

# Rozpoznawanie pisma za pomocą sieci konwolucyjnych

Bartłomiej Wujec

22 czerwca 2022

## 1 Wstęp

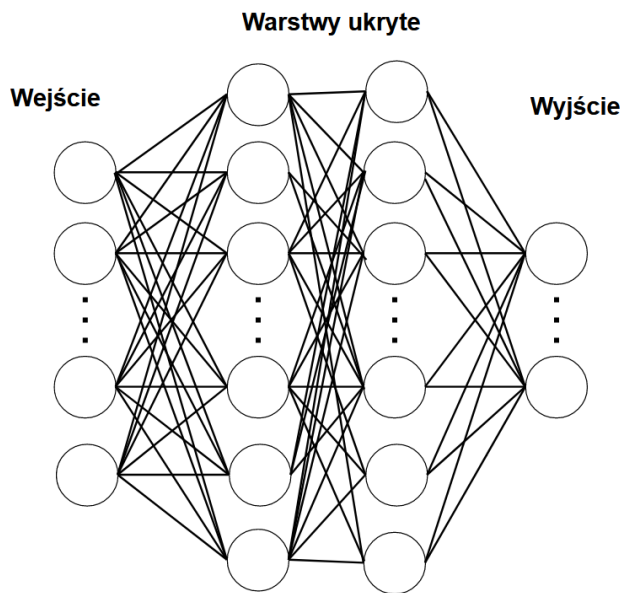
W erze digitalizacji wiele danych wciąż przechowywanych jest w formie papierowej. Sam skan takich danych zostawia nas jedynie z obrazem z którego nie można łatwo wyciągnąć danych tekstowych. Z pomocą sztucznych sieci neuronowych możemy rozpoznawać wiele obiektów w tym pismo. W mojej pracy postaram się przedstawić na ile możliwa jest to technika, jakie są jej korzyści i jakie istnieją problemy w jej implementacji.

## 2 Klasyfikatory i ich implementacje

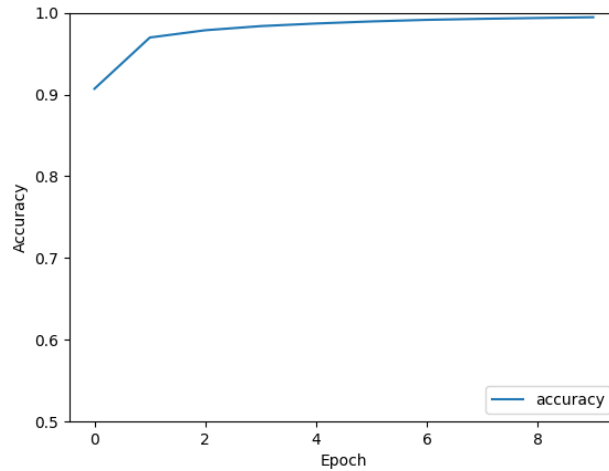
Wszystkie klasyfikatory uczyły się na tym samym zbiorze danych składającym się z 370 tys. obrazów. Każdy obraz jest rozmiaru 28x28 px i w czarno-białej przestrzeni barw. Obrazy przedstawiają duże litery angielskiego alfabetu.

### 2.1 Wielowarstwowy perceptron

Najpopularniejszy typ sztucznych sieci neuronowych. Sieć tego typu składa się zwykle z jednej warstwy wejściowej, kilku warstw ukrytych oraz jednej warstwy wyjściowej. Funkcją aktywacji jest sigmoid. Klasyfikator ma dokładność 98.7%. Nauka trwała 150s.



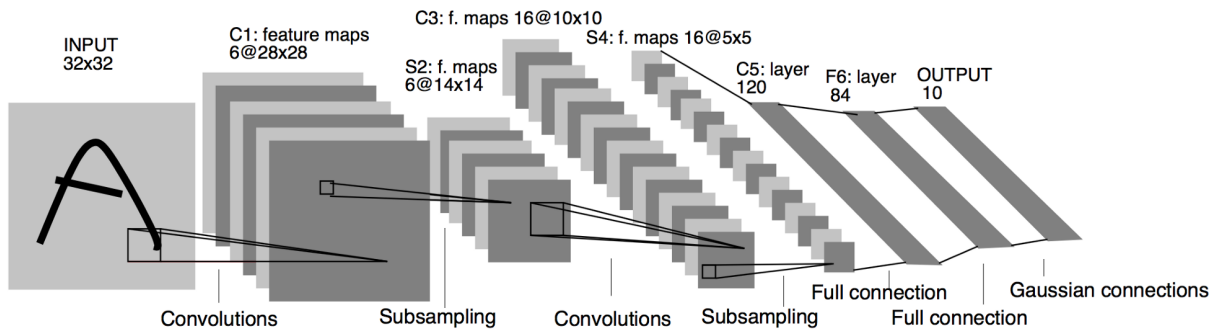
Rysunek 1: Diagram wielowarstwowego perceptronu.



