

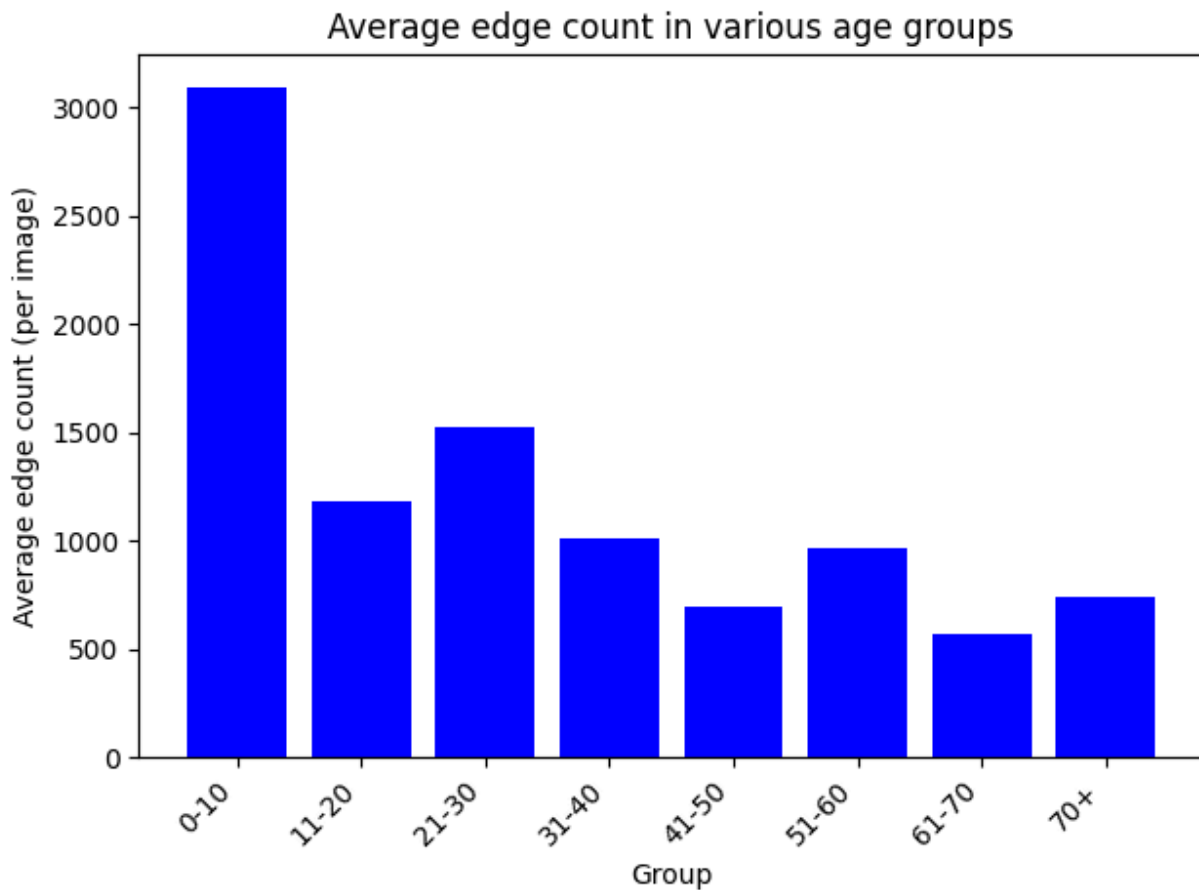
IML face recognition project - documentation

