Architektura i trenowanie własnego modelu

Projektowany model to klasyczna sieć konwolucyjna (CNN) przeznaczona do rozpoznawania elementów na obrazach, składająca się z dwóch warstw konwolucyjnych, funkcji aktywacji ReLU, maxpoolingu oraz w pełni połączonych warstw.

1. Warstwy Konwolucyjne – Convolution Layers:
    - **Warstwa 1:** Conv2d(3, 32, kernel_size=3, stride=1, padding=1) – Przyjmuje obrazy RGB o wymiarach 128x128 i tworzy 32 mapy cech (128x128). Kernel 3x3 służy do konwolucji.
    - **Aktywacja:** ReLU – Wprowadza nieliniowość, wspomagając model w uczeniu bardziej złożonych zależności.
    - **Pooling:** MaxPool2d(kernel_size=2, stride=2) – Zmniejsza wymiary przestrzenne mapy cech do 64x64, co zmniejsza liczbę parametrów i zapobiega nadmiernemu dopasowaniu.
    - **Warstwa 2:** Conv2d(32, 64, kernel_size=3, stride=1, padding=1) – Tworzy 64 mapy cech (64x64).
    - Aktywacja: ReLU
    - **Pooling:** MaxPool2d(kernel_size=2, stride=2) – Zmniejsza wymiary do 32x32.
2. Warstwy w pełni połączone – Fully Connected Layers:
    - **Warstwa 1:** Linear(64 * 32 * 32, 128) – Spłaszcza dane wejściowe do wektora 128-wymiarowego.
    - Aktywacja: ReLU
    - **Warstwa 2:** Linear(128, 1) – Ostateczna warstwa klasyfikująca (klasyfikacja binarna).

Wybór tej architektury oparty jest na efektywności sieci konwolucyjnych w zadaniach rozpoznawania elementów w obrazach. Dwie warstwy konwolucyjne umożliwiają stopniową redukcję wymiarów obrazu oraz uchwycenie różnych poziomów cech.

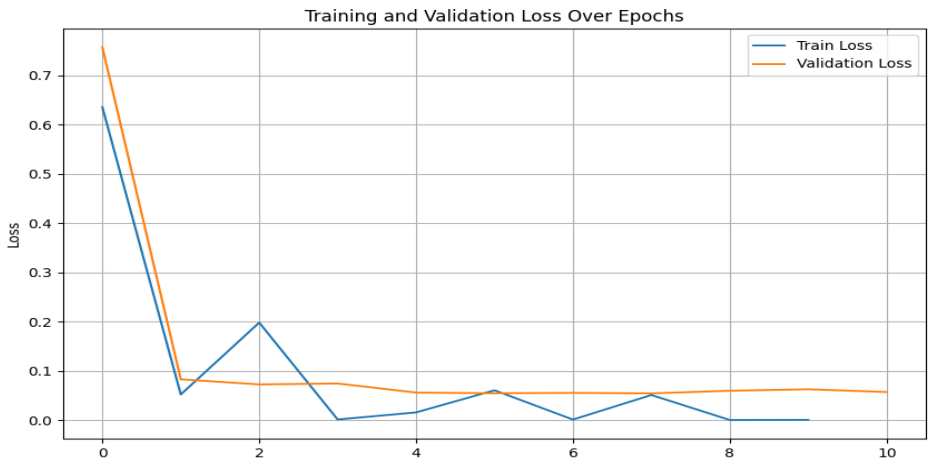
```
  | Name      | Type              | Params | Mode
-------------------------------------------------------
0 | model     | FaceID_CNN        | 8.4 M  | train
1 | train_acc | BinaryAccuracy    | 0      | train
2 | val_acc   | BinaryAccuracy    | 0      | train
3 | criterion | BCEWithLogitsLoss | 0      | train
-------------------------------------------------------
8.4 M     Trainable params
0         Non-trainable params
8.4 M     Total params
33.633    Total estimated model params size (MB)
15        Modules in train mode
0         Modules in eval mode
```

Rysunek 1 Zrzut ekranu podczas treningu przy wykorzystaniu biblioteki PyTorch Lightning.

Model został zaimplementowany w klasie **FaceIDModel**, dziedziczącej po PyTorch Lightning, co upraszcza proces trenowania i zarządzania cyklem życia modelu (np. wprowadzenie early stopping w celu zapobiegania przeuczeniu). Trenowanie odbywa się za pomocą optymalizatora **Adam**, wydajnego w przypadku dużych zbiorów danych, takich jak CelebA, oraz stosującego adaptacyjną strategię uczenia.

Funkcją celu jest **Binary Cross Entropy Loss**, użyta w wersji **nn.BCEWithLogitsLoss** dla lepszej efektywności obliczeniowej. Zastosowano także tensor **pos_weight**, który pomaga radzić sobie z niezbalansowanym zbiorem danych.

Aby poprawić generalizację modelu, zastosowano augmentację danych w postaci **szumu Gaussa**, **zmiany koloru** oraz **rotacji zdjęć**, co pozwala na lepsze dostosowanie modelu do nieznanych danych, np. obrazów z kamery.



Rysunek 2 Wartosc funkcji celu per epoka.

Architektura i trenowanie wykorzystanego modelu z torchvision

Do zadania klasyfikacji obrazu, polegającego na rozpoznawaniu atrybutu „Smiling”, wykorzystano model **ResNet-18** z uprzednio wytrenowanymi wagami na zbiorze **ImageNet1K_V1**. Jest to sieć o 18 warstwach, która wykorzystuje **residual connections**, umożliwiające radzenie sobie z problemem zanikającego gradientu w głębokich sieciach.

