

1. Wprowadzenie

Celem zadania było stworzenie i wytrenowanie klasyfikatora opartego na algorytmie Random Forest, który miałby rozpoznawać cyfry z własnych obrazów zapisanych w formacie PNG. Obrazy te były wstępnie przetwarzane, aby dopasować je do formatu podobnego do zbioru MNIST. Do implementacji wykorzystano język Python oraz bibliotekę scikit-learn.

2. Opis rozwiązania

W tym przypadku, dane wejściowe pochodziły z obrazów zapisanych w folderze PNG, a ich etykiety były przechowywane w pliku labels.txt. W pierwszym kroku przetworzono obrazy, konwertując je na obrazy w odcieniach szarości, a następnie normalizując je do rozmiaru 28x28 pikseli oraz stosując próg do binaryzacji.

Po wstępnym przetworzeniu, obrazy zostały spłaszczone do wektorów 784 elementów, co stanowiło wejście do modelu klasyfikacyjnego. Zastosowano zbiór danych MNIST jako dodatkowy zestaw danych do trenowania modelu, aby uzyskać lepsze wyniki.

Model Random Forest został wytrenowany na zbiorze danych MNIST i następnie oceniony na podstawie predykcji dotyczących własnych obrazów.

3. Wyniki

Po przeprowadzeniu eksperymentu uzyskano następujące wyniki:

- Współczynnik prawidłowej rozpoznawalności (accuracy): 0.1000 (10%)
- Precyzja (precision): 0.8111 (81.11%)
- Czułość (recall): 0.1000 (10%)

4. Wnioski

Uzyskane wyniki wskazują na niską skuteczność modelu w rozpoznawaniu własnych obrazów. Współczynnik dokładności wynoszący tylko 10% wskazuje, że model miał trudności z klasyfikowaniem własnych danych. Jednak precyzja na poziomie 81.11% oznacza, że dla pewnych klas model był w stanie rozpoznać obrazy poprawnie, ale w kontekście całego zbioru danych rezultaty nie były zadowalające.

Niska czułość również sugeruje, że model nie był w stanie poprawnie klasyfikować większości przypadków. Możliwe przyczyny to niekompatybilność cech danych własnych obrazów z danymi treningowymi (MNIST) lub potrzeba bardziej zaawansowanego przetwarzania obrazów.

Aby poprawić wyniki, warto rozważyć dostosowanie parametrów modelu, np. liczby drzew (`n_estimators`) lub głębokości drzew, a także lepsze przetwarzanie obrazów, takie jak normalizacja, zwiększenie rozmiaru zbioru danych czy augmentacja.