Piotr Głowacki, Damian Kąkol, Jakub Kępka, Piotr Baranowski, Vlada Gromova

EDA

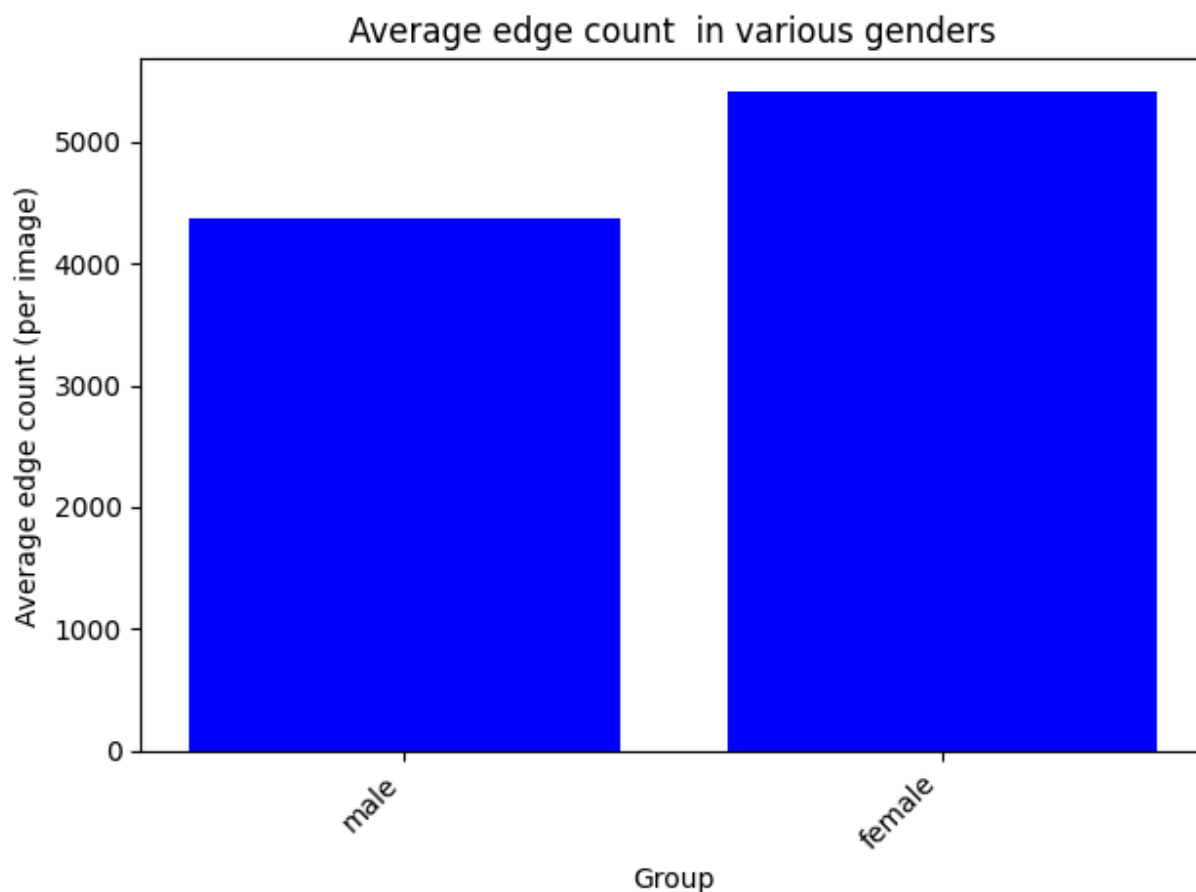
Zbiorem danych, który wykorzystujemy do trenowania modeli rozpoznających wiek jest folder cropped_part1 ze zbioru utk_faces. Zawiera on wycięte z innych zdjęć same twarze, zatem nie ma problemu z detekcją twarzy na zdjęciu. Wykonując EDA policzyliśmy pewne cechy na zdjęciach (thresholding, histogramy RGB, liczba krawędzi), a także przeanalizowaliśmy w jakim stopniu różne grupy osób reprezentowane są w zbiorze.

