

Politechnika Wrocławska

Wydział Informatyki i Telekomunikacji

Algorytmy sztucznej inteligencji w Przemysle 4.0

Wtorek 13:15 – 15:00

Klasyfikacja tkanin na podstawie zdjęć

Autor:

Dominik Goleniewski, 252851

Paweł Struczewski, 252950

Prowadzący:

dr inż. Radosław Idzikowski

17 stycznia 2024



Politechnika
Wrocławska

Spis treści

1	Wybór tematu	2
1.1	Zakres projektu	2
1.1.1	Badany problem	2
1.1.2	Wybrane podejścia	2
2	Przygotowanie danych	3
2.1	Zbiór uczący	3
3	Implementacja I metody i badania - Własna sieć neuronowa	4
4	Implementacja II metody i badania - 'ResNet-50'	6
5	Porównanie obu sieci	7
6	Wnioski	8

1 Wybór tematu

1.1 Zakres projektu

1.1.1 Badany problem

Klasyfikacja obrazów: Porównanie zdolności własnej sieci neuronowej i ResNet-50 do klasyfikacji obrazów w ramach konkretnej kategorii, na przykład klasyfikacja tkanin.

1.1.2 Wybrane podejścia

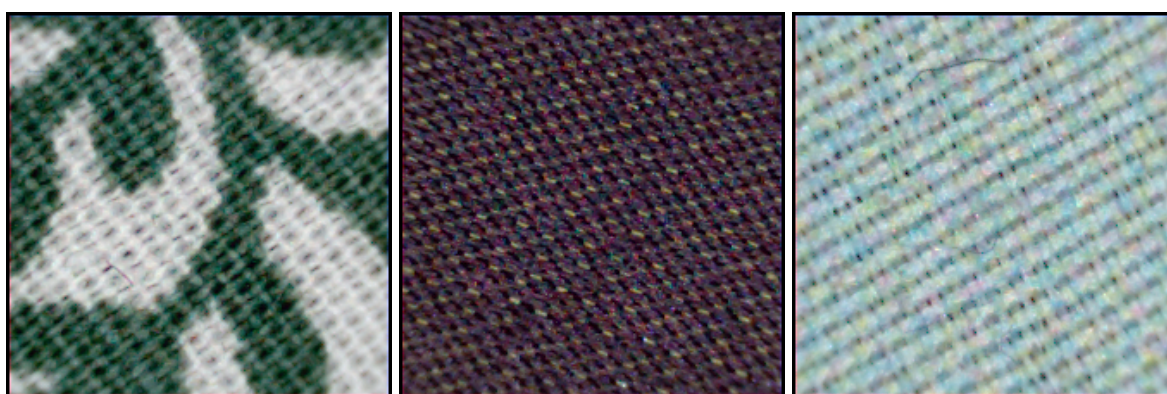
W projekcie można zastosować kilka podejść w celu porównania własnej sieci neuronowej z siecią ResNet-50:

- Porównanie wydajności: Ocena dokładności, precyzji, czułości, specyficzności i innych metryk klasyfikacyjnych na zbiorze testowym.
- Analiza czasowa: Pomiar czasu potrzebnego do treningu i predykcji obu modeli. Porównanie szybkości i zasobów obliczeniowych.

