Grupa 2 Środa 17:05	Algorytmy sztucznej inteligencji w przemyśle 4.0
Dawid Jordan 255330 Natalia Stępień 254978	Temat: Konwolucyjne sieci neuronowe - CIFAR

Sprawozdanie

1. Krótki wstęp teoretyczny.

Klasyfikacja obrazów jako proces przypisywania obrazom określonych kategorii na podstawie ich zawartości jest jednym z kluczowych zagadnień w uczeniu maszynowym. Chętnie stosuje się tu sztuczną inteligencję i widzenie komputerowe. Przykłady obejmują między innymi: rozpoznawanie obiektów, diagnostykę medyczną, systemy bezpieczeństwa, analizę obrazów satelitarnych, a także pojazdy autonomiczne.

Podstawowe kroki w klasyfikacji obrazów obejmują:

- Zbieranie i wstępne przetwarzanie obrazów – zmiana rozmiaru, normalizacja wartości pikseli czy augmentacja danych
- Ekstrakcję cech, która w konwolucyjnych sieciach neuronowych wykonuje się automatycznie
- Uczenie modelu – model trenowany jest na zbiorze danych (np. CIFAR-10, CIFAR-100, ImageNet), często wykorzystując różne metody optymalizacji dla minimalizacji błędu klasyfikacji takie jak gradient descent
- Predykcję oraz ocenę modelu – obraz przypisywany jest przez model do jednej z dostępnych klas, na koniec ocenia się jakość klasyfikacji na podstawie różnych metryk (dokładność (accuracy), precyzja (precision), czułość (recall), F1-score)

Podstawową architekturą stosowaną w klasyfikacji obrazów są **konwolucyjne sieci neuronowe** (Convolutional Neural Networks, CNN), czyli rodzaj sztucznych sieci neuronowych, które dzięki swojej strukturze i specjalnym warstwom umożliwiają skuteczne rozpoznawanie wzorców i cech w obrazach, co czyni je niezastąpionym narzędziem we wspomnianej już klasyfikacji obrazów, wykrywaniu obiektów, segmentacji oraz innych zastosowaniach związanych z widzeniem komputerowym. CNN nie wymagają ręcznego projektowania cech, ponieważ samodzielnie uczą się najważniejszych cech obrazu w trakcie treningu, co w ogólnym przypadku prowadzi do znacznie lepszych wyników i możliwości generalizacji do nowych danych.

CNN składają się z kilku charakterystycznych warstw, które współpracują w celu ekstrakcji istotnych cech obrazu i przypisania go do odpowiedniej kategorii.

Warstwa konwolucyjna stanowi podstawowy element sieci CNN i jest odpowiedzialna za wykrywanie cech na różnych poziomach obrazu. Działa poprzez zastosowanie filtrów (nazywanych także jądrami lub maskami) o określonych rozmiarach, które przesuwają się po obrazie i wykonują operację matematyczną zwaną splotem (convolution). Filtry te uczą się rozpoznawać istotne elementy obrazu, takie jak krawędzie, tekstury, a na głębszych poziomach – bardziej złożone struktury, np. fragmenty obiektów. Dzięki takiej budowie CNN mogą skutecznie analizować obrazy w sposób hierarchiczny – pierwsze warstwy wykrywają podstawowe cechy, a kolejne łączą je w bardziej skomplikowane reprezentacje.

Po każdej warstwie konwolucyjnej stosuje się funkcję aktywacji, najczęściej ReLU (Rectified Linear Unit), która wprowadza nieliniowość do modelu i sprawia, że wartości ujemne zostają zastąpione zerami, natomiast wartości dodatnie pozostają niezmienione. Dzięki temu sieć może lepiej odwzorować złożone zależności w danych.

Warstwa normalizacji jak sama nazwa wskazuje pozwala normalizować dane przechodzące przez sieć, co przyspiesza proces uczenia i stabilizuje jego przebieg. Jest to istotne, ponieważ pozwala uniknąć problemów związanych z dużymi wahaniami wartości aktywacji neuronów.

W warstwie próbkowania następuje operacja redukcji wymiarów danych, która zmniejsza liczbę parametrów sieci i zapobiega przeuczeniu (overfitting). Najczęściej stosowany jest Max Pooling, który wybiera największą wartość z określonego okna (np. 2×2). Alternatywnie można zastosować Average Pooling, który oblicza średnią wartość z okna.

Warstwa w pełni połączona odpowiada za to, żeby cechy obrazu były spłaszczane (flattening) i przekazywane do w pełni połączonych warstw neuronów po przejściu przez warstwy konwolucyjne i poolingowe. Ma za zadanie interpretację wyodrębnionych cech i dokonanie ostatecznej klasyfikacji obrazu.

Warstwa wyjściowa to ostatni element sieci. Zwraca prawdopodobieństwo przynależności obrazu do poszczególnych klas. Dla zadań klasyfikacji wieloklasowej stosuje się funkcję aktywacji softmax, która zwraca rozkład prawdopodobieństw dla każdej klasy.

Teoretycznie CNN powinny zapewnić lepszą generalizację danych oraz efektywność obliczeniową.

CIFAR (Canadian Institute for Advanced Research**)** to instytut badawczy zajmujący się zaawansowanymi zagadnieniami sztucznej inteligencji i uczenia maszynowego. W kontekście widzenia komputerowego i klasyfikacji obrazów termin CIFAR odnosi się do dwóch popularnych zestawów danych: **CIFAR-10** i **CIFAR-100**, które są szeroko wykorzystywane w badaniach nad sieciami neuronowymi i ich zastosowaniami w analizie obrazów. Często używa się ich jako benchmarków do oceny skuteczności różnych modeli klasyfikacyjnych sieci CNN. Od prostych zbiorów typu MNIST odróżnia je obecność koloru na obrazach (MNIST zawiera czarno-białe obrazy).

CIFAR-10 posiada 10 klas obiektów (samolot, samochód, ptak, kot, jeleń, pies, żaba, koń, statek, ciężarówka), przy czym liczba obrazów należących do zbioru treningowego wynosi 50 000, a do testów 10 000. Rozmiar zawartych w nim obrazów to 32×32 piksele, 3 kanały (RGB).