Liczebność grup

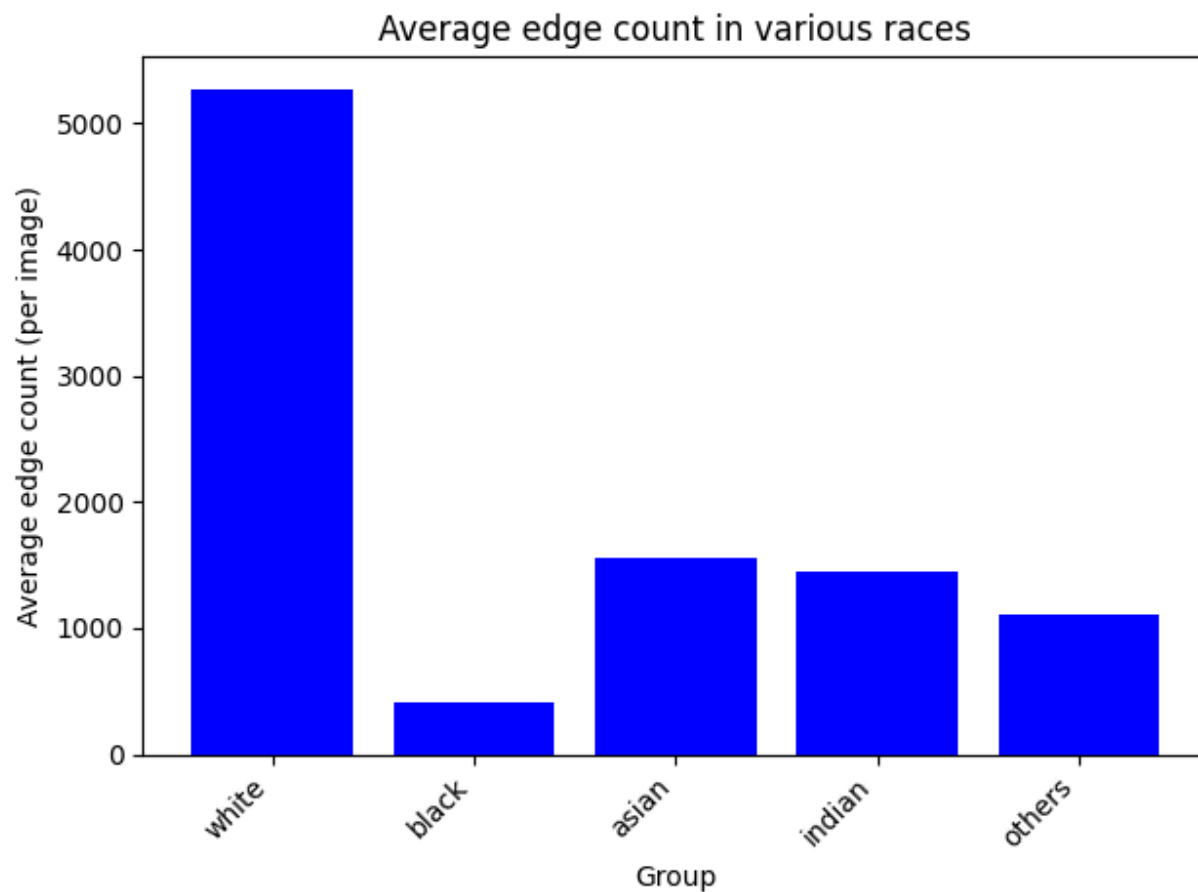


W naszym zbiorze danych każda grupa wiekowa (dekada) oprócz najstarszych ma dość liczną

reprezentację, ale dzieci w wieku 0-10 jest parokrotnie więcej niż pozostałych grup wiekowych. Zdjęć osób wyjątkowo starych jest dość mało: po kilka dla danego wieku, a dla wieku 94 nie ma żadnego zdjęcia.

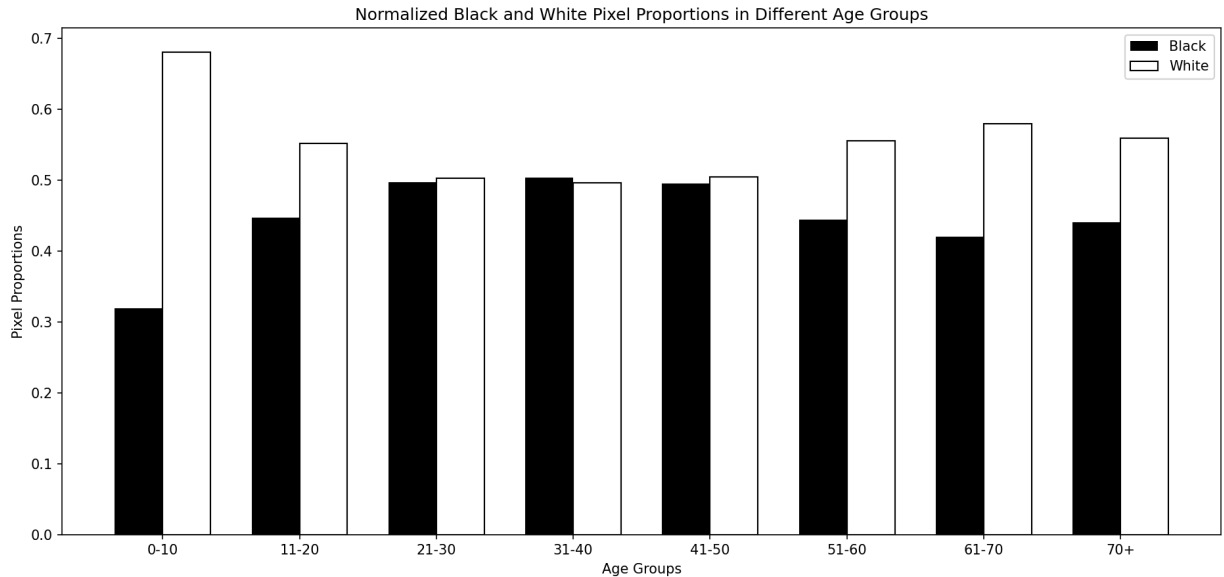


W zbiorze danych jest lekka nadreprezentacja kobiet.