Struktura Modelu ResNet-18

Model składa się z 4 bloków konwolucyjnych, gdzie każdy blok odpowiada za ekstrakcję cech z obrazu:

- **Blok 1:** 64 filtry
- **Blok 2:** 128 filtrów
- **Blok 3:** 256 filtrów
- **Blok 4:** 512 filtrów

```
  | Name      | Type               | Params | Mode
-------------------------------------------------------
0 | model     | Ready_faceID_CNN   | 11.2 M | train
1 | train_acc | BinaryAccuracy     | 0      | train
2 | val_acc   | BinaryAccuracy     | 0      | train
3 | criterion | BCEWithLogitsLoss  | 0      | train
-------------------------------------------------------
11.2 M    Trainable params
0         Non-trainable params
11.2 M    Total params
44.969    Total estimated model params size (MB)
75        Modules in train mode
0         Modules in eval mode
```

Rysunek 3 Zrzut ekranu podczas treningu przy wykorzystaniu biblioteki PyTorch Lightning.

Przygotowanie Danych

Obrazy wejściowe musiały zostać przeskalowane do rozdzielczości *224x224 px*, zgodnie z wymaganiami modelu, który został wytrenowany na zbiorze *ImageNet*. Dodatkowo, obrazy normalizowane są według wartości średnich i odchyleń standardowych dla kanałów RGB:

- **Średnie:** (0.485, 0.456, 0.406)
- Odchylenie standardowe: (0.229, 0.224, 0.225)

Warstwy W Pełni Połączone

Do modelu ResNet-18 dodano dwie warstwy w pełni połączone:

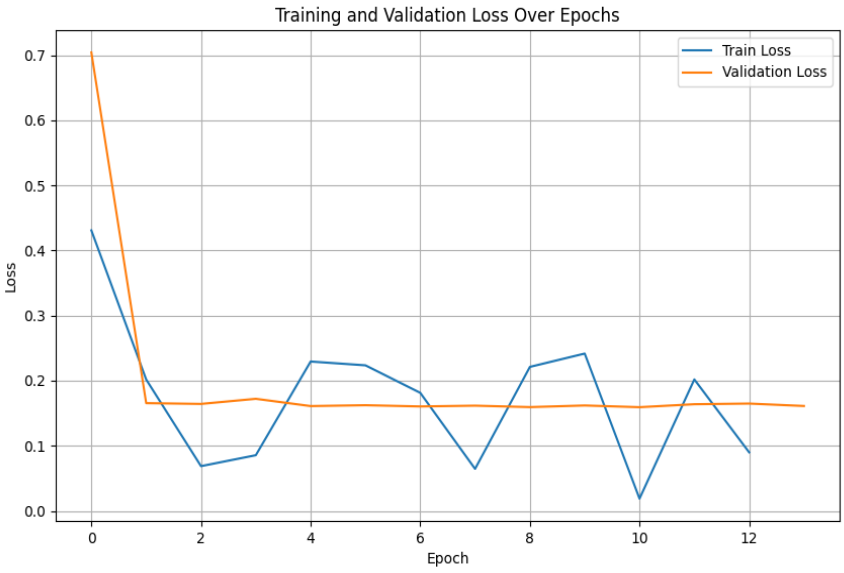
- **Warstwa 1:** Linear(model.fc.in_features, 128) – Spłaszcza dane do wektora o rozmiarze 128.
- **Warstwa 2:** Linear(128, 1) – Końcowa warstwa klasyfikująca z jedną jednostką, ponieważ jest to klasyfikacja binarna.

Transfer Learning i Fine-Tuning

Zastosowano **transfer learning** (wykorzystanie nabytej przez model wiedzy przy podobnym zadaniu) oraz **fine-tuning**, adaptując model ResNet-18 do nowego zadania. Po wstępnym załadowaniu wag z ImageNet, model przeszedł proces fine-tuningu, w którym zamrożono wagi w początkowych warstwach, a następnie dostosowano wyższe warstwy do specyfiki danych wejściowych, w tym warstwy w pełni połączone.

Optymalizacja

Do trenowania modelu użyto **Adam** jako optymalizatora, zapewniającego adaptacyjne tempo uczenia, co było kluczowe w procesie fine-tuningu. Funkcją celu była **Binary Cross Entropy Loss**, odpowiednia dla klasyfikacji binarnej, co pozwoliło na efektywne rozróżnienie pomiędzy obrazami z uśmiechem a bez niego.

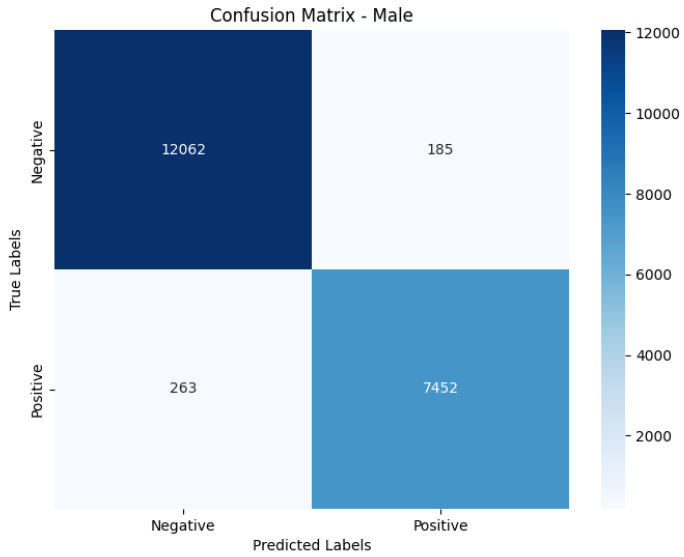


Rysunek 4 Wartosc funkcji celu per epoka.