CIFAR-100 posiada 100 klas obiektów zgrupowanych w 20 nadklas (przykład nadklasy: pojazdy, zwierzęta domowe, owady, przykład klasy: pies, kot, motocykl, autobus), przy czym liczba obrazów należących do zbioru treningowego wynosi 50 000, a do testów 10 000. Rozmiar zawartych w nim obrazów to 32×32 piksele, 3 kanały (RGB).

CIFAR-100 jest trudniejszy do klasyfikacji niż CIFAR-10, ponieważ zawiera większą ilość klas i większą różnorodność danych przy zachowaniu tej samej liczby obrazów.

ResNet50 (*Residual Network-50*) to głęboka sieć konwolucyjna zaprojektowana w celu rozwiązania problemu **zanikającego gradientu**, który utrudnia trenowanie bardzo głębokich sieci neuronowych. ResNet50, opracowany przez **Kaiminga He i współautorów**, wykorzystuje mechanizm **połączeń resztkowych** (*skip connections*), który pozwala na propagację informacji między warstwami, co znacząco poprawia skuteczność uczenia się.

ResNet50 jest szeroko stosowany w zadaniach klasyfikacji obrazów, w tym w zbiorach CIFAR-10 oraz CIFAR-100, które stanowią standardowe benchmarki w uczeniu maszynowym. Jednakże, ze względu na różnice w liczbie klas i poziomie skomplikowania zbiorów, dostosowanie ResNet50 do tych danych wymaga zastosowania odpowiednich technik fine-tuningu i regularyzacji.

składa się z 50 warstw głębokości i jest oparty na tzw. blokach resztkowych, które zawierają operacje konwolucyjne, normalizację Batch Normalization oraz funkcję aktywacji ReLU. W przeciwieństwie do tradycyjnych sieci, w których dane przechodzą w sposób sekwencyjny przez kolejne warstwy, w ResNet50 zastosowano połączenia skrótowe, które umożliwiają omijanie niektórych warstw i sumowanie wartości aktywacji. Dzięki temu sieć może być znacznie głębsza bez problemu zanikania gradientu.

Struktura ResNet50:

- Warstwa wejściowa – konwolucja 7×7 z filtrem 64 i MaxPooling.
- Bloki resztkowe:
    - Blok 1: 3 warstwy konwolucyjne (64, 64, 256)
    - Blok 2: 4 warstwy (128, 128, 512)
    - Blok 3: 6 warstw (256, 256, 1024)
    - Blok 4: 3 warstwy (512, 512, 2048)
- Warstwa wyjściowa – globalne uśrednianie cech (*Global Average Pooling*) i warstwa Dense (softmax).

ResNet50 może być trenowany od podstaw lub wykorzystywany w technice transfer learning, co pozwala na użycie wstępnie nauczonych wag z zestawu ImageNet i dostosowanie ich do nowych danych.

Adaptacja ResNet50 do CIFAR-10:

Oczekiwane wyniki: przy odpowiedniej konfiguracji ResNet50 osiąga ~93-96% dokładności na CIFAR-10.

Adaptacja ResNet50 do CIFAR-100:

Oczekiwane wyniki: Przy odpowiedniej optymalizacji ResNet50 może osiągnąć ~50-60% dokładności na CIFAR-100.

2. Cel badań

Cel badań dotyczył zaimplementowania i porównania działania dwóch relatywnie prostych sieci CNN klasyfikujących obrazy zawarte w bazach CIFAR-10 i CIFAR-100 oraz porównanie otrzymanych wyników i charakterystyk jakości uczenia. Do tego zastosowany został gotowy model ResNet50 dostępny w bibliotece keras dla porównania jakości uczenia, przy czym dla każdej z baz danych, tj CIFAR-10 i CIFAR-100 dostosowano hiperparametry do poziomu skomplikowania problemu.

3. Kod programu

Oba programy wczytują dane z zbioru Cifar-100 jako 60 000 kolorowych obrazków o wymiarach 32x32 piksele, przyporządkowanych do 100 klas. Obrazki są normalizowane i w tym celu skalowane do zakresu [0,1]. Etykiety zapisane jako liczby (od 0 do 99) są zamieniane na wektory binarne, aby otrzymać format kompatybilny do obliczania funkcji strat. Programy są wyposażone również w augmentację danych w postaci generatora, który losowo modyfikuje obrazy o funkcje rotacji, przesunięcia czy odbicia lustrzanego. Rotacja była losowo wybierana z maksimum 15 stopni, przesunięcia poziome i pionowe o 10% szerokości bądź wysokości. Generator jest również dostosowany do znormalizowanych obrazów za pomocą funkcji fit(x_train).

Model konwolucyjnej sieci neuronowej (CNN) składa się z:

- Trzech bloków z warstwami Conv2D (o 32, 64 i 128 filtrach) z funkcją aktywacji ReLU. Dodatkowo, każda warstwa konwolucyjna ma nałożony regularyzator L2, który pomaga ograniczyć przeuczenie (overfitting).
- Po każdej warstwie konwolucyjnej stosowana jest warstwa normalizująca dane, dla stabilizacji treningu.
- Po bloku konwolucyjnym następuje redukcja wymiarów w warstwie MaxPooling2D.
- Warstwy Flatten spłaszczające dane przed podaniem ich do warstwy gęstej.
- Warstwa gęsta (dense) z 256 neuronami i funkcją aktywacji ReLU, oraz dropout o współczynniku 0.5, który losowo wyłącza połowę neuronów w celu redukcji dopasowania.
- Warstwa gęsta z 100 neuronami (po jednym dla każdej klasy w CIFAR-100) i funkcją aktywacji softmax, która przekształca wyjścia na rozkład prawdopodobieństwa.
- Model jest kompilowany z użyciem optymalizatora Adam (learning rate ustawiony na 0.001), funkcji strat categorical_crossentropy.

Model wyposażony jest dodatkowo w callbacki monitorujące wartość strat walidacyjnych i zmniejsza learning rate (współczynnik = 0.5) jeżeli przez 2 epoki nie nastąpi poprawa jej wartości oraz callback, który zatrzymuje trening jeśli przez 4 kolejne epoki nie nastąpi poprawa strat walidacyjnych. Dodatkowo przywraca najlepsze wagi do modelu po jego zatrzymaniu.

Program tworzy wykresy zmian dokładności i funkcji straty dla zbioru treningowego oraz walidacyjnego. Po zakończeniu treningu model oceniany jest na niezmodyfikowanych danych testowych, określana jest końcowa strata i dokładność.