Poszczególne rasy są reprezentowane w zbiorze w zupełnie innym stopniu: jest bardzo dużo osób rasy białej i bardzo mało czarnej. Może to budzić obawy, czy nasz model będzie w stanie dobrze rozpoznawać wiek osoby rasy czarnej.

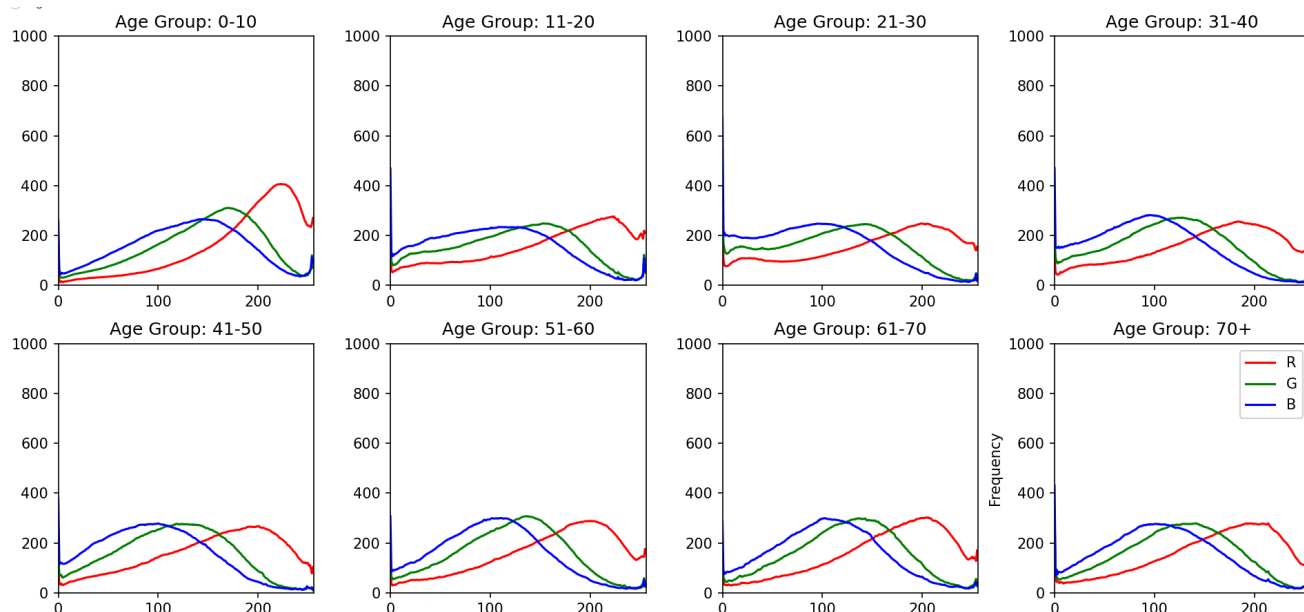
Thresholding



Zgodnie z oczekiwaniami, na zdjęciach młodszych i starszych osób było więcej białych pikseli, niż na zdjęciach osób w średnim wieku. Po pierwsze, to wynika z tego, że skóra dzieci jest bardzo gładka i ma mniej widoczne defekty. Po drugie, często skóra dzieci jest jaśniejsza w porównaniu do skóry starszych osób.

