

| | |
|---|--|
| Podstawy sieci neuronowych | |
| Kierunek <i>Informatyczne systemy automatyki</i> | Termin <i>Poniedziałek 18:55 TP</i> |
| Imię, nazwisko, numer albumu <i>Jakub Janiczuk 280141, Oskar Mulcan 280122</i> | Data <i>19 stycznia 2026</i> |



PODSTAWY SIECI NEURONOWYCH – DOKUMENTACJA PROJEKTOWA

Spis treści

| | | |
|----------|--|----------|
| 1 | Analiza problemu | 2 |
| 2 | Architektura programu | 2 |
| 3 | Selekcja i format danych | 2 |
| 3.1 | Wybrane dane | 2 |
| 3.2 | Odrzucone danych | 3 |
| 4 | Architektura sieci | 3 |
| 4.1 | Warstwy sieci: | 4 |
| 5 | Trenowanie sieci | 4 |
| 6 | Wyniki | 4 |
| 6.1 | Dokładność i straty | 4 |
| 6.2 | Macierz trafień | 5 |
| 6.3 | Raport klasyfikacji | 6 |
| 6.4 | Krzywa ROC (Receiver Operating Characteristic) i krzywa Precision-Recall | 6 |
| 7 | Wnioski | 7 |

1 Analiza problemu

W ramach wykonania projektu wybraliśmy problem klasyfikacji emocji na podstawie zdjęcia. W celu wytrenowania sieci wybraliśmy 4 emocje: neutralna, szczęśliwa, obrzydzona i zdziwiona z zestawu danych pod adresem <https://www.kaggle.com/datasets/fahadullah/facial-emotion-recognition-dataset>.

2 Architektura programu

W celu stworzenia sieci neuronowej użyliśmy języka Python 3 w środowisku Google Collab, wraz z bibliotekami: Do zrealizowania projektu wybraliśmy następujące środowisko:

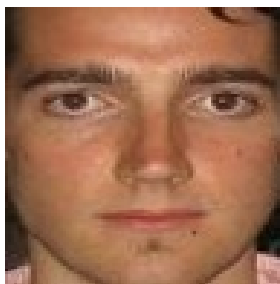
- środowisko wykonawcze Google Colab
- język programowania Python3
- biblioteka TensorFlow i Keras do implementacji sieci neuronowej
- biblioteka matplotlib do tworzenia wykresów
- biblioteka sklearn do analizy wyników

3 Selekcja i format danych

Spośród dostępnych klas emocji wybraliśmy cztery: neutral, happy, disgust oraz angry. Z zestawów danych ręcznie wybraliśmy po 1000 reprezentacji każdej z nich. Wszystkie obrazy są w formacie jpg (RGB) i mają rozmiar $96\text{px} \times 96\text{px}$. Aby zachować spójność danych każde z nich jest w kolorze (pomimo, iż oryginalne zestawy zawierały również zdjęcia w skali szarości). Wybierając zdjęcia do naszych zestawów szukaliśmy takich, które nie były zbyt zniekształcone, jednocześnie kierując się ich różnorodnością, aby uniknąć zbyt wąskiej klasyfikacji. Zdjęcia zostały zgrupowane w pakiety po 32 przy pomocy parametru batch size na etapie preprocessingu danych. Dane zostały podzielone na zestawy treningowy, walidacyjny i testowy w proporcjach 8:1:1.

Przykłady wybranych oraz odrzuconych zdjęć:

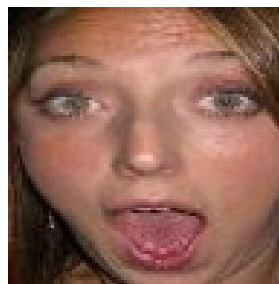
3.1 Wybrane dane



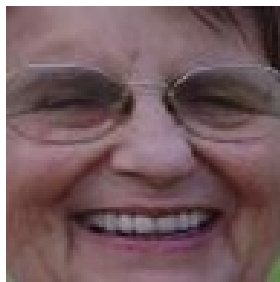
(a) Zestaw: Neutralny



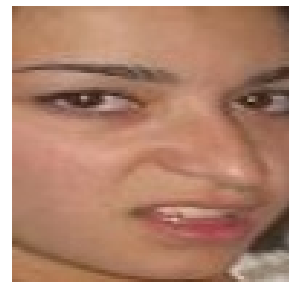
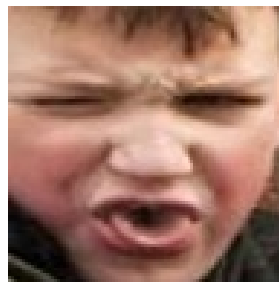
(b) Zestaw: Surprise



(c) Zestaw: Happy



(d) Zestaw: Disgust



Rysunek 1: Zestawienie wybranych obrazów dla wszystkich klas emocji.

