

Klasyfikacja obrazów z wykorzystaniem sieci neuronowych

Zestaw danych:

<https://www.kaggle.com/datasets/mikoaikolman/pokemon-images-first-generation17000-files>

Zadania:

1. Stworzenie, przetrenowanie i ocena własnej prostej sieci CNN.
2. Wykorzystanie gotowych sieci pretrenowanych (np. ResNet, VGG) do klasyfikacji obrazów.
3. Porównanie wyników obu podejść oraz ocena skuteczności modeli.
4. Zastosowanie technik augmentacji danych i ich wpływ na jakość klasyfikacji.
5. Porównanie skuteczności różnych optymalizatorów i funkcji aktywacji.

Kroki do wykonania w projekcie:

1. Przygotowanie danych

Zbiór danych: Pobierz obrazy Pokémonów ze zbioru [Kaggle: Pokémon Images](#).

- Podziel zbiór na dane treningowe, walidacyjne i testowe (np. 70% trening, 15% walidacja, 15% test).
- Zadbaj o równowagę klas w każdej części danych.

Przetwarzanie obrazów:

- Zmiana rozmiaru: Skaluj obrazy do jednolitych wymiarów (np. 128x128 pikseli).
- Konwersja kolorów: Jeśli wymagane, skonwertuj obrazy do skali szarości lub pozostaw w trybie RGB.

Normalizacja danych:

- Przeskaluj wartości pikseli do zakresu [0, 1].
- Użyj normalizacji z parametrami średnia i odchylenie standardowe, odpowiednimi dla sieci pretrenowanych (np. ImageNet).

2. Eksploracja i wizualizacja danych

Podstawowa analiza:

- Wyświetl przykładowe obrazy z każdej klasy.
- Policz liczebność przykładów w poszczególnych klasach.

3. Stworzenie własnej prostej sieci CNN

Architektura sieci:

- Warstwy konwolucyjne (np. Conv2D) i poolingowe (np. MaxPooling2D).

- Warstwy w pełni połączone (Fully Connected).
- Funkcje aktywacji, np. ReLU w warstwach ukrytych i Softmax na wyjściu.

Trenowanie sieci:

- Wybierz odpowiedni optymalizator (np. Adam lub SGD) i funkcję strat (np. CrossEntropyLoss).
- Zdefiniuj liczbę epok i batch size.

Ocena modelu:

- Wyznacz miary takie jak dokładność (accuracy), macierz konfuzji, precision, recall i F1-score dla danych testowych.

4. Zastosowanie gotowych sieci pretrenowanych

Fine-tuning:

- Skorzystaj z bibliotek PyTorch (np. torchvision.models) do użycia pretrenowanych sieci, takich jak ResNet, VGG, EfficientNet.
- Zamroź wybrane warstwy, trenując jedynie ostatnie warstwy klasyfikatora.
- Dostosuj liczbę neuronów w warstwie wyjściowej do liczby klas.

Ocena skuteczności:

- Porównaj wyniki modeli pretrenowanych z wynikami własnej sieci CNN.

5. Augmentacja danych

Techniki augmentacji:

- Obrót (rotation),
- Przesunięcie (translation),
- Skalowanie (scaling),
- Odbicie lustrzane (horizontal flip),
- Zmiana jasności lub kontrastu.

Ocena wpływu augmentacji:

- Porównaj skuteczność modeli trenowanych na danych z augmentacją i bez augmentacji.

6. Wnioski końcowe

- Porównaj skuteczność różnych podejść i technik.
- Oceń, które przetwarzanie obrazu i sieci neuronowe dały najlepsze rezultaty (tabela i wykres podsumowujący).
- Przedstaw rekomendacje dotyczące klasyfikacji obrazów Pokémonów na podstawie wyników projektu.