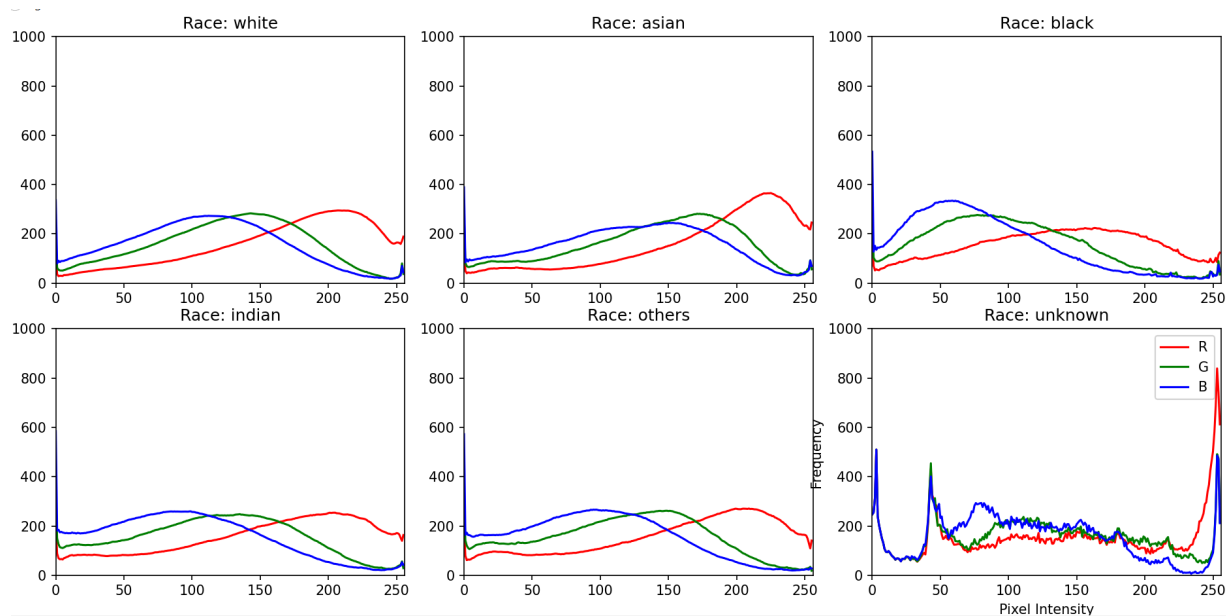
Z innej strony, z wiekiem skóra staje się bardziej podatna na zmarszczki, linie i inne nierówności. Duża liczba takich zmian może być traktowana przez algorytm thresholding'u jako jaśniejszy obszar.

RGB

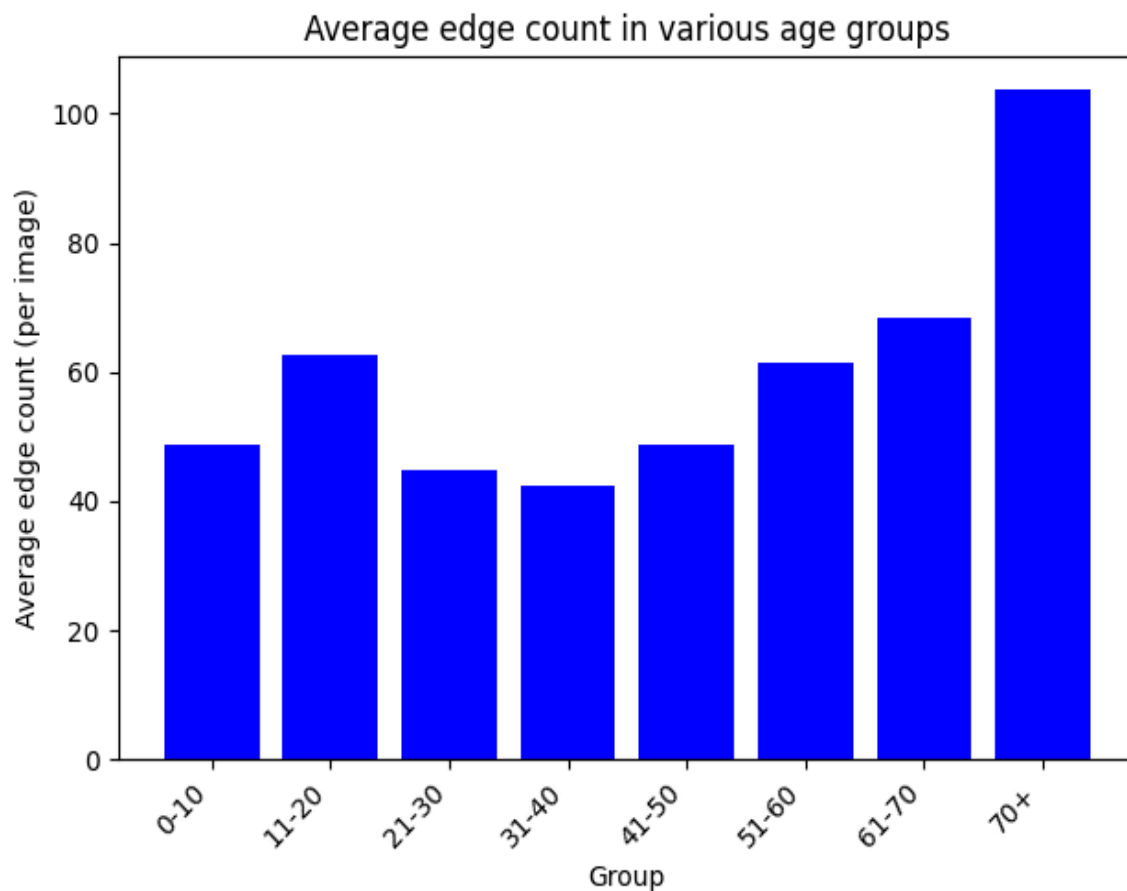


Przyglądając się wykresom, zauważyliśmy, że najbardziej intensywne czerwone pikseli występowały na zdjęciach osób w wieku od 0 do 10 lat. To może być spowodowane tym że skóra młodszych osób może być bardziej wrażliwa na różne bodźce zewnętrzne, co może prowadzić do zaczerwienienia się policzków.