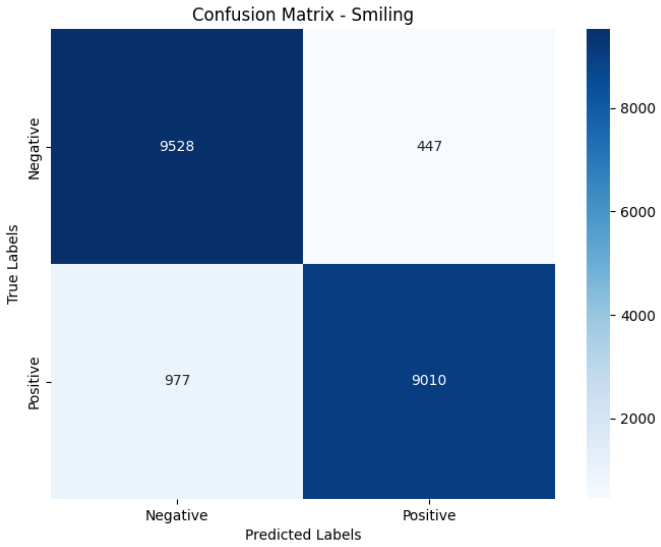
Podsumowanie

Model **ResNet-18** z pretrenowanymi wagami z **ImageNet** został skutecznie zaadoptowany do zadania klasyfikacji obrazu „Smiling” przy pomocy **transfer learning** i **fine-tuningu**. Zastosowane techniki pozwoliły na efektywne wykorzystanie wstępnie wytrenowanych wag oraz dostosowanie modelu do specyfiki nowych danych.

Wyniki testów obydwu modeli na danych testowych dostępnych w CelebA



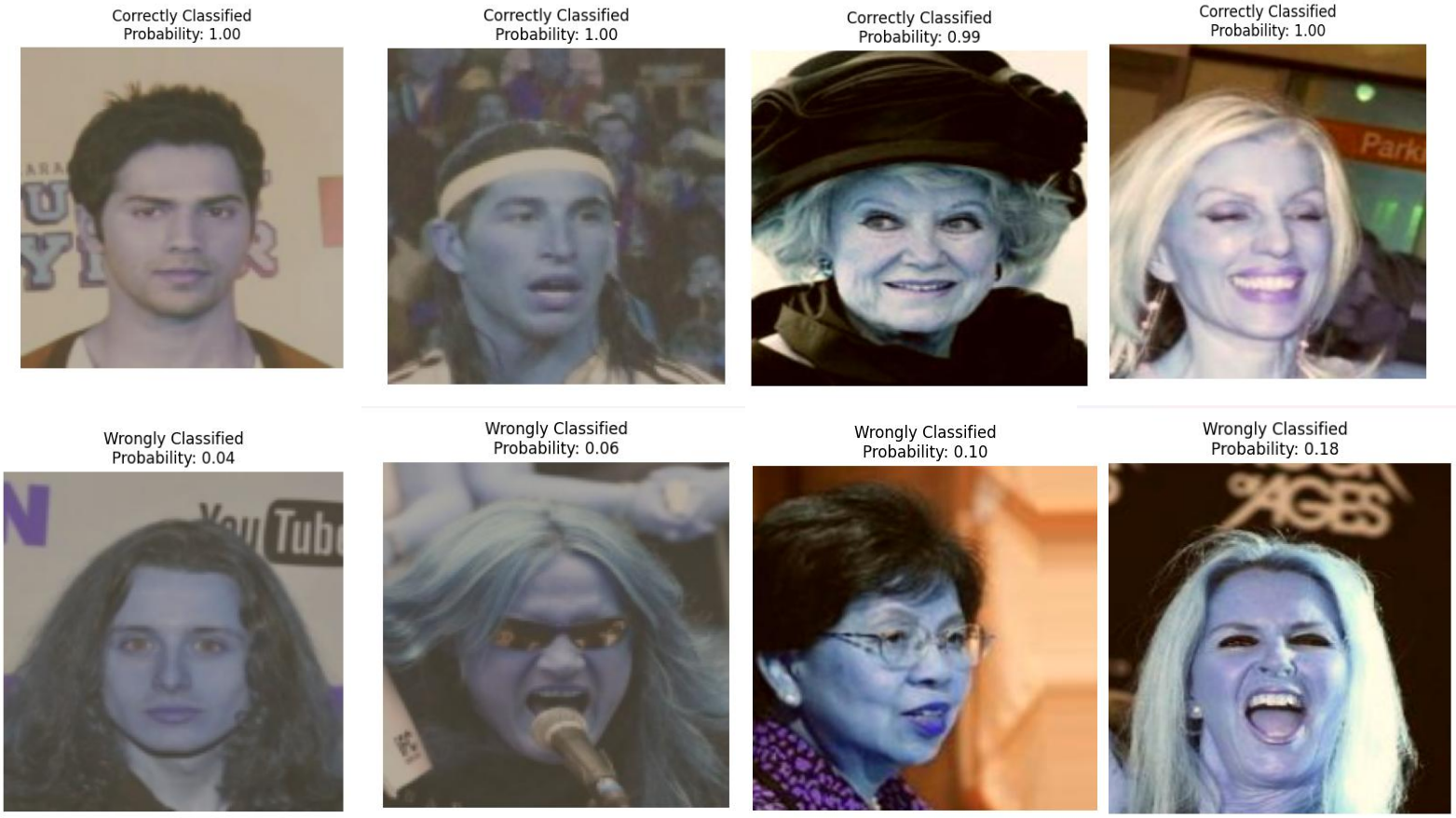
Rysunek 5 Macierz pomyłek dla naszego modelu - atrybut „Male”.