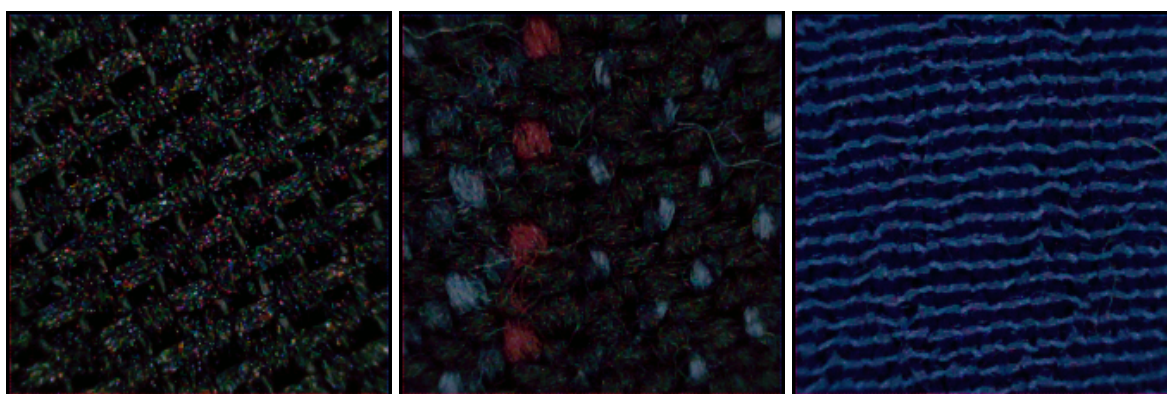
2 Przygotowanie danych

2.1 Zbiór uczący

Każdego materiału wykonano 500 zdjęć, przy każdym zdjęciu zmieniano kąt kamery. Pozwoliło to na zrobienie zdjęć struktury materiału z każdej strony. W wyniku czego zbiór liczył łącznie 3000 próbek uczących. Zbiór został podzielony również na zbiór walidacyjny oraz na zbiór testowy. Zbiór walidacyjny to około 20% całego zbioru uczącego, natomiast zbiór uczący to około 70% całego zbioru uczącego, a zbiór testowy to około 10% całego zbioru uczącego. Przykładowe próbki uczące:



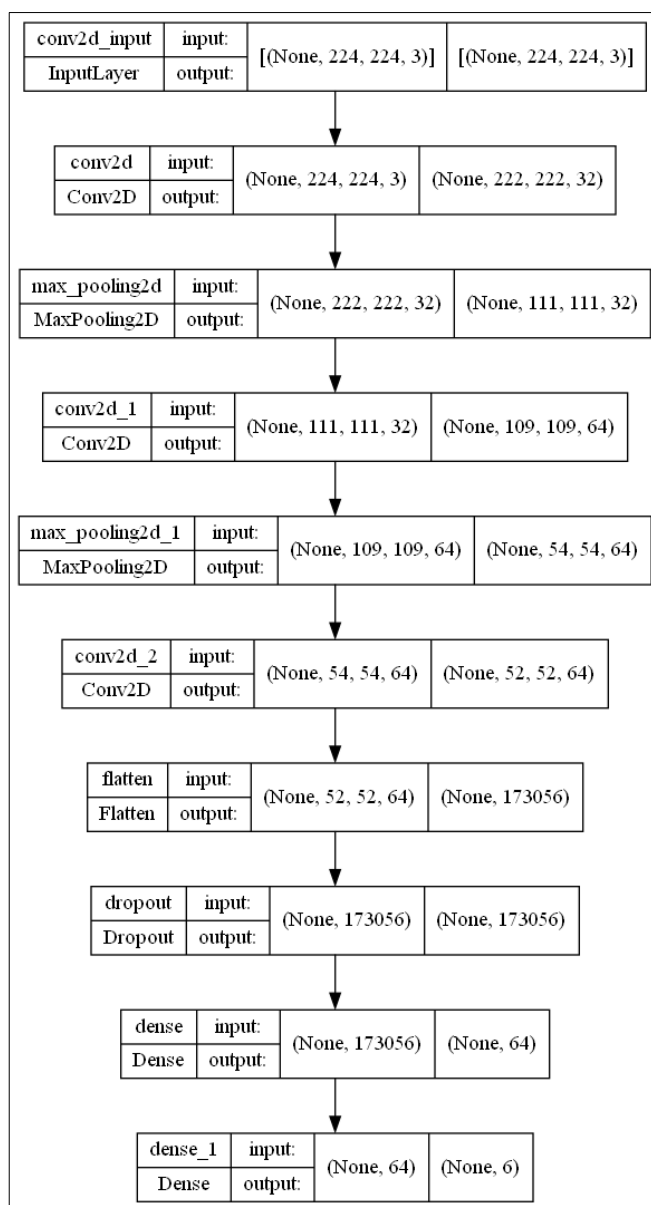
Rysunek 2.1: (a) Próbką ucząca bawełny. (b) Próbką ucząca jedwabiu.
(c) Próbką ucząca mikrofibry.



Rysunek 2.2: (a) Próbką ucząca poliestru. (b) Próbką ucząca wełny.
(c) Próbką ucząca wiskozy.