3.2 Odrzucone danych



Rysunek 2: Zestawienie odrzuconych obrazów dla wszystkich klas emocji.

4 Architektura sieci

Poniżej przedstawiono wykorzystaną przez nas architekturę sekwencyjnej sieci CNN:

```
1 model = models.Sequential([
2
3     layers.Rescaling(1./255),
4
5     layers.Conv2D(32, (3,3), activation='relu', padding="same"),
6     layers.Conv2D(32, (3,3), activation='relu', padding="same"),
7     layers.MaxPooling2D(),
8     layers.BatchNormalization(),
9
10    layers.Conv2D(64, (3,3), activation='relu', padding="same"),
11    layers.Conv2D(64, (3,3), activation='relu', padding="same"),
12    layers.MaxPooling2D(),
13    layers.BatchNormalization(),
14
15    layers.Conv2D(128, (3,3), activation='relu', padding="same"),
16    layers.Conv2D(128, (3,3), activation='relu', padding="same"),
17    layers.MaxPooling2D(),
18    layers.BatchNormalization(),
19
20    layers.Conv2D(256, (3,3), activation='relu', padding="same"),
21    layers.Conv2D(256, (3,3), activation='relu', padding="same"),
22    layers.MaxPooling2D(),
23    layers.BatchNormalization(),
24
25    layers.Flatten(),
26    layers.Dense(512, activation="relu"),
27    layers.Dropout(0.5),
28    layers.Dense(4, activation="softmax")
29 ])
```

4.1 Warstwy sieci:

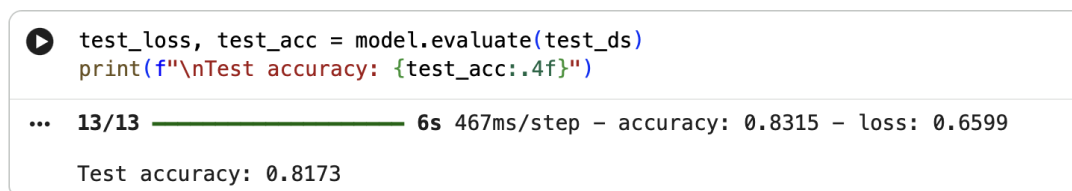
- **Warstwa Rescaling:** Jest to warstwa wstępnego przetwarzania. Przekształca ona wartości pikseli z zakresu $[0, 255]$ na zakres $[0, 1]$. Normalizacja danych wejściowych jest kluczowa dla stabilności procesu uczenia i szybszej zbieżności gradientu.
- **Warstwy Conv2D:** Warstwy te wykorzystują filtry do wykrywania cech obrazu. Początkowe warstwy (32 filtry) wykrywają proste krawędzie i kształty. Głębsze warstwy (do 256 filtrów) potrafią rozpoznawać złożone wzorce, takie jak układ ust czy zmarszczki wokół oczu. Zastosowanie funkcji ReLU wprowadza nieliniowość, pozwalając sieci uczyć się skomplikowanych zależności.
- **Warstwy MaxPooling2D:** Zadaniem tej warstwy jest redukcja wymiarów przestrzennych obrazu (szerokości i wysokości) przy zachowaniu najważniejszych informacji. Pozwala to na zmniejszenie liczby parametrów obliczeniowych.
- **Warstwy BatchNormalization:** Warstwa ta stabilizuje proces uczenia poprzez normalizację aktywacji wewnątrz sieci. Dzięki niej model jest mniej wrażliwy na początkową inicjalizację wag, co pozwala na zastosowanie wyższego tempa uczenia i drastycznie redukuje czas potrzebny na dotrenowanie modelu.
- **Warstwa Flatten:** Pełni rolę łącznika między częścią spłotową (wyciągającą cechy) a częścią gęstą (klasyfikującą). Przekształca ona wielowymiarową mapę cech w jeden wektor danych, który może zostać podany na wejście warstwy Dense.
- **Warstwa Dense:** Warstwa o 512 neuronach dokonuje ostatecznej interpretacji cech wykrytych przez sploty.
- **Warstwa Dropout:** Służy do regularyzacji. Podczas każdej epoki treningu losowo "wyłącza" 50% neuronów. Zapobiega to zjawisku overfittingu, zmuszając sieć do nauki bardziej ogólnych wzorców zamiast zapamiętywania konkretnych zdjęć treningowych.
- **Warstwa Wyjściowa:** Ostatnia warstwa o czterech neuronach odpowiadającym czterem klasom. Funkcja aktywacji Softmax zamienia wyniki sieci na prawdopodobieństwo. Suma wyników z tych 4 neuronów zawsze wynosi 100%, co pozwala określić, z jaką pewnością model przypisuje twarz do danej emocji.