Rysunek 6 Macierz pomyłek dla modelu ResNet-18 – atrybut „Smiling”.

Accuracy dla naszego modelu to 97.57%.

Accuracy modelu to 92.86%



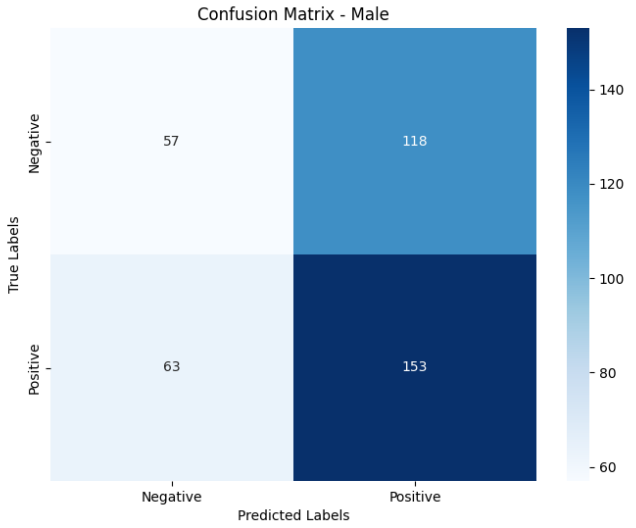
Model klasyfikujący atrybut „Male” świetnie radzi sobie na zdjęciach ze zbioru CelebA, może pochwalić się dokładnością predykcji na poziomie 95%. Zamieszczone przykłady obrazują zdjęcia poprawnie skategoryzowane (dodatkowy warunek to pewność predykcji na poziomie większym niż 80%) oraz zdjęcia niepoprawnie skategoryzowane (dodatkowy warunek to pewność predykcji na poziomie mniejszym niż 40%). Widać, że model ma skłonność do kategoryzowania osób z długimi włosami jako kobiety, co obrazują zamieszczone przykłady, natomiast dla postaci Sergio Ramosa, który akurat na zdjęciu posiada dłuższe włosy, był już w 100% pewien swojej poprawnej decyzji. Również w przypadku modelu klasyfikującego atrybut „Smiling” występuje problem przy krzyku, który został sklasyfikowany jako uśmiech – model prawdopodobnie wyuczył się konkretnego ustawienia dolnej części twarzy człowieka, sugerując, że otwarta buzia będzie oznaczała uśmiech.

Wyniki testów obydwu modeli na danych ze zbioru WIDERFACE

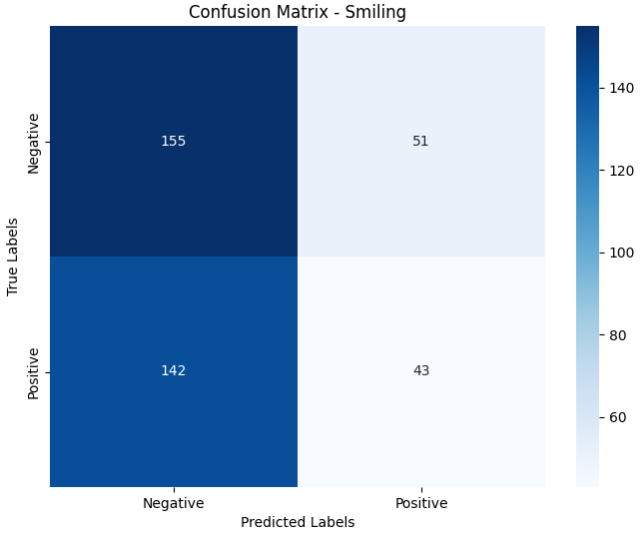
Zbiór testowy WIDERFACE został przygotowany w dwóch etapach: automatycznym oraz ręcznym. W początkowej fazie opracowano algorytm do wstępnej filtracji zdjęć, który selekcjonował obrazy zawierające od 3 do 5 twarzy. Algorytm ten, na podstawie dostępnych bounding boxów, wyodrębniał poszczególne twarze i zapisywał je jako osobne pliki w formacie JPG.

Jednakże proces ten napotkał na trudności związane z obsługą plików. Problemy dotyczyły głównie zbyt długich nazw plików oraz komplikacji związanych ze ścieżkami dostępu, co ograniczyło możliwość przetworzenia całego zbioru do mniej niż 50% zdjęć.

Po zakończeniu etapu automatycznego, wybrane obrazy poddano ręcznej obróbce. Dla wyodrębnionych twarzy dodano adnotacje określające wartości wybranych atrybutów. Proces ten, choć precyzyjny, był podatny na błędy ludzkie (tzw. **human error**), co mogło wpłynąć na dokładność i spójność wprowadzonych danych – proces oceny wybranych atrybutów na zdjęciu jest dosyć subiektywny i może odbiegać od wyuczonych wzorów przez model.



Rysunek 7 Macierz pomyłek dla naszego modelu – atrybut „Male”..