3 Implementacja I metody i badania - Własna sieć neuronowa

Sieć konwolucyjną zaimplementowano w Pythonie z wykorzystaniem biblioteki keras. W tym celu zastosowano warstwy konwolucyjne, które są rozdzielane są warstwami maxpoolingu. Pierwszą warstwą jest warstwa Conv2D, która jest warstwą splotową. Warstwa ta zawiera wyuczone filtry, które wyodrębniają cechy odróżniające od siebie różne obrazy. Drugą warstwą jest warstwa MaxPooling2D, która wykonuje operacje max-pooling, w celu zmniejszenia rozmiaru obrazu. Schemat własnej sieci neuronowej:



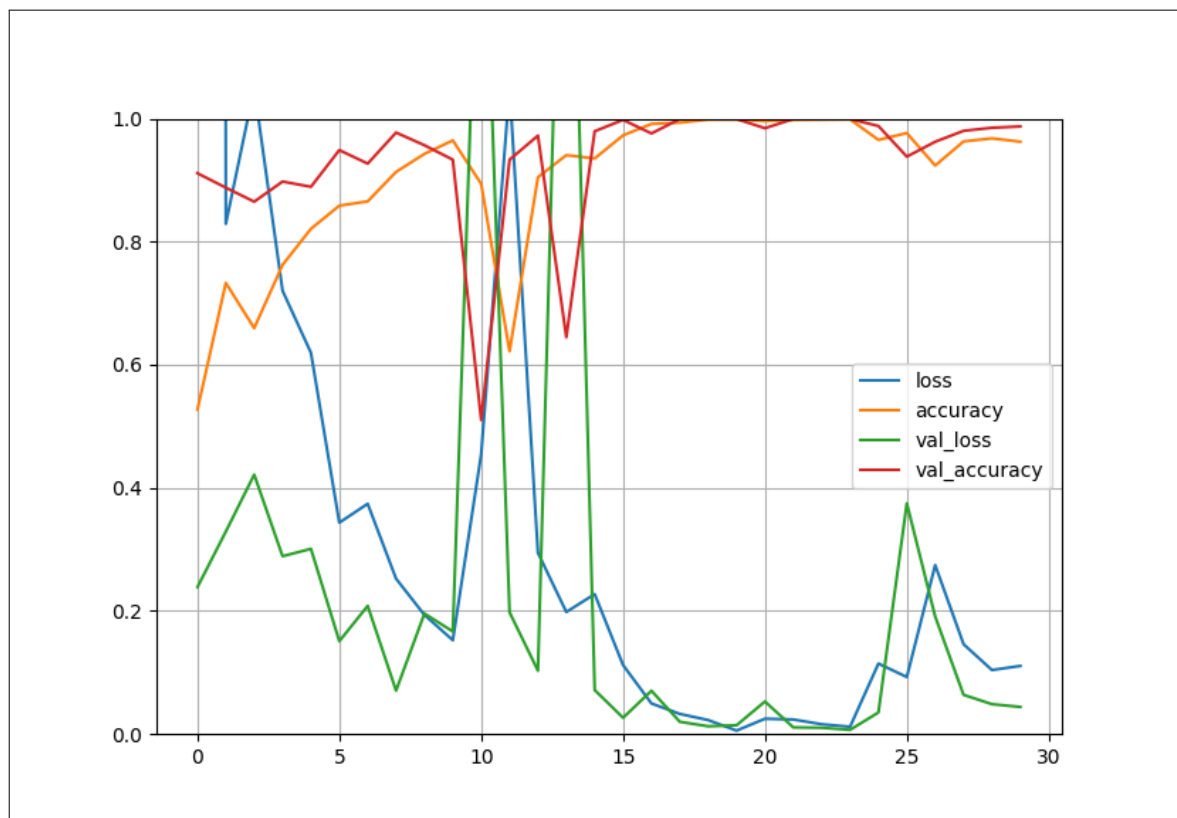
Rysunek 3.1: Schemat modelu własnej sieci neuronowej.