Drugi program zawiera implementację przewidywania obrazów z bazy Cifar-100 z wykorzystaniem gotowego modułu ResNet50. Program działa na podobnej zasadzie co poprzednio opisany kod, z modelem konwolucyjnej sieci neuronowej, z kilkoma różnicami. Dane

wejściowe są dodatkowo przygotowywane do wymagań sieci za pomocą funkcji preprocess_input. W augmentacji danych dodano dodatkowo możliwy zoom, co pomaga zwiększyć różnorodność danych.

Model zbudowany jest na podstawie ResNet50, używanego jako model bazowy. Wykorzystano wagi wytrenowane na ImageNet, wyuczone do cech ogólnych. Na początku cała sieć bazowa jest zamrożona a jej wagi nie są aktualizowane przez pierwszy trening. Model bazowy został dodatkowo wyposażony w kilka warstw:

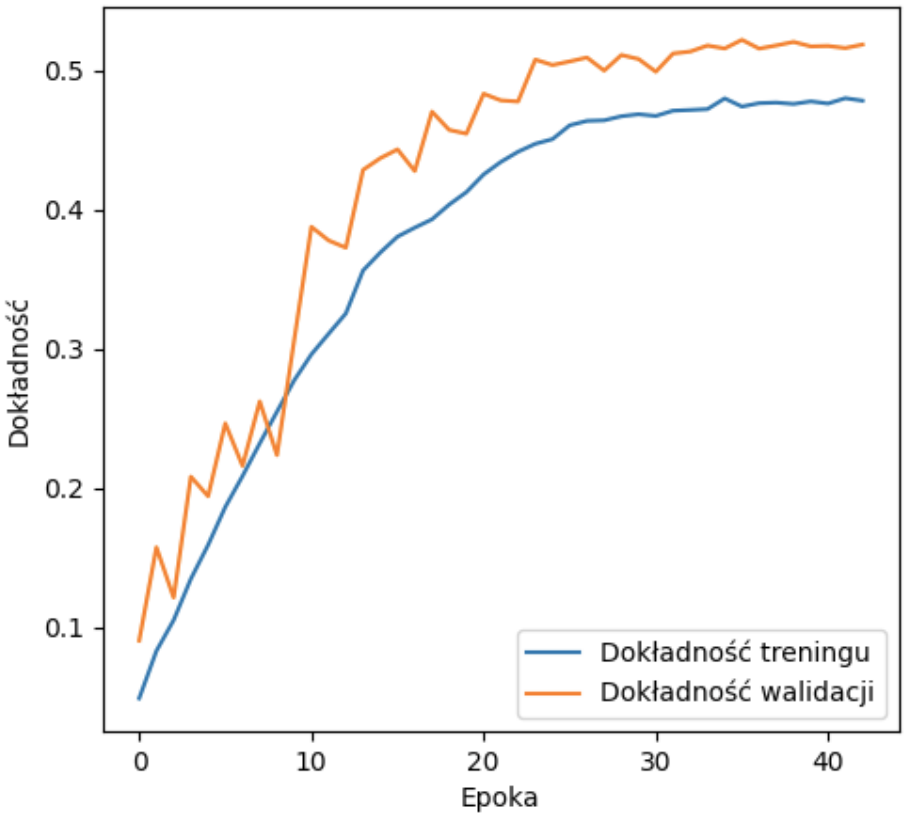
- Normalizująca aktywację, BatchNormalization.
- Zmniejszająca wymiary tensorów, GlobalAveragePooling2D.
- Warstwy gęste z funkcją aktywacji ReLU oraz warstwy droput.
- Warstwa gęsta z 100 neuronami i funkcją aktywacji softmax.

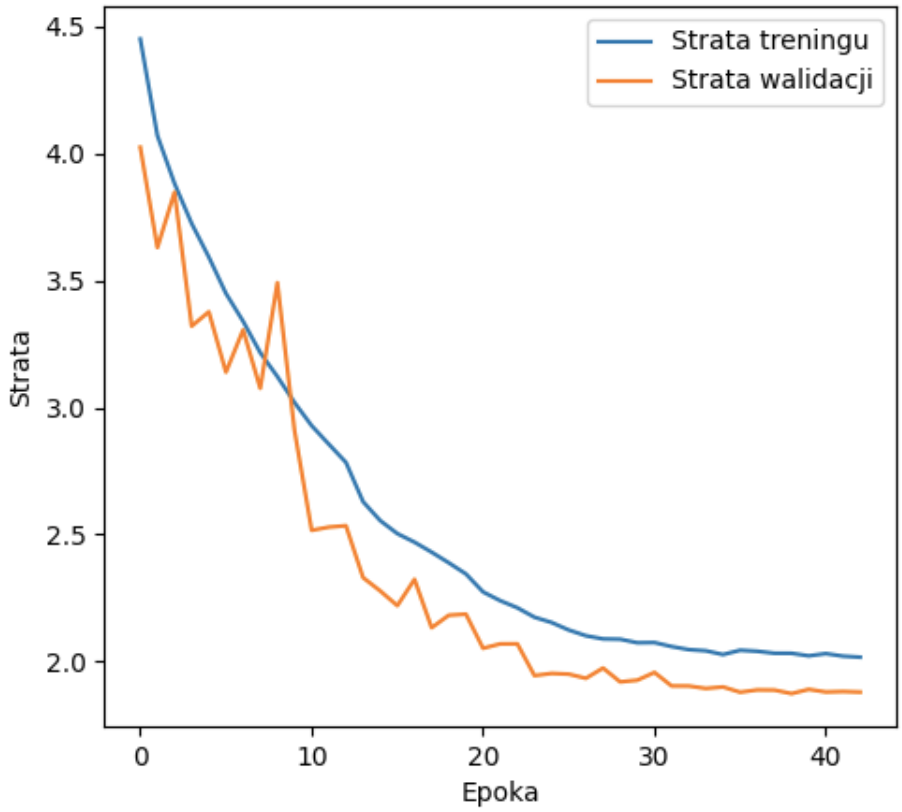
Ten model również wyposażono w callbacki, z tą różnicą, że callback sprawdzający wartość straty walidacyjnej zmniejsza learning rate w przypadku braku zmiany wartości przez 5 epok, zaś callback odpowiedzialny za zatrzymanie treningu działa w przypadku braku poprawy wartości straty przez 10 epok. Model jest zapisywany do pliku, jeżeli dokładność wyniesie maksimum.