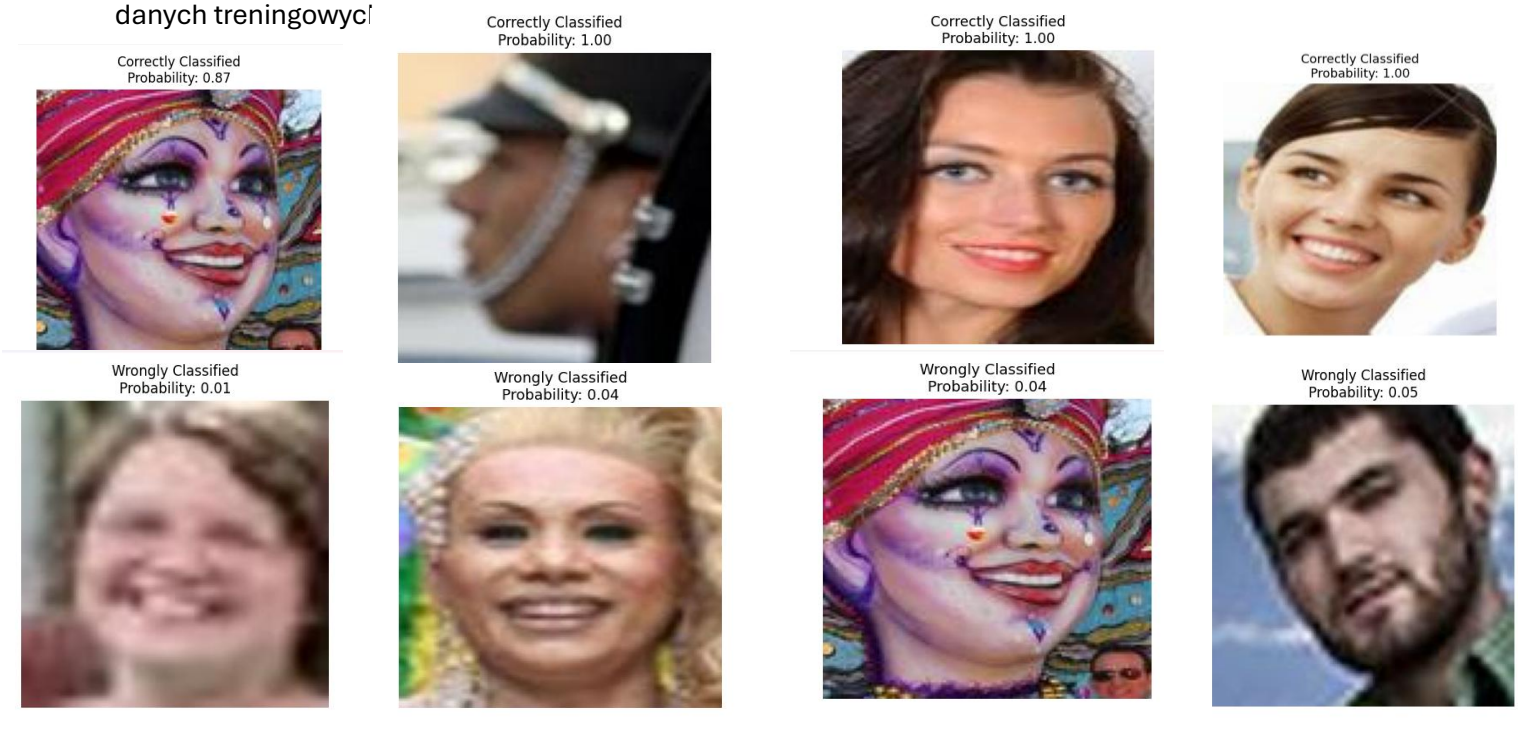


Rysunek 6 Macierz pomyłek dla modelu ResNet-18 – atrybut „Smiling”.

Dokładność naszego modelu na zbiorze testowym wyniosła **53,8%**, podczas gdy model bazowy ResNet-18 osiągnął dokładność na poziomie **50,64%**. Obie wartości są zbliżone do losowego wyboru, co jest charakterystyczne dla problemów binarnej klasyfikacji przy braku wyraźnych wzorców w danych testowych.

Tak niskie wyniki mogą wynikać z dwóch kluczowych czynników:

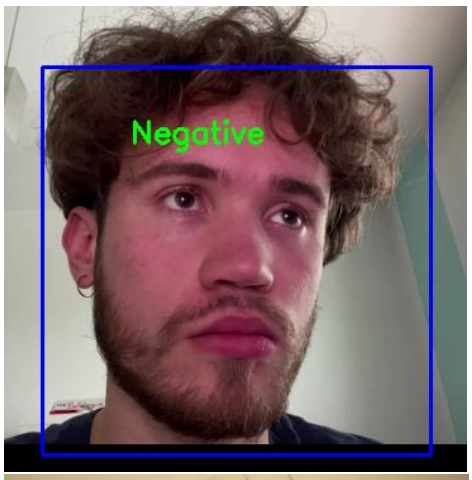
1. **Przetrenowanie modelu na zbiorze CelebA** – model został zoptymalizowany pod kątem specyficznych cech tego zbioru, co ograniczyło jego zdolność do generalizacji na inne dane.
2. **Ograniczone możliwości generalizacji** – pomimo zastosowania augmentacji danych w trakcie treningu, model nie wykazał wystarczającej elastyczności w rozpoznawaniu cech w zbiorze WIDERFACE, który znacząco różnił się od danych treningowyc

