz AlexNet w różnych środowiskach operacyjnych

Model	Średni czas treningu [s]	Średni czas klasyfikacji [s]
T4 GPU + High RAM		
VGG16	*78*	*2*
VGG19	*100*	*3*
AlexNet	*1209*	*9*
CPU		
VGG16	*4 247*	*62*
VGG19	*4 594*	*95*
AlexNet	*54 030*	*204*

Analiza wyników wyraźnie wskazuje na dużą przewagę GPU nad CPU w kontekście głębokiego uczenia. Czasy treningu i klasyfikacji są znacznie krótsze na GPU dla wszystkich badanych modeli, co oznacza bardziej efektywne wykorzystanie zasobów obliczeniowych i skrócenie czasu potrzebnego na uzyskanie wyników. W praktycznych zastosowaniach, gdzie czas obliczeń ma kluczowe znaczenie, wykorzystanie GPU staje się nie tylko korzystne, ale wręcz konieczne. Przewaga GPU nad CPU jest szczególnie widoczna w przypadku bardziej złożonych modeli, takich jak AlexNet, gdzie różnice w czasach obliczeniowych są najbardziej znaczące.

5.4. Ocena dokładności klasyfikatorów

Model VGG-19 (Rysunek 6) podobnie jak VGG-16 i AlexNet wykazuje wysoką zdolność do poprawnej klasyfikacji. W przypadku modelu VGG-19 model 23 razy sklasyfikował cyfrę, 9 jako 8 i 18 razy sklasyfikował, 5 jako 2. Wartości te jednak w porównaniu do tych *TP* są znikome, biorąc pod uwagę, że 850 razy model sklasyfikował klasę 8 poprawnie.

Wyniki ewaluacji na podstawie macierzy pomyłek wskazują, że model AlexNet przewyższa modele VGG-16 i VGG-19 pod względem precyzji, czułości i F1-score (Tabela 2). Chociaż różnice między modelami VGG-16 i VGG-19 są niewielkie, AlexNet wyraźnie dominuje, osiągając najwyższe wyniki we wszystkich metrykach. Wysokie wartości precyzji, czułości i F1-score dla AlexNet sugerują, że jest to model najbardziej skuteczny w dokładnej klasyfikacji obrazów, minimalizując zarówno fałszywe pozytywy, jak i fałszywe negatywy.

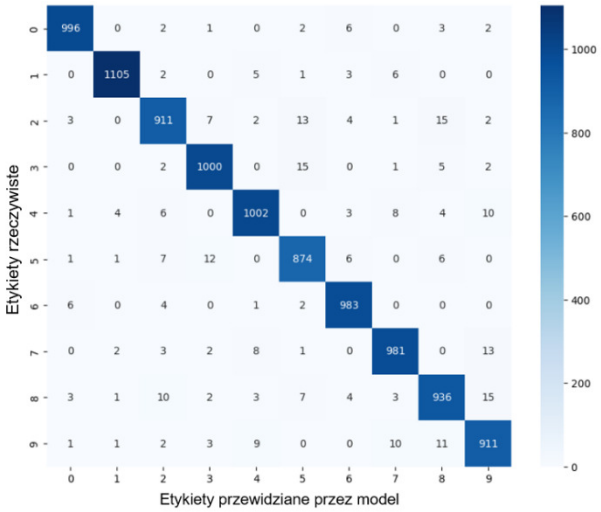
AlexNet demonstruje najwyższą wydajność w zadaniu klasyfikacji obrazów, co czyni go preferowanym modelem w porównaniu do VGG-16 i VGG-19, jeśli

priorytetem jest maksymalizacja precyzji, czułości i ogólnej dokładności klasyfikacji.

Tabela 2: Metryki oceny modeli VGG-16, VGG-19 oraz AlexNet

Metryka	VGG-16	VGG-19	AlexNet
Precyzja	*96,83%*	*96,93%*	*98,76%*
Czułość	*96,78%*	*96,95%*	*98,72%*
F1-score	*96,80%*	*96,94%*	*98,74%*