Podczas trenowania modelu wykorzystano kilka optymalizatorów. Dla optymalizatora "rmsprop" występowało

zjawisko, w którym dokładność modelu zatrzymywała się w jednej wartości i nie chciała się zmienić. Ostatecznie wykorzystanym optymalizatorem w czasie trenowania modelu został "adam". Natomiast wykorzystana funkcja straty to "categorical_crossentropy"

Model był trenowany na komputerze z kartą graficzną NVIDIA GeForce 1050 Ti. Czas trwania działania jednej epoki trwał około 80 sekund. Ilość wszystkich epok podczas trenowania wynosiła 30, w tym wypadku całkowity czas trenowania modelu wynosił około 40 minut.

Przebieg uczenia można zaobserwować na wykresie poniżej:



Rysunek 3.2: Przebieg nauczania modelu własnej sieci

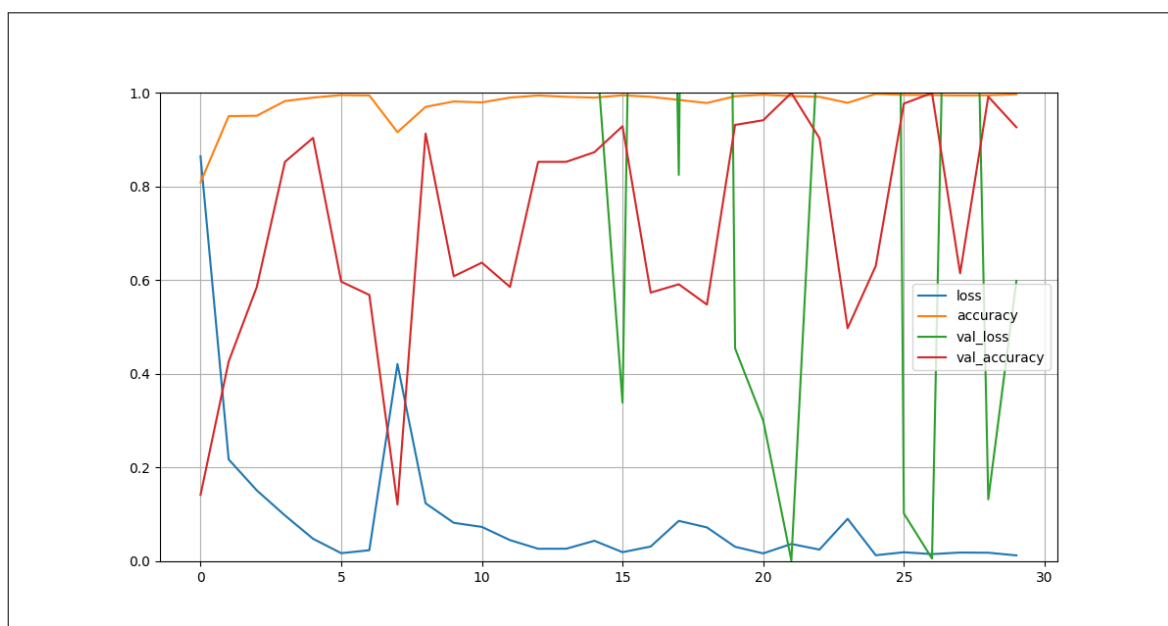
Na osi odciętych wykresu 5.3 znajdują się epoki. Na wykresie można zaobserwować zależność między funkcją straty a dokładnością modelu. Im wartość funkcji straty była mniejsza tym dokładność modelu była większa. Analogicznie im dokładność modelu była mniejsza tym wartość funkcji straty była większa. Ostatecznie dokładność modelu wynosi około 0.96.