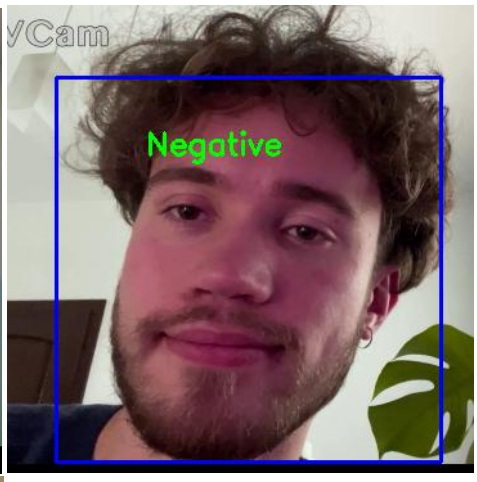


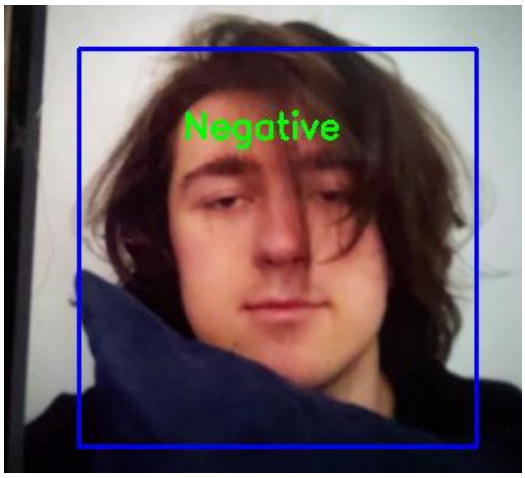
Wyniki testów modeli w programie obsługującym obraz z kamery

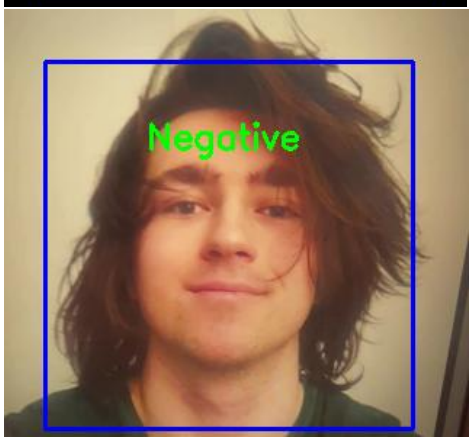
Wytrenowane na zbiorze CelebA modele słabo radzą sobie w klasyfikacji cech na rzeczywistym obrazie z kamery – mimo zastosowania odpowiedniej augmentacji na danych treningowych (wprowadzenie szumu, zmiana koloru, saturacji, odbicie lustrzane), przyczyniło się to do nieznacznego wzrostu dokładności modelu.

- Testowanie modelu ResNet-18 – atrybut „Smiling”: model nie radzi sobie z ekstrakcją atrybutu uśmiechu

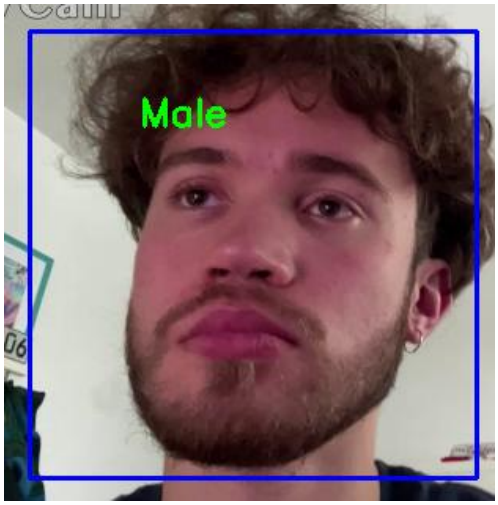








- Testowanie naszego modelu – atrybut „Male”: patrząc na próbkę badawczą w postaci dwóch mężczyzn model działa perfekcyjnie, natomiast testując go na kobietach, często dawał false positive, co odzwierciedla wyniki załączone w macierzy pomyłek.



Opis architektury modelu do detekcji twarzy

Do zadania detekcji twarzy wybrano model **Faster R-CNN ResNet-50 FPN**, oparty na architekturze **Feature Pyramid Network (FPN)**, ze względu na jego zaawansowaną strukturę, skuteczną w zadaniach detekcji obiektów. Model ten jest szeroko wykorzystywany ze względu na swoją efektywność w generowaniu regionów zainteresowania (ROIs) oraz precyzyjnej detekcji obiektów w obrazach.

Aby wykorzystać model w zadaniu detekcji twarzy, przeprowadzono następujące czynności na zbiorze treningowym: przygotowano dane wejściowe w postaci zdjęć oraz odpowiadających im adnotacji, zawierających informacje o pozycjach twarzy w obrazach (bounding boxy). Zbiór danych pochodził z pliku tekstowego, który wymagał odpowiedniego parsowania, aby przypisać dane do właściwych obrazów. W trakcie tego procesu napotkano problem związany z nadmiernie długą ścieżką do pliku, co prowadziło do błędów podczas treningu. Po rozwiązaniu tego problemu (pominięciu wadliwych z poziomu systemowego zdjęć) każde zdjęcie zostało przeskalowane do rozdzielczości **800x800 pikseli**, co jest wymaganym rozmiarem wejściowym modelu Faster R-CNN, który był trenowany na danych o tym rozmiarze.