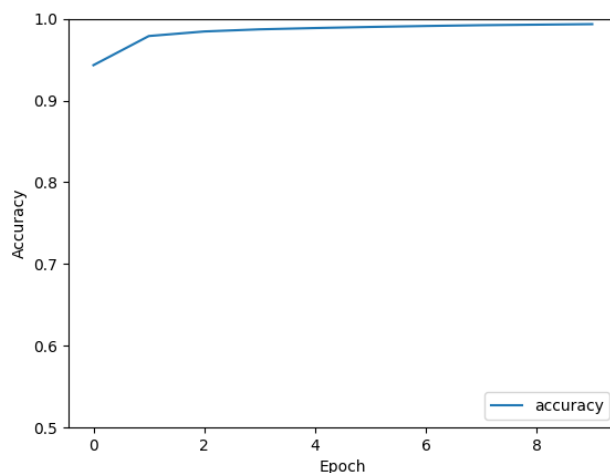
Rysunek 2: Wykres dokładności względem czasu nauki.

## 2.2 LeNET

Klasyczna i pierwsza sieć konwolucyjna z pomocą której rozpoznawano odręcznie pisane litery. Była oczywistym wyborem ze względu na podobieństwo tematów. Została minimalnie zmodyfikowana gdyż oryginalnie na wejściu przyjmowała obraz rozmiaru 32x32px. Klasyfikator ma dokładność 98.8%. Nauka trwała 410s.



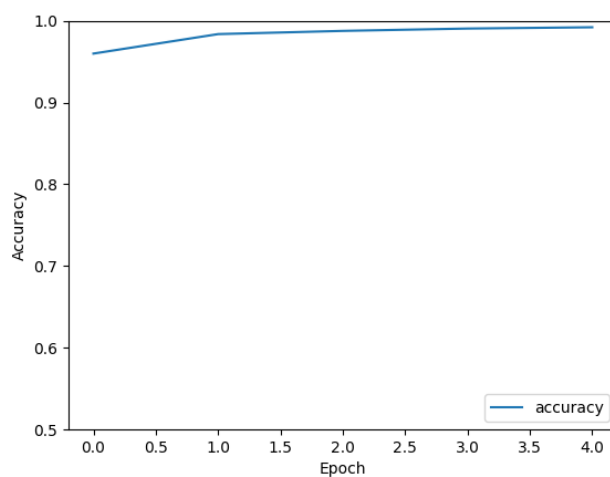
Rysunek 3: Diagram CNN LeNET.



Rysunek 4: Wykres dokładności względem czasu nauki.

## 2.3 Własna sieć konwolucyjna

Poza dwoma klasycznymi klasyfikatorami stworzyłem też własną sieć konwolucyjną. Zawiera zaledwie 2 konwolucyjne warstwy, 2 warstwy poolingowe, 1 warstwę w pełni połączoną i 1 warstwę decyzyjną. Wszystkie warstwy poza ostatnią są aktywowane przez funkcje relu. Ostatnia jest aktywowana przez softmax. Klasyfikator ma dokładność 98.8%. Nauka trwała 400s.



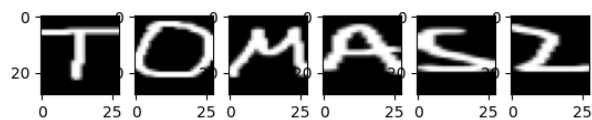
Rysunek 5: Wykres dokładności względem czasu nauki.

## 3 Preprocessing danych

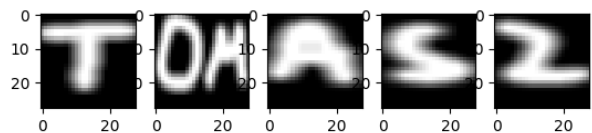
Preprocessing danych jest dużą częścią klasyfikacji. Ponieważ klasyfikatory uczą się na ładnie oddzielonych i czystych danych, dane z rzeczywistego świata mogą sprawiać problemy. Tym samym jednolity preprocessing danych jest ciężki ze względu na różnorodność rozmiarów, kolorów i różnych patternów zaszytych w obrazach. Ręczne dopasowanie preprocessingu poprawia dokładność klasyfikacji.