Macierz pomyłek dla modelu VGG-16 potwierdza wysoką zdolność do poprawnego przewidywania klasy przynależności. Błędy klasyfikacji, sugerują, że model może mieć trudności ze zróżnicowaniem między klasami o zbliżonych cechach. Częstość zmiany między klasą 2 i 3 oraz 2 i 5 może wskazywać na podobieństwo w cechach, które model interpretuje, jako wspólne dla tych klas. Podobnie do modelu VGG-16, jego następca VGG-19 również wykazał wysoką zdolność do poprawnej klasyfikacji (Rysunek 6). W tym przypadku model 23 razy sklasyfikował cyfrę 9, jako 8 i 18 razy 5, jako 2. Wartości te jednak w porównaniu do tych TP są znikome, biorąc pod uwagę, że 850 razy model sklasyfikował 8 poprawnie. Model AlexNet wykazał się jednak najniższą tolerancją na błędy. Jedynie 5 razy błędnie sklasyfikował, 4 jako 9 oraz 4 razy pomylił 7 z 9. Reszta pomyłek dotyczy pojedynczych wartości, względem *TP* na poziomie w zakresie od 564 do 654 dla 10 klas.



Rysunek 6: Macierz pomyłek dla modelu VGG-19.

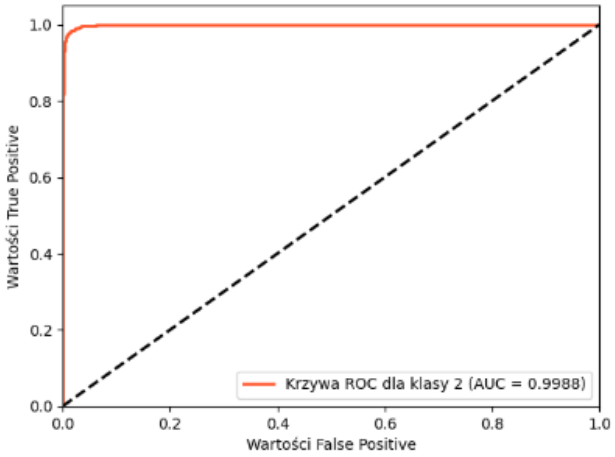
5.5. Ocena wydajności modeli

Dla każdej z klas każdego modelu obliczono AUC (Tabela 3) i zwizualizowano krzywe ROC. Wszystkie wartości AUC są bliskie zeru, co za tym idzie każda z krzywych ROC niemal idealnie przylega do krawędzi wykresu, co oznacza, że modele mają bardzo wysoką czujność oraz bardzo niski odsetek fałszywie pozytywnych wyników dla wszystkich klas.

Tabela 3: Wartości AUC dla modeli VGG-16, VGG-19 i AlexNet

Klasa	VGG-16	VGG-19	AlexNet
0	*99,99%*	*99,99%*	*100%*
1	*99,99%*	*99,99%*	*99,9%*
2	*99,88%*	*99,84%*	*99,9%*
3	*99,89%*	*99,91%*	*100%*
4	*99,94%*	*99,93%*	*100%*
5	*99,88%*	*99,91%*	*100%*
6	*99,98%*	*99,98%*	*100%*
7	*99,96%*	*99,93%*	*100%*
8	*99,91%*	*99,87%*	*99,9%*
9	*99,90%*	*99,92%*	*99,9%*
Średnia AUC	*99,93 p.p.*	*99,93 p.p.*	*99,96 p.p*

Rysunek 7 przedstawia przykładową krzywą ROC dla klasy 2 modelu VGG-16. Pomarańczowa linia niemal przylega do krawędzi wykresu, co świadczy o wysokiej skuteczności modelu. Krzywe ROC dla wszystkich klas wszystkich modeli mają podobny przebieg. W przypadku modelu AlexNet, krzywa ROC znajduje się jeszcze bliżej idealnej krawędzi, co wskazuje na jego nieco wyższą wydajność w klasyfikacji [10].



Rysunek 7: Krzywa ROC dla klasy 2 modelu VGG-16.

6. Podsumowanie

Jednym z głównych czynników wpływających na wymagania pamięci obliczeniowej jest rozmiar wejściowego obrazu. Sieć AlexNet przyjmuje obrazy o rozmiarze 224x224 pikseli, podczas gdy modele VGG zostały dostosowane do rozmiaru 32x32 pikseli. Większy rozmiar obrazu w AlexNet zwiększa liczbę operacji konwolucyjnych i wymaga więcej pamięci do przechowywania map cech, co może prowadzić do dłuższego czasu treningu w porównaniu do modeli VGG, nawet, jeśli AlexNet ma mniej warstw.