Dodatkowo, zastosowano **normalizację danych**, zgodną z procesem pretrenowania modelu na zbiorze **ImageNet**:

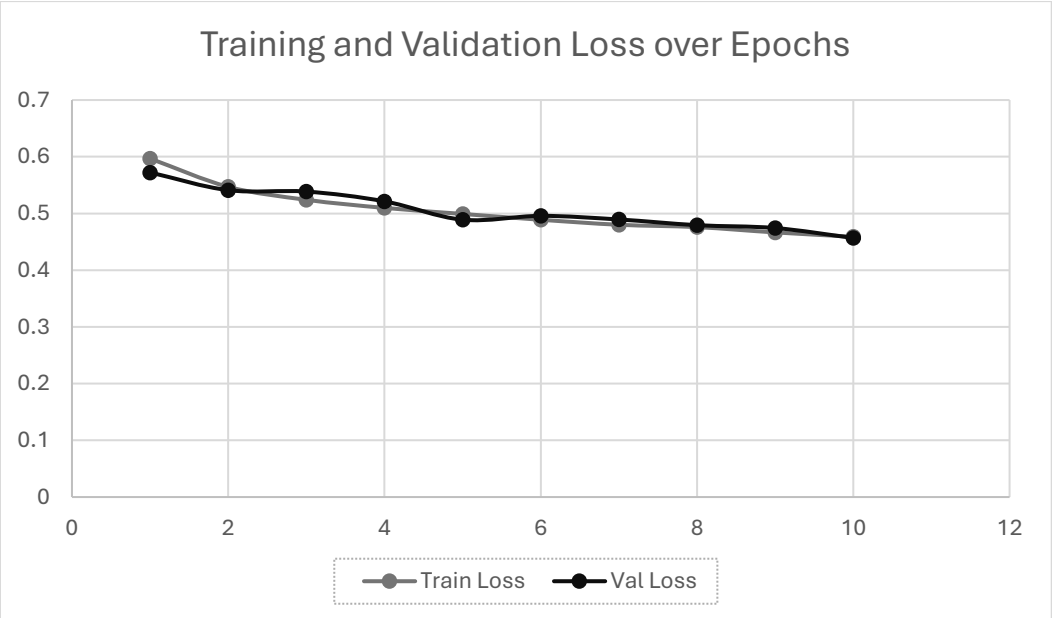
- **Średnie:** (0.485, 0.456, 0.406)
- Odchylenie standardowe: (0.229, 0.224, 0.225)

Aby poprawić zdolność modelu do generalizacji oraz radzenia sobie z różnorodnymi danymi, zastosowano augmentację danych, w tym:

- Obrót zdjęć, co umożliwiło modelowi rozpoznawanie twarzy niezależnie od orientacji.
- Odbicie lustrzane, uwzględniające różne kierunki patrzenia na twarz.
- Delikatne modyfikacje kolorów oraz dodanie szumu, szczególnie w przypadku obrazów z kamerki, co miało na celu zwiększenie odporności modelu na zmienne warunki oświetleniowe oraz różną jakość obrazu.

Proces trenowania odbywał się na zbiorze danych **WIDER FACE**, który zawiera obrazy twarzy w różnych warunkach (np. zmienne oświetlenie, różne kąty). Podczas treningu zastosowano optymalizator **SGD z momentum**, a funkcję celu stanowiła suma strat związanych z klasyfikacją i regresją. Aby zapobiec przeuczeniu, wprowadzono mechanizm **early stopping** na podstawie monitorowania strat walidacyjnych oraz zastosowano harmonogram zmiany współczynnika uczenia (**ReduceLROnPlateau**).

Wybór danych walidacyjnych i treningowych polegał na losowym podziale zbioru danych na zestawy treningowe i walidacyjne w proporcji 80/20. Całkowity czas treningu modelu wyniósł 3h.