Niestety, nie zaobserwowano widocznej zależności pomiędzy G i B pikselami oraz innymi grupami wiekowymi.

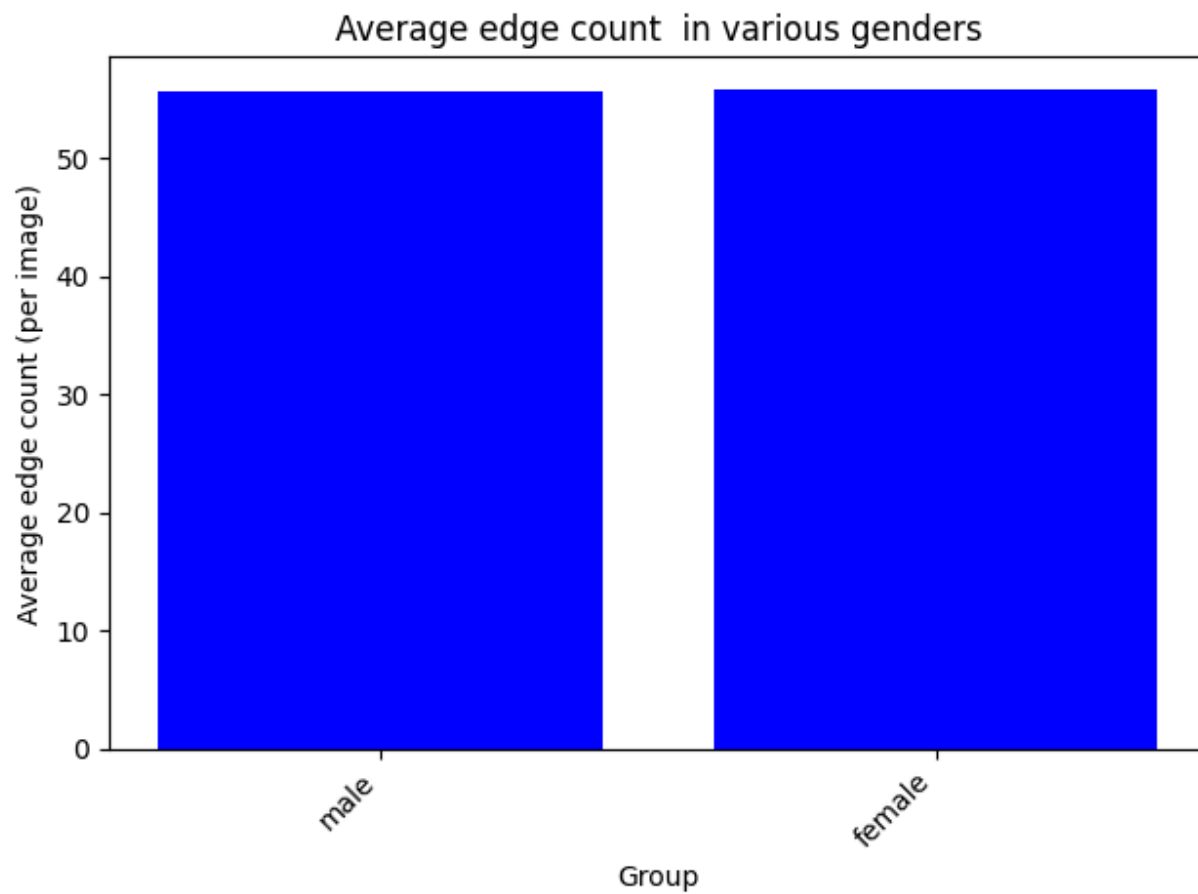


Kolejnym parametrem, który analizowaliśmy w naszym zbiorze danych jest średnia liczba krawędzi w zdjęciu danej grupy.