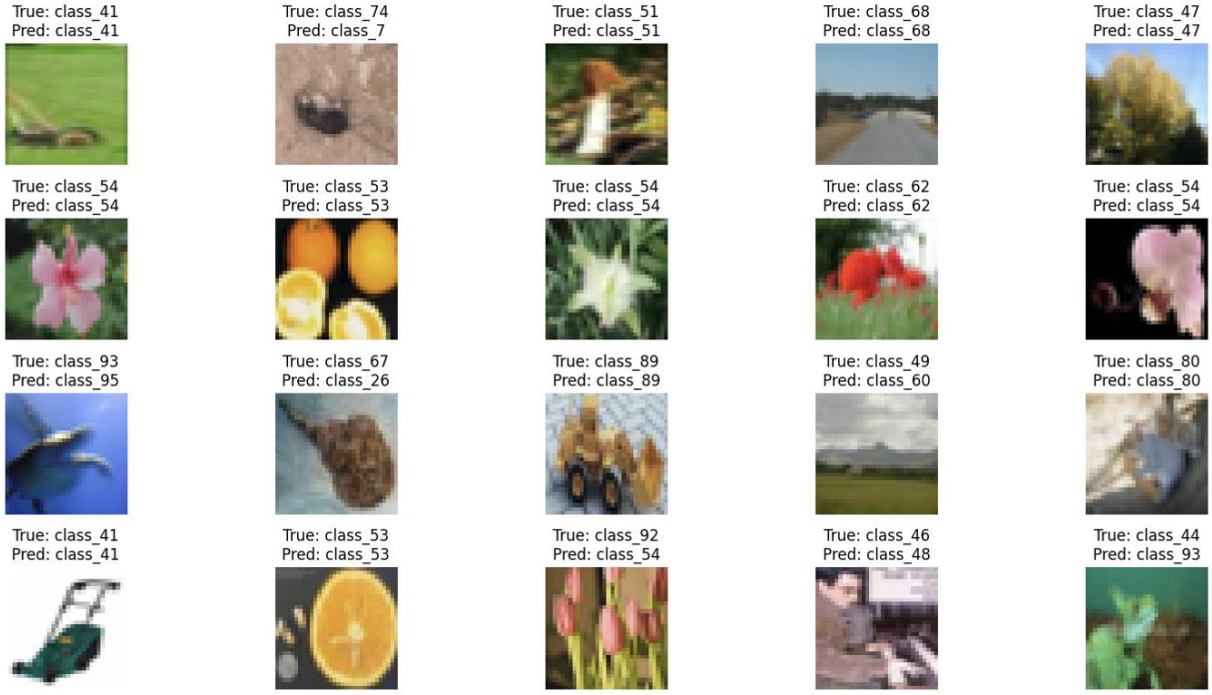
Pierwszą fazę treningu modelu rozpoczęto treningu przez 20 epok, w tej fazie wagi sieci bazowej ResNet50 zamrożono i nie ulegają modyfikacji. W drugiej fazie odmrożono wagi sieci bazowej (z wyłączeniem dolnych 100 warstw), aby uzyskać dostosowanie modelu ResNet50 do zadania Cifar-100, zachowując cechy treningu z ImageNet. Trening w drugiej fazie trwa 70 epok. Learning rate został obniżony ze względu na specyfikę treningu metodą fine-tuning. Historia jest zapisywana w postaci metryk dokładności i strat w obu fazach treningu osobno.