5 Trenowanie sieci

W procesie uczenia wykorzystaliśmy algorytm optymalizacji Adam o współczynniku uczenia 0.0001, funkcję strat entropii krzyżowej (*cross entropy loss*) i metrykę *accuracy*. Uczenie odbyło się na przestrzeni 25 epok. Ponadto, do procesu dodaliśmy funkcję *callback*. Wybraliśmy funkcję ReduceLROnPlateau, która w przypadku braku poprawy dokładności przez 5 epok z rzędu zmniejsza *learning rate* dziesięciokrotnie, oraz EarlyStopping, która przy 10 epokach z rzędu bez poprawy całkowicie przerywa trening.

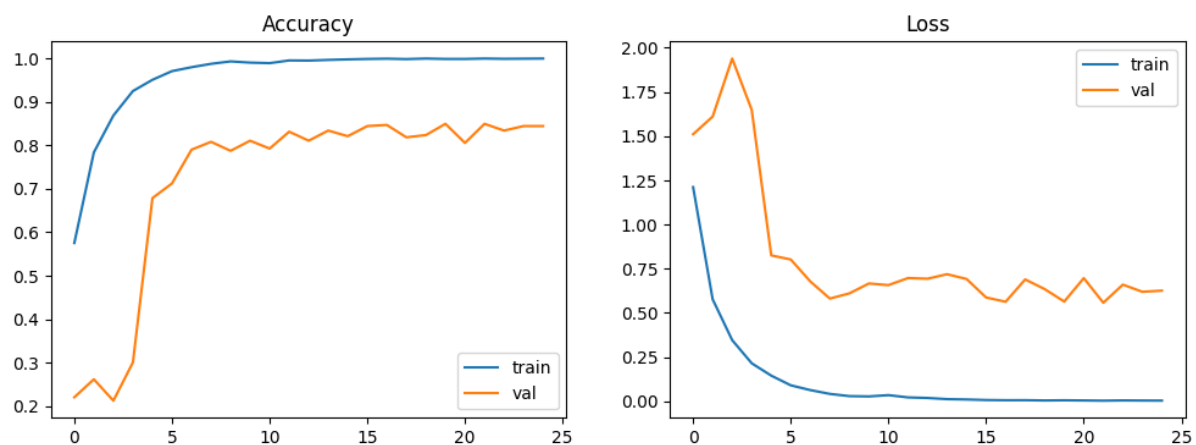
6 Wyniki

6.1 Dokładność i straty



Rysunek 3: Wyniki testów sieci neuronowej pod kątem dokładności

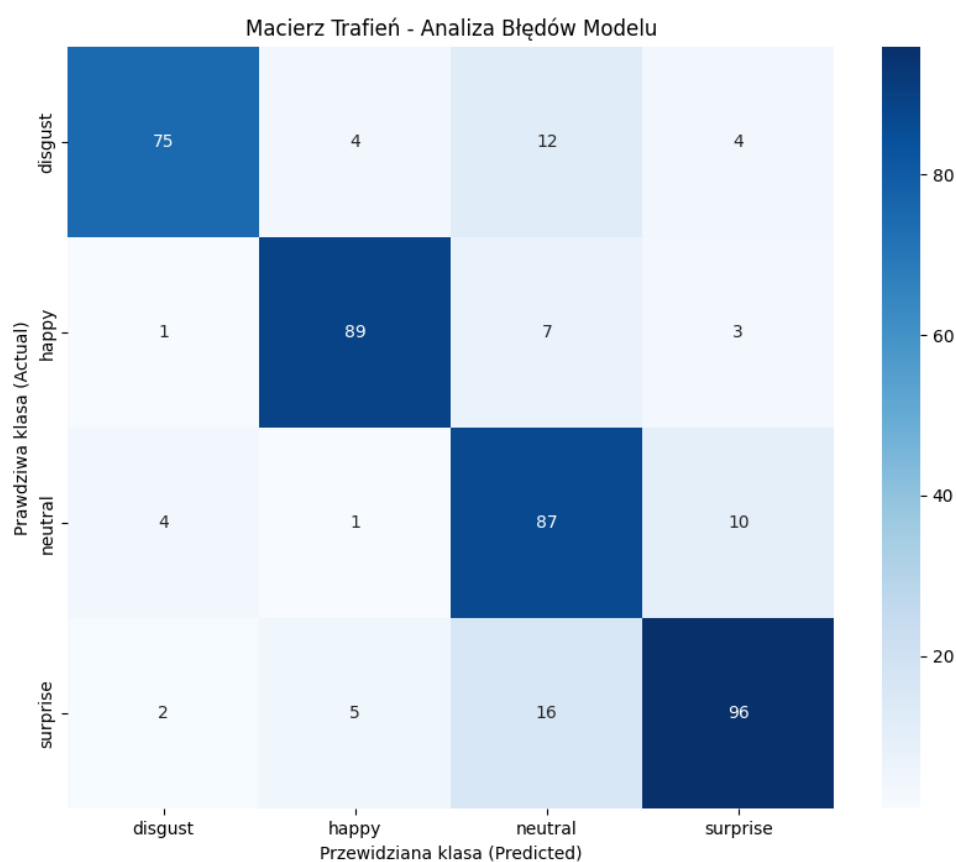
Po trenowaniu sieci osiągnęliśmy dokładność 81.73% po 25 epokach. Poniżej zamieszczono wykresy przedstawiające zmiany dokładności i strat na przestrzeni kolejnych epok:



Rysunek 4: Celność sieci w zależności od iteracji uczenia

6.2 Macierz trafień

Poniższa macierz przedstawia rozkład prawidłowych oraz błędnych predykcji modelu dla każdej z klas. Główna przekątna macierzy reprezentuje poprawnie sklasyfikowane przypadki (True Positives), pozostałe pola wskazują natomiast liczbę błędnych trafień. Największym zidentyfikowanym problemem są pomyłki między klasami neutral i surprise, oraz omyłkowe wybieranie neutral zamiast disgust. Może to sugerować, że model w dalszym ciągu ma problemy z rozpoznawaniem bardziej subtelnych zmian w mimice, jak np. lekkie uniesienie brwi.



Rysunek 5: Macierz trafień

6.3 Raport klasyfikacji

Poniższy raport klasyfikacji przedstawia zbiór metryk pomocnych w ocenie jakości klasyfikacji każdej z emocji.

1. **Precision:** prawdopodobieństwo poprawnego wskazania klasy przez model
2. **Recall:** procentowe wskazanie wszystkich wystąpień danej klasy
3. **F-1-Score:** średnia harmoniczna z metryk precision i recall
4. **Support:** liczba faktycznych wystąpień danej klasy

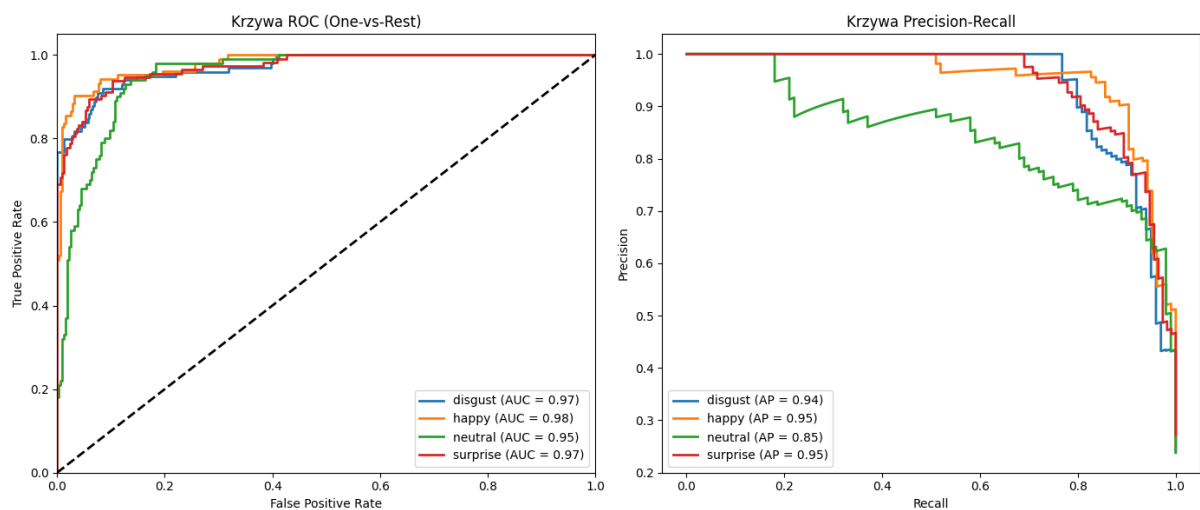
| Szczegółowy Raport Klasyfikacji: | | | | |
|----------------------------------|-----------|--------|----------|---------|
| | precision | recall | f1-score | support |
| disgust | 0.91 | 0.79 | 0.85 | 95 |
| happy | 0.90 | 0.89 | 0.89 | 100 |
| neutral | 0.71 | 0.85 | 0.78 | 102 |
| surprise | 0.85 | 0.81 | 0.83 | 119 |
| accuracy | | | 0.83 | 416 |
| macro avg | 0.84 | 0.83 | 0.84 | 416 |
| weighted avg | 0.84 | 0.83 | 0.84 | 416 |