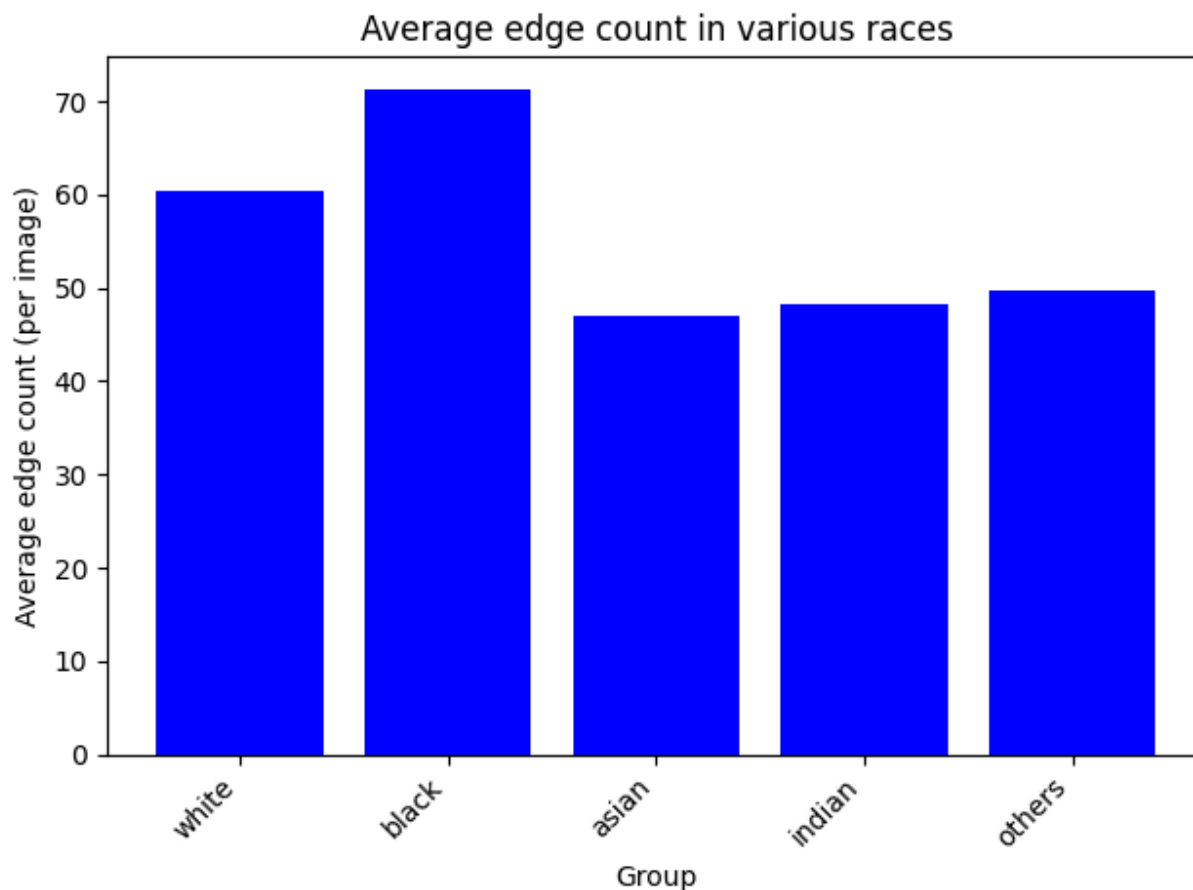


Analiza liczby krawędzi w poszczególnych grupach wiekowych częściowo potwierdziła nasze przypuszczenia. Osoby starsze (70+) mają więcej zmarszczek, stąd większa liczba krawędzi wykrytych na ich zdjęciach. To pozwala przypuszczać, że model będzie w stanie odróżnić je od osób młodszych.

Natomiast dla grup wiekowych w przedziale 0-60 nie ma określonego trendu, najmniej krawędzi jest na zdjęciach osób w grupie 31-40 lat, zarówno dla starszych jak i młodszych obserwujemy więcej krawędzi.



Liczba krawędzi jest taka sama dla obydwu płci. U mężczyzn krawędzie były czasem wykrywane na zaroście, natomiast u kobiet - na włosach.



Zdjęcia osób rasy azjatyckiej i hinduskiej zawierają średnio mniej krawędzi niż zdjęcia osób rasy białej i czarnej. Spodziewaliśmy się tego, ponieważ azjaci mają mniej zmarszczek i mają gładze twarze nawet gdy się uśmiechają.

Podsumowując, w naszym zbiorze danych istnieją spodziewane zależności liczby krawędzi od wieku i rasy, co pozytywnie świadczy o jego jakości.