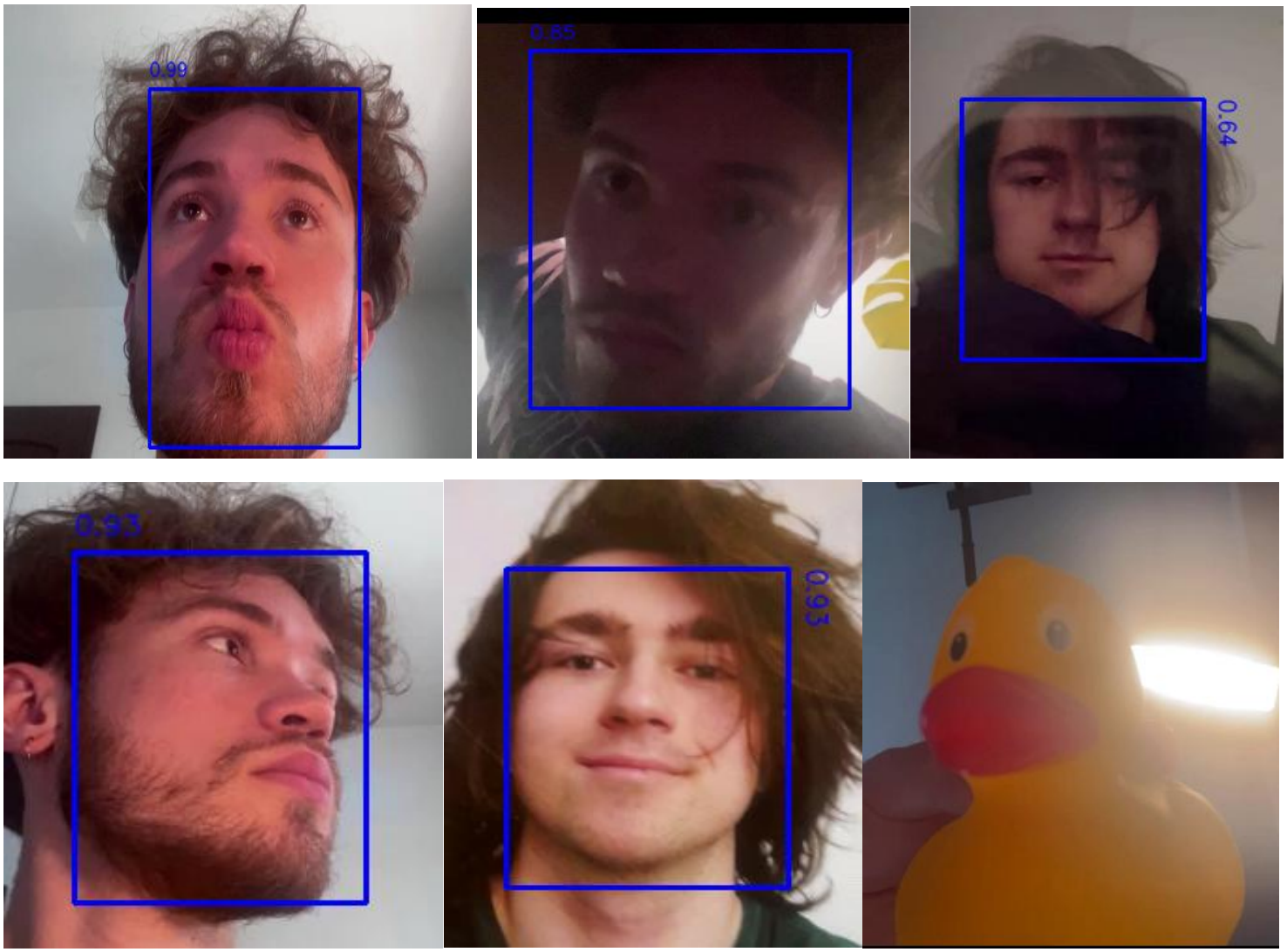


Rysunek 8 Wartosci funkcji celu na zbiorze treningowym i walidacyjnym.

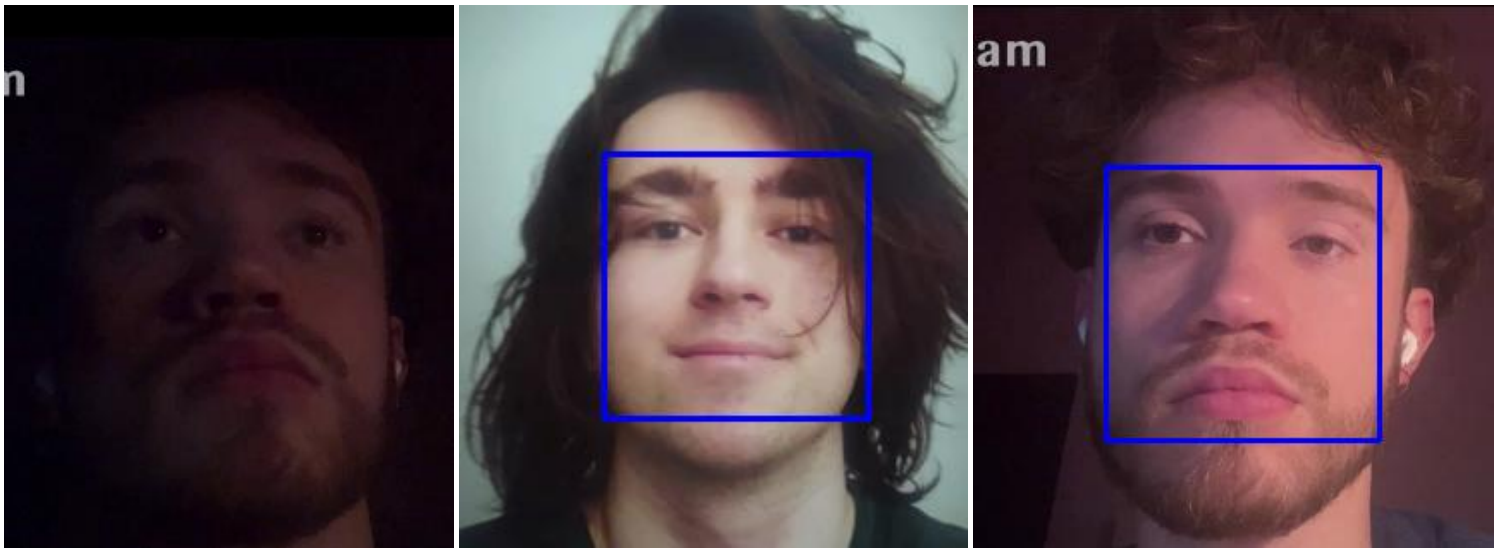
Wyniki działania modelu detekcji twarzy

Model charakteryzuje się dosyć dużą dokładnością, ze względu na fakt wytrenowania go na zbiorze WIDERFACE. Doskonale radzi sobie podczas różnego rodzaju oświetlenia, czy przy większej liczbie osób na kamerze.

- Wytrenowany model detekcji twarzy



- Model kaskadowy



Po przeprowadzeniu szeregu testów na członkach rodziny możemy stwierdzić, że model kaskadowy wypada zdecydowanie gorzej, szczególnie w warunkach słabego oświetlenia, gdzie nie jest w stanie wykryć twarzy znajdującej się przed kamerą. Ponadto, model ten ma trudności w przypadku większej liczby osób, niezależnie od jakości obrazu czy dostępnego oświetlenia. Dla tego zadania, z wyraźną przewagą, lepsze wyniki osiąga wytrenowany model Faster R-CNN z architekturą ResNet-50 FPN.