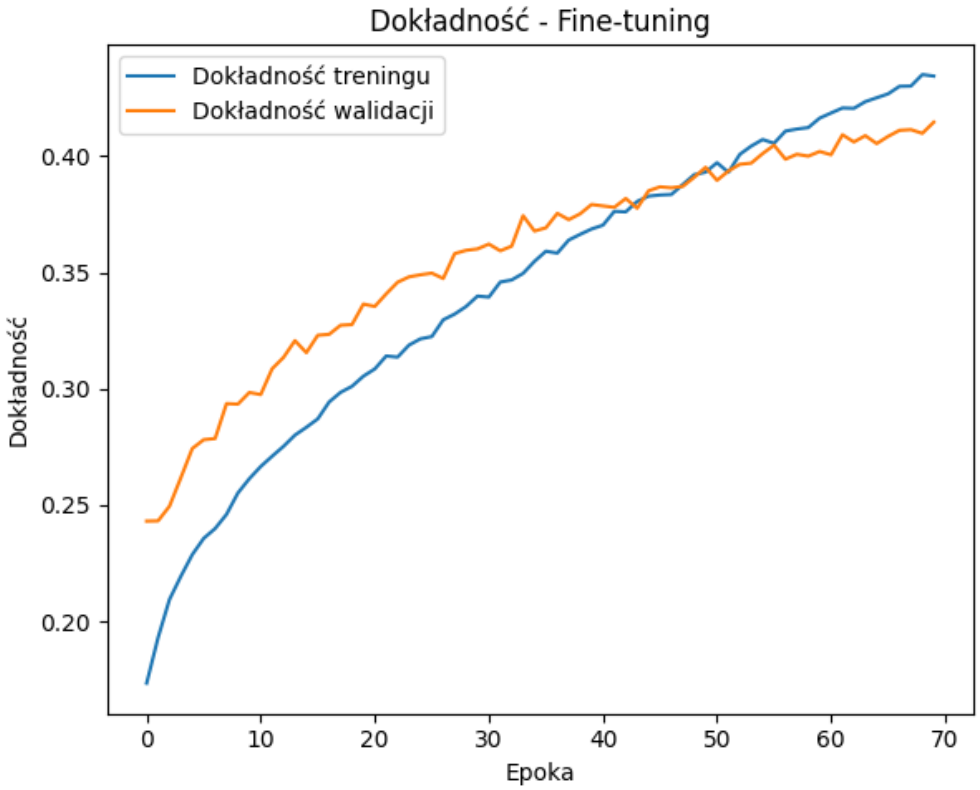
4. Wyniki badań

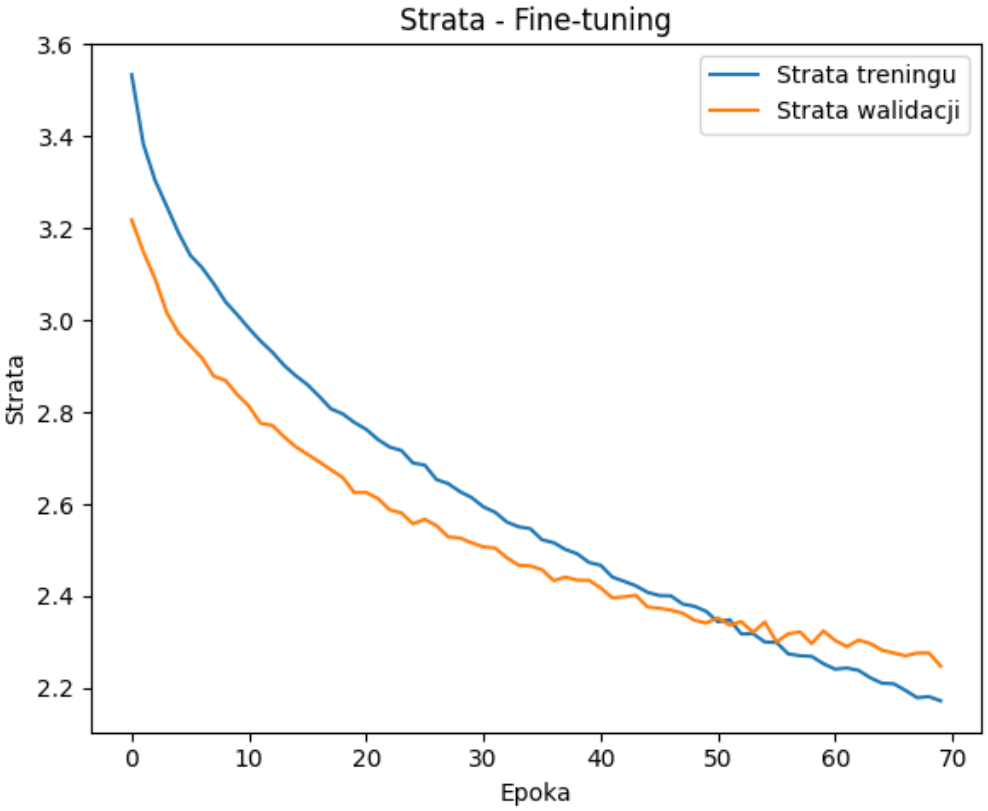
a. Cifar100

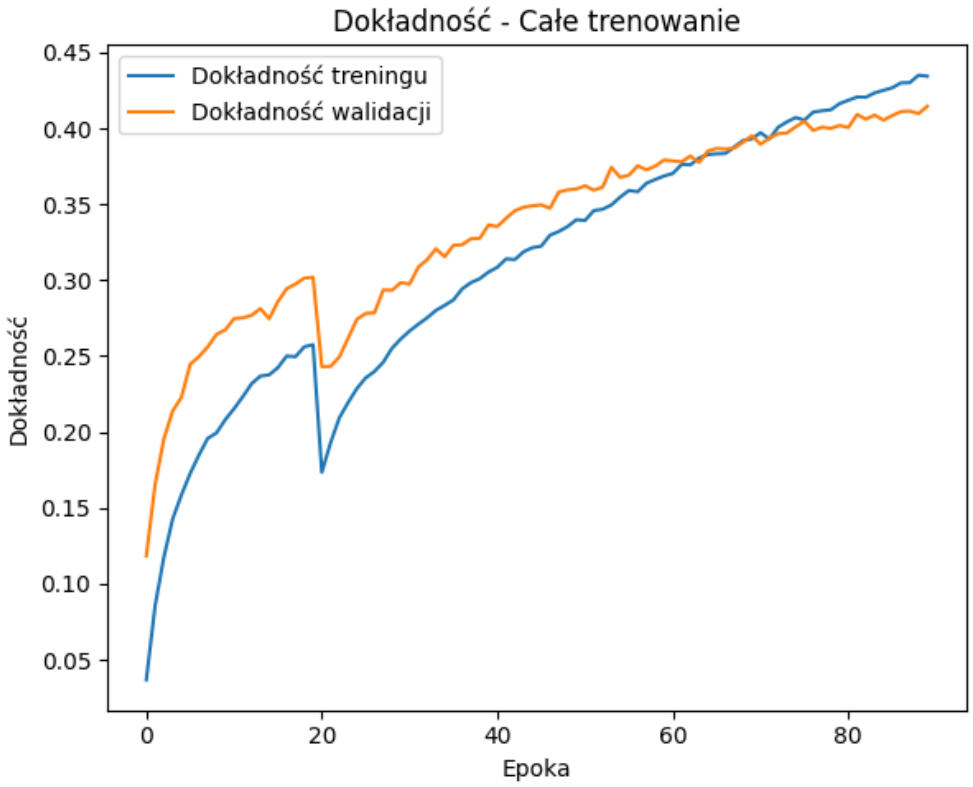


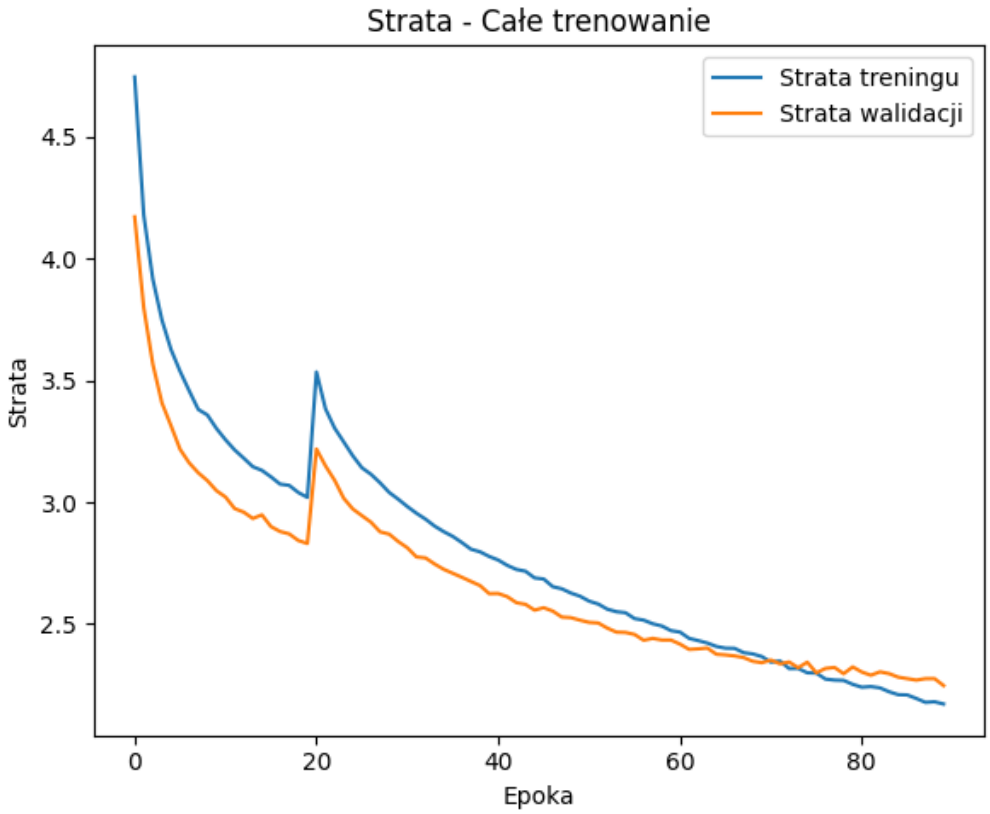


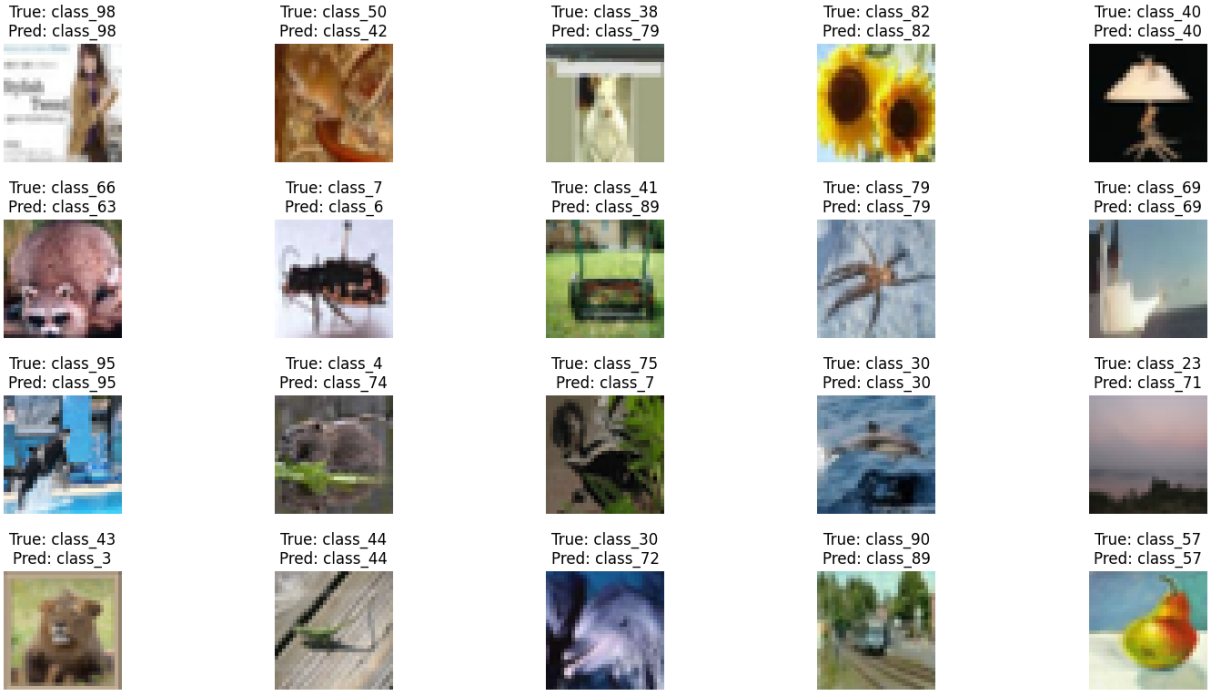

True: class_41
Pred: class_41
True: class_74
Pred: class_7
True: class_51
Pred: class_51
True: class_68
Pred: class_68
True: class_47
Pred: class_47
True: class_54
Pred: class_54
True: class_53
Pred: class_53
True: class_54
Pred: class_54
True: class_62
Pred: class_62
True: class_54
Pred: class_54
True: class_93
Pred: class_95
True: class_67
Pred: class_26
True: class_89
Pred: class_89
True: class_49
Pred: class_60
True: class_80
Pred: class_80
True: class_41
Pred: class_41
True: class_53
Pred: class_53
True: class_92
Pred: class_54
True: class_46
Pred: class_48
True: class_44
Pred: class_93


Dokładność - Fine-tuning
Dokładność treningu
Dokładność walidacji
Dokładność
0.20
0.25
0.30
0.35
0.40
0
10
20
30
40
50
60
70
Epoka