4 Implementacja II metody i badania - 'ResNet-50'

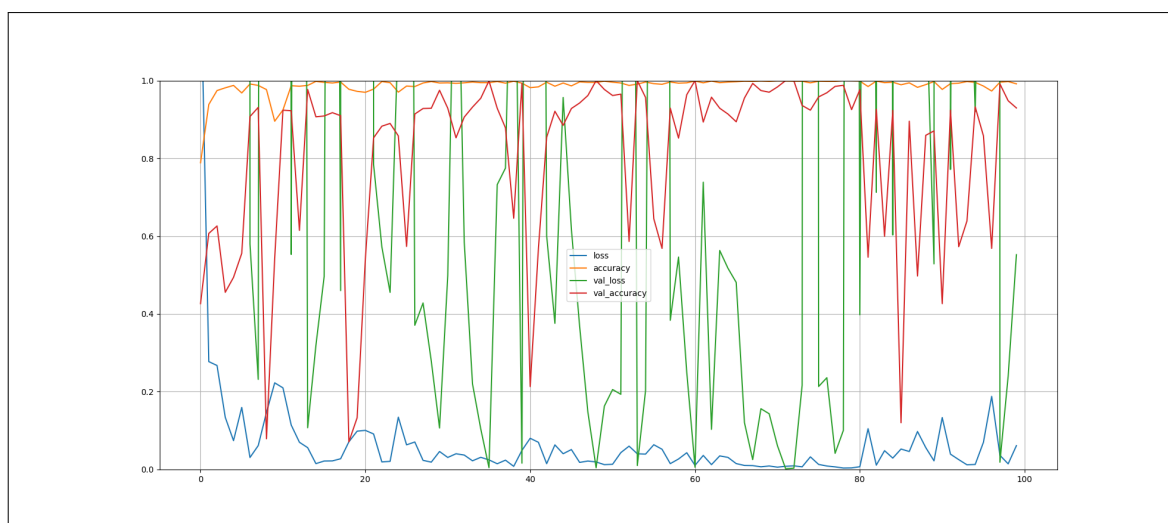
ResNet-50, czyli Residual Network z 50 warstwami, to jedna z zaawansowanych architektur sieci neuronowej, która została zaprojektowana w celu skutecznego rozwiązywania problemów związanych z klasyfikacją obrazów.

Sieć została zaimportowana z biblioteki Tensorflow, a następnie wykonano na niej uczenie na własnym zestawie danych. Model ten uczył się 16 godzin dla 100 epok.

Przebieg uczenia można zaobserwować na wykresie poniżej:



Rysunek 4.1: Przebieg nauczania modelu sieci ResNet-50 dla 30 epok



Rysunek 4.2: Przebieg nauczania modelu sieci ResNet-50 dla 100 epok

5 Porównanie obu sieci

Porównanie czasu uczenia obu sieci neuronowych.

	Sieć własna	Resnet50
30 epok	42 minut	4 godziny 47 minut
100 epok	1 godzina 43 minuty	16 godzin 23 minuty

Tabela porównawcza dokładności, precyzji, czułości i F1 dla różnych konfiguracji treningowych: Sieć własna vs. Resnet50

	Sieć własna		Resnet50	
	30 epok	100 epok	30 epok	100 epok
Dokładność	0,9011	0,8514	0,9268	0,9296
Precyzja	0,83	0,79	0,87	0,96
Czułość	0,91	0,86	0,93	0,93
F1	0,87	0,82	0,89	0,91

6 Wnioski

- Własna sieć neuronowa nauczyła się znacznie szybciej niż gotowa sieć ResNet-50 zaimportowana z biblioteki Tensorflow,
- Gotowa sieć ResNet-50 jest mniej wrażliwa na niespodziewane zakłócenia w danych,
- Dla zbioru danych zawierających unikalne dane związane z klasyfikacją materiałów, własna sieć neuronowa może być dostosowana do tych specyficznych wymagań. ResNet-50, będąc bardziej ogólnym modelem, może nie uwzględniać specyficznych cech charakterystycznych dla materiałów w danym zbiorze danych.
- W przypadku sieci własnej, dokładność spadła z 0,9011 dla 30 epok do 0,8514 dla 100 epok. Takie obniżenie dokładności może sugerować, że model zaczyna przeuczać się na danych treningowych.
- W przypadku ResNet-50, dokładność utrzymała się na wysokim poziomie, osiągając 0,9268 dla 30 epok i lekko wzrastając do 0,9296 dla 100 epok. Można wnioskować, że ResNet-50 jest bardziej stabilny i mniej podatny na przeuczenie niż sieć własna.
- Precyzja sieci własnej zmniejszyła się z 0,83 dla 30 epok do 0,79 dla 100 epok, co potwierdza podejrzenie o przeuczeniu.
- W przypadku ResNet-50, precyzja wzrosła z 0,87 dla 30 epok do 0,96 dla 100 epok, co wskazuje na zdolność modelu do lepszego radzenia sobie z klasyfikacją.
- ResNet-50 utrzymał wysoką czułość, oscylując wokół 0,93 bez większych zmian między 30 a 100 epoką.