TOMASZ

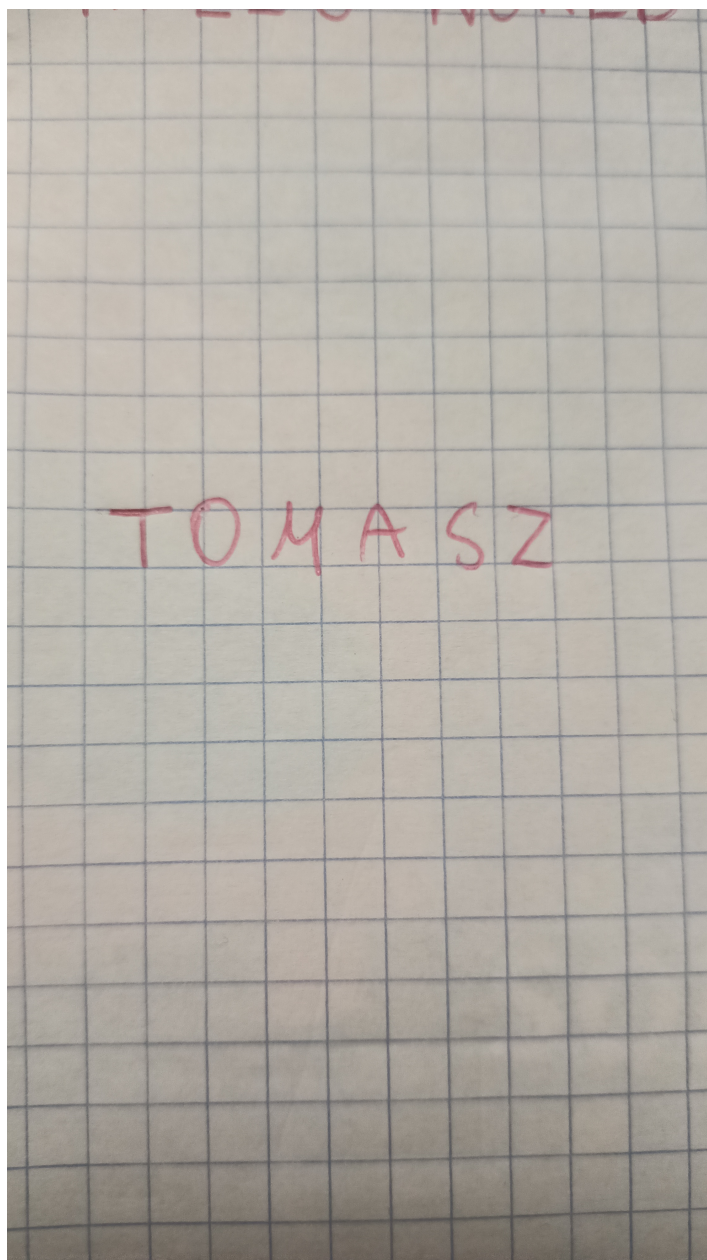
Rysunek 6: Mały obraz z tekstem przed preprocessingiem



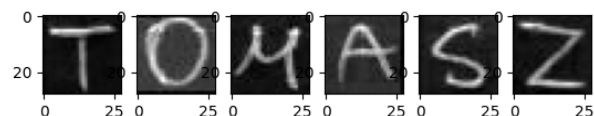
Rysunek 7: Mały obraz z tekstem po preprocessingu



Rysunek 8: Mały obraz z tekstem po złym preprocessingu



Rysunek 9: Duży obraz z kratą w tle



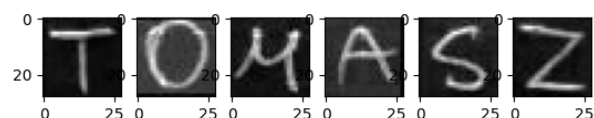
Rysunek 10: Duży obraz preprocessowany

## 4 Wyniki

Małe obrazki które praktycznie nie wymagają preprocessingu są bardzo dobrze konwertowane. Duże natomiast mają problemy. Po mimo wysokiej dokładności klasyfikatorów zupełnie nowe dane ze źródeł które zawierają szum są bliskie nierozpoznawalnym.

### 4.1 Duży obraz

Wszystkie klasyfikatory nie radzą sobie z dużym obrazem rozpoznawając maksymalnie jedną literę albo nierozpoznawając ich wcale.

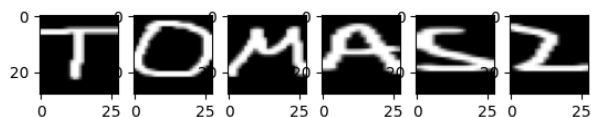


Rysunek 11: Duży obraz preprocessowany

MLP	AAAAAA
LeNET	TWYAQY
CCNN	TLYPTT

## 4.2 Mały obraz

Klasyfikatory radzą sobie dobrze, mylą niektóre specyficznie litery (np. Ó"). Na małych i bardzo czytelnych danych dokładność wydaje się być bliższa wyliczonej.



Rysunek 12: Mały obraz preprocessowany

MLP	TJMASZ
LeNET	TSMPEZ
CCNN	TJMASZ

## 5 Podsumowanie

Stworzenie modeli sieci neuronowych pozostaje najprostrzą częścią projektu. Sieci uczą się szybko i na precyzyjnych danych osiągają wysoką dokładność klasyfikacji. Preprocessing danych sprawia największe kłopoty i stanowi "bottle neck" projektu uniemożliwiając klasyfikację pisma na większych obrazach.