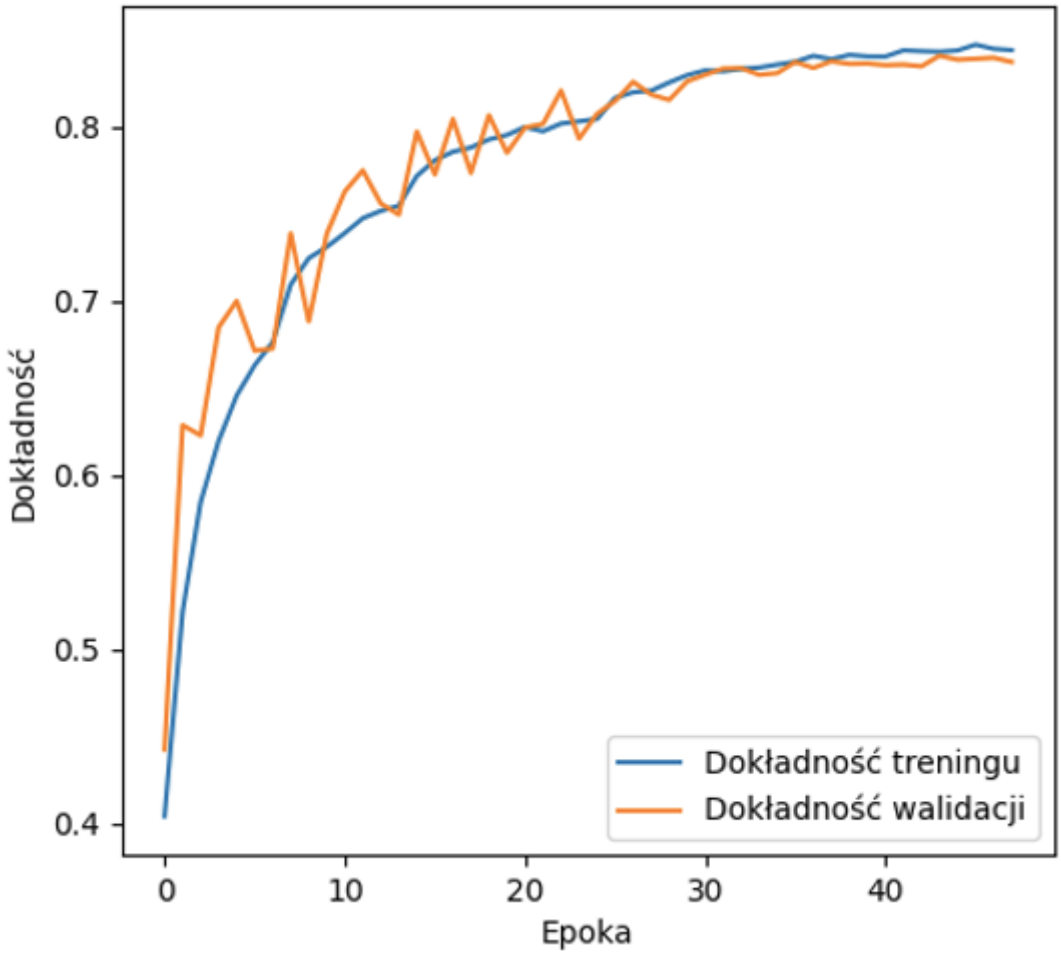

Strata - Fine-tuning
Strata treningu
Strata walidacji
Strata
Epoka
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
0
10
20
30
40
50
60
70

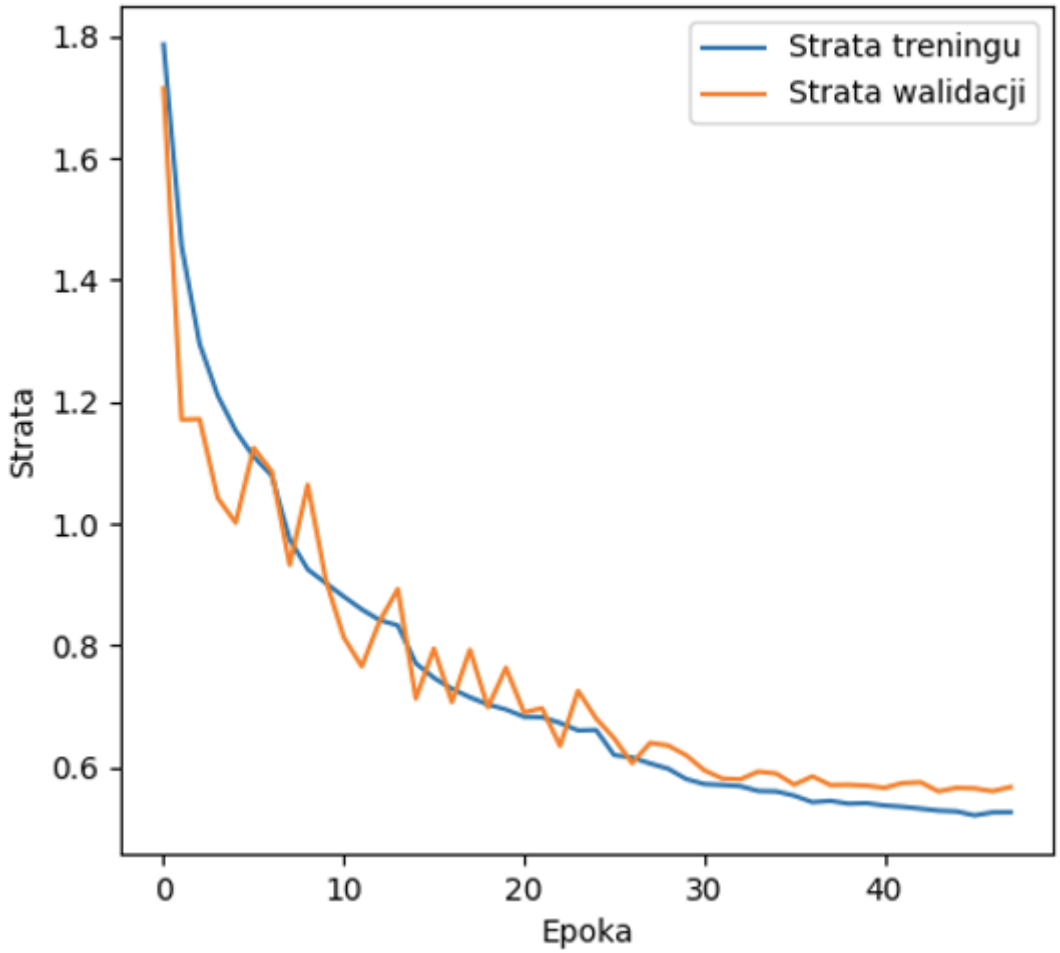

Dokładność - Całe trenowanie
Dokładność treningu
Dokładność walidacji
Dokładność
Epoka
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
20
40
60
80