Rysunek 6: Raport klasyfikacji

6.4 Krzywa ROC (Receiver Operating Characteristic) i krzywa Precision-Recall

Krzywa ROC ilustruje zdolność modelu do rozróżniania klas poprzez przedstawienie relacji między czułością (True Positive Rate), a odsetkiem fałszywych alarmów (False Positive Rate). Kluczowym wskaźnikiem jest AUC (Area Under Curve). Wartość AUC bliska 1.0 świadczy o idealnej separacji klas, natomiast 0.5 odpowiada losowemu zgadywaniu.

Krzywa Precision-Recall przedstawia z kolei kompromis między precyzją (wiarygodnością wskazania danej emocji), a czułością (zdolnością do wyłapania wszystkich przypadków danej emocji).



Rysunek 7: Wykres ROC

7 Wnioski

- Model osiągnął satysfakcjonującą dokładność na poziomie 81.73%.
- Odpowiedni dobór danych do zestawów uczących, walidacyjnych i testowych przyczynił się do znacznej poprawy zdolności modelu do trafnej klasyfikacji emocji.
- Analiza krzywych uczenia wykazuje, że sieć szybko adaptuje się do danych treningowych, osiągając niemal 100% dokładności. Istnieje jednak zauważalna rozbieżność między zbiorem treningowym a walidacyjnym (overfitting), którą częściowo ograniczono dzięki warstwie Dropout (0.5) oraz technikom regularyzacji.
- Zastosowanie funkcji ReduceLROnPlateau pozwoliło na precyzyjne dotrenowanie wag w momentach stagnacji, co widać na wykresie straty (Loss), gdzie po początkowym gwałtownym spadku następuje stabilizacja i wygładzenie krzywej.
- W procesie trenowania analizowanego tu modelu nie został wykorzystany mechanizm EarlyStopping. Oznacza to, że fluktuacje straty walidacyjnej były na tyle małe lub występowały na tyle rzadko, że licznik cierpliwości algorytmu (10 epok bez poprawy) nie został w pełni wyczerpany.
- Najwyższą skuteczność odnotowano dla emocji disgust i happy, co potwierdzają wysokie wartości wskaźników AUC (0.97 dla disgust i 0.98 dla happy) oraz F-1-Score (0.85 dla disgust i 0.89 dla happy). Sugeruje to, że te emocje posiadają najbardziej charakterystyczne cechy wizualne, które sieć spłotowa potrafi łatwo wyodrębnić.
- Kluczowym problemem zidentyfikowanym w macierzy trafień jest mylenie emocji surprise z neutral (16 przypadków) oraz neutral z surprise (10 przypadków). Może to wynikać z faktu, że subtelne zmiany mimiczne, takie jak lekkie uniesienie brwi, są interpretowane przez model jako stan neutralny, jeśli nie towarzyszą im inne wyraźne cechy.
- Błędy klasyfikacji między disgust a neutral (12 przypadków) sugerują, że negatywne emocje o mniejszej intensywności są dla modelu trudne do odróżnienia od braku ekspresji.
- Na wykresie dokładności (Rysunek 4) widać, że o ile dokładność na zbiorze treningowym (train) szybko osiąga niemal 100%, o tyle dokładność na zbiorze walidacyjnym (val) zatrzymuje się w okolicy 80-85%. To sugeruje, że doszło do przeuczenia – model nauczył się rozpoznawać konkretne cechy zdjęć treningowych, zamiast szukać ogólnych reguł pozwalających na bezbłędną klasyfikację nowych obrazów. Na wykresie Loss można zaobserwować, że po początkowym spadku, strata dla zbioru walidacyjnego wykazuje tendencję do oscylacji i lekkiego wzrostu w końcowych epokach, podczas gdy strata treningowa dąży do zera. Świadczy to o tym, że sieć traci zdolność generalizacji.