Optymalizacja architektury sieci neuronowych może obejmować stosowanie technik regularyzacyjnych, takich jak *dropout*, oraz odpowiednie dostosowanie hiperparametrów, takich jak liczba epok i rozmiar partii. Zmniejszenie rozmiaru *batch* również może zmniejszyć zapotrzebowanie na pamięć podczas treningu. Ponadto, wykorzystanie architektur hybrydowych, które łączą cechy różnych modeli, może prowadzić do lepszej wydajności. Ciągłe badania nad nowymi technikami, takimi jak sieci rekurencyjne czy modele transformatorów, mogą dostarczyć dalszych usprawnień.

Warto zaznaczyć, że badane modele zostały pierwotnie zaprojektowane do klasyfikacji znacznie bardziej skomplikowanych obrazów niż te analizowane

w niniejszym artykule Modele CNN użyte do klasyfikacji cyfr, takie jak VGG-16, VGG-19 i AlexNet, są zbyt rozbudowane dla tego zadania, co sprawia, że wyniki tych badań mają charakter wyłącznie eksploracyjny i nie nadają się do bezpośredniego zastosowania w rzeczywistych scenariuszach.

7. Wnioski

Wyniki przedstawione w niniejszym artykule wskazują, że AlexNet zapewnia najwyższą dokładność w klasyfikacji cyfr pisma ręcznego z zestawu danych MNIST, pomimo dłuższego czasu treningu na GPU T4 (ok. 1209 sekundy) w porównaniu do VGG-19 (ok. 100 sekund) i VGG-16 (ok. 78 sekund). Model AlexNet wykazuje większą efektywność, wymagając mniej danych wejściowych (30 000 obrazów) niż modele VGG (50 000 obrazów) do osiągnięcia podobnej dokładności. Ponadto, AlexNet osiąga lepsze wyniki w zakresie precyzji, czułości oraz miary F1, osiągając wartości odpowiednio: 98,76%, 98,72% oraz 98,74%. Średnia wartość AUC dla modelu AlexNet jest wyższa o 2 p.p. w porównaniu z konkurentami, osiągając poziom 99,96%, podczas gdy modele VGG-16 osiągają wartość AUC na poziomie 99,93%.

Stabilność modelu AlexNet po dłuższym treningu jest kluczowa dla rzeczywistych zastosowań, mimo że wymaga on więcej epok do stabilizacji. Model ten oferuje lepszą równowagę między dokładnością a zasobami obliczeniowymi, co czyni go bardziej skutecznym modelem w praktycznych zastosowaniach.

Wyniki dla modeli VGG były tylko nieznacznie niższe. Jeśli w praktyce większe znaczenie ma czas treningu i klasyfikacji niż dokładność, zaleca się użycie modeli VGG, których czas treningu jest nawet o 128 razy szybszy niż AlexNet (trenując modele na zwykłym CPU).

Literatura

[1] A. Przegalińska, L. Ciechanowski, Wykorzystanie algorytmów sztucznej inteligencji w instytucjach kultury, Ministerstwo Kultury i Dziedzictwa Narodowego, Ekspertyza Ministerialna, Warszawa, 2019-2020.

[2] P. Norvig, S. Russel, Artificial Intelligence: a modern approach, Fourth Edition, Pearson, University of California at Berkeley, 2021.

[3] K. Simonyan, A. Zisserman, Very Deep convolutional networks for large-scale image recognition, In International Conference on Learning Representations (ICLR) 6 (2016) 1-11.

[4] A. Krizhevsky, I. S Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Communications of the ACM 6 (2017) 84-90.

[5] Lecture MIT 6.S191 (2023): Convolutional Neural Networks, Massachusetts Institute of Technology https://www.youtube.com/watch?v=NmLK_WQBxB4 [01.03.2024]

[6] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition in Proceedings of the IEEE 11 (1998) 2278-2324.

[7] A. Baldominos, Y. Saez, P. Isasi, A Survey of handwritten character recognition with MNIST and EMNIST, Applied Sciences 9 (2019) 15-31.

[8] G. S. Handelman, et al. Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods, American Journal of Roentgenology 1 (2019) 38-43.

[9] J. Lever, M. Krzywinski, N. Altman, Classification evaluation, Nature Methods 13 (2016) 603-604.

[10] S. Mascarenhas, M. Agarwal, A comparison between VGG16, VGG19 and ResNet50, architecture frameworks for Image Classification In International conference on disruptive technologies for multi-disciplinary research and applications (Centon) IEEE 1 (2021) 96-99.