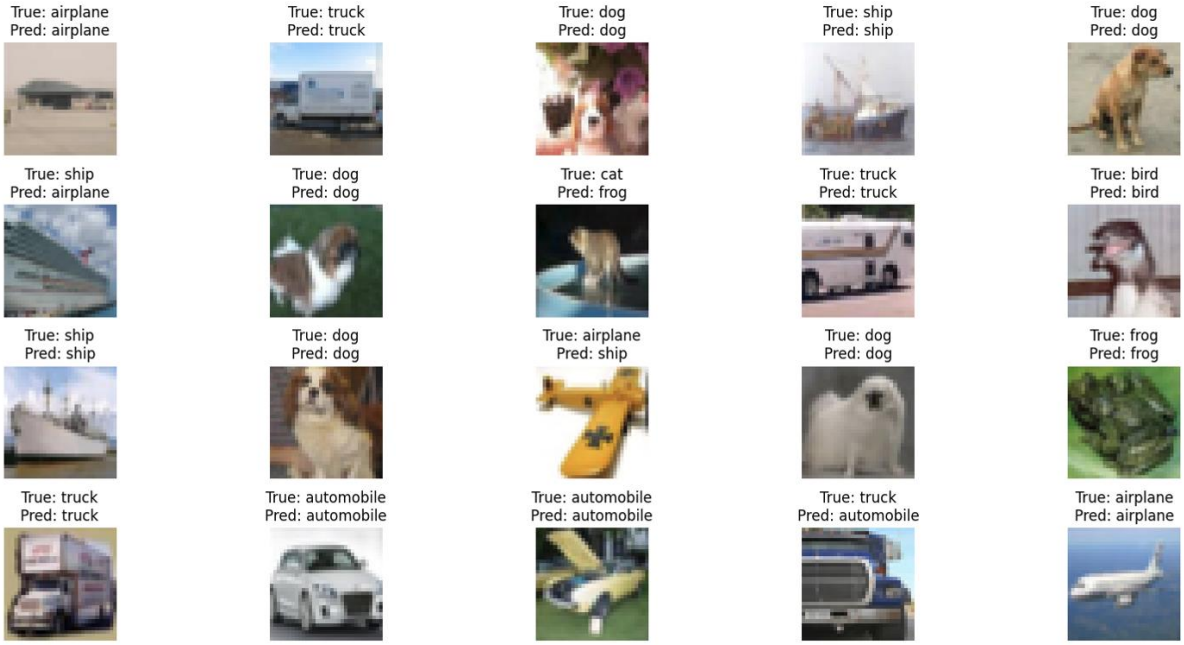

Strata - Całe trenowanie
Strata treningu
Strata walidacji
Strata
4.5
4.0
3.5
3.0
2.5
0
20
40
60
80
Epoka


True: class_98
Pred: class_98
True: class_50
Pred: class_42
True: class_38
Pred: class_79
True: class_82
Pred: class_82
True: class_40
Pred: class_40
True: class_66
Pred: class_63
True: class_7
Pred: class_6
True: class_41
Pred: class_89
True: class_79
Pred: class_79
True: class_69
Pred: class_69
True: class_95
Pred: class_95
True: class_4
Pred: class_74
True: class_75
Pred: class_7
True: class_30
Pred: class_30
True: class_23
Pred: class_71
True: class_43
Pred: class_3
True: class_44
Pred: class_44
True: class_30
Pred: class_72
True: class_90
Pred: class_89
True: class_57
Pred: class_57

b. Cifar10


0.8
0.7
0.6
0.5
0.4
Dokładność
0
10
20
30
40
Epoka
Dokładność treningu
Dokładność walidacji


Strata treningu
Strata walidacji
Strata
Epoka
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
10
20
30
40


True: airplane
Pred: airplane
True: truck
Pred: truck
True: dog
Pred: dog
True: ship
Pred: ship
True: dog
Pred: dog
True: ship
Pred: airplane
True: dog
Pred: dog
True: cat
Pred: frog
True: truck
Pred: truck
True: bird
Pred: bird
True: ship
Pred: ship
True: dog
Pred: dog
True: airplane
Pred: ship
True: dog
Pred: dog
True: frog
Pred: frog
True: truck
Pred: truck
True: automobile
Pred: automobile
True: automobile
Pred: automobile
True: truck
Pred: automobile
True: airplane
Pred: airplane

5. Wnioski

Model CNN zaimplementowany od podstaw do rozpoznawania zbioru CIFAR-100 osiągnął maksymalną dokładność na poziomie 45% - 55% podczas trwania treningu. Ewaluacja po wytrenowaniu sieci wykazała poprawność klasyfikacji wynoszącą 52,13%. Model został wsparty przez regularyzację L2 i warstwy dropout z współczynnikiem 0,5 oraz augmentację danych treningowych w postaci rotacji, przesunięć i odbić lustrzanych, co poprawiło umiejętność generalizacji modelu, ale nie wystarczyło na całkowite wyeliminowanie przeuczenia.

Użycie przetrenowanego modelu z ImageNet pozwoliło na lepsze wykorzystanie cech do niskopoziomowej generalizacji obrazków oraz uzyskanie znacznie bardziej stabilnego i efektywnego treningu. Początkowa faza uczenia na zamrożonych warstwach modelu ResNet50 pozwoliła na uzyskanie dokładności powyżej 50%. Przeprowadzenie drugiej fazy treningu z odmrożonymi górnymi warstwami pozwoliło na zwiększenie osiągniętego wyniku do około 60%, co pokazuje lepszą skuteczność tego modelu w porównaniu do modelu CNN. Sieć osiągnęła dokładność na poziomie 45,17% podczas ewaluacji. Model ResNet50 wykazał większą odporność na przeuczenie w porównaniu do prostszego modelu CNN.

Model CNN został również użyty do klasyfikacji na zbiorze CIFAR-10, w którym znajduje się 10 klas. Dla tego zbioru model osiągnął znacznie lepsze wyniki, dokładność na zbiorze testowym wyniosła 88%, co wskazuje na bardzo dobrą skuteczność tego modelu w problemie klasyfikacji dla prostszego zbioru. Model osiągnął dokładność walidacji na poziomie 84,10%. Model skutecznie rozróżniał i klasyfikował obrazki o dużych różnicach wizualnych, na przykład pojazdy i maszyny w porównaniu do zwierząt, ale popełniał błędy w klasyfikacji podobnych klas takich jak koty lub psy.

Model CNN działa skutecznie na prostszych zbiorach danych, takich jak CIFAR-10, ale ma trudności z CIFAR-100 ze względu na większą liczbę klas i złożoność wizualną obrazów. Wyniki poprawności rozróżnianych obrazków mogą sugerować, że do problemu klasyfikacji z 100 klasami mogły być zastosowane głębsze sieci neuronowe w celu osiągnięcia bardziej satysfakcjonujących wyników. Model ResNet50 dzięki swojej głębokiej architekturze i przetrenowanym wagom radzi sobie lepiej w problemie klasyfikacji CIFAR-100 niż standardowa sieć CNN. ResNet50 daje możliwość osiągnięcia jeszcze lepszych wyników poprzez zastosowanie lepszych metod augmentacji bądź lepszego doboru hiperparametrów.