Ewaluacja modeli detekcji twarzy

Do policzenia poniższych wartości użyliśmy zbioru WIDER Face.


```

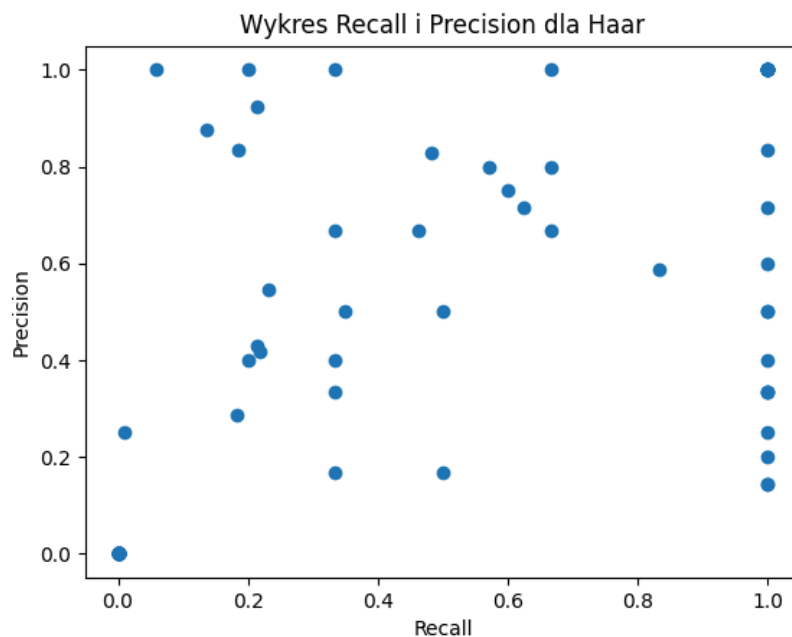
95
Mean Average Precision for haarCascad: 0.36619904862938923
Mean Average Precision for MTCNN: 0.7884390224788955
Precision for haarCascad: 0.49303621169916434
Precision for MTCNN: 0.9120654396728016
Recall for haarCascad: 0.17117988394584138
Recall for MTCNN: 0.43133462282398455
F1 score for haarCascade: 0.2541277817659727
F1 score for MTCNN: 0.5856861457649376

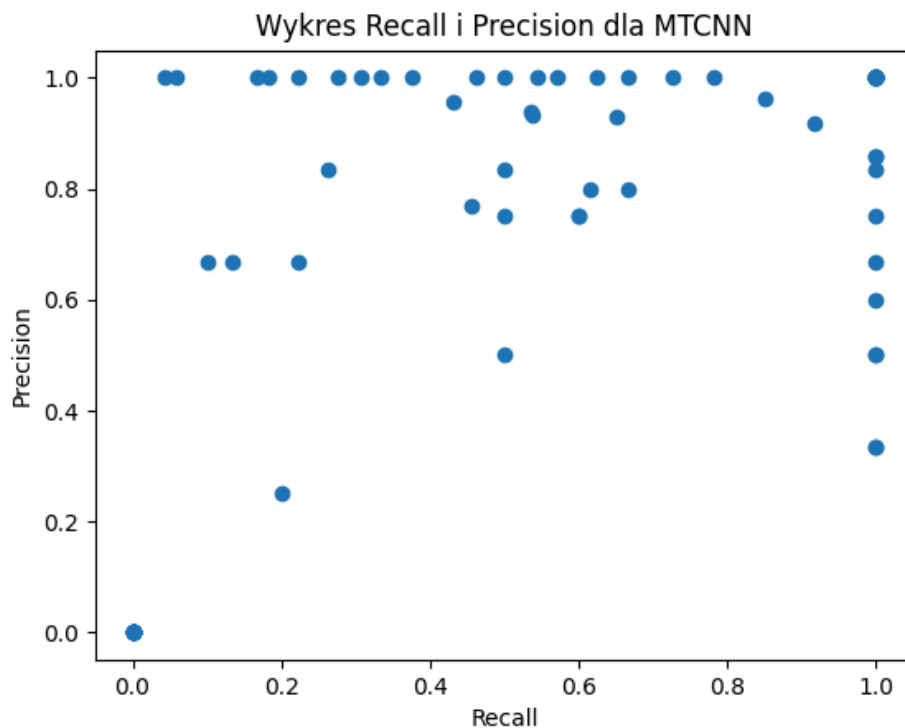
```

Wybrane Metryki dla modeli MTCNN i HaarCascade

	MTCNN	HaarCascade
MAP	0.78	0.36
Precision	0.91	0.49
Recall	0.43	0.17
F1	0.58	0.25

Wykresy Precision/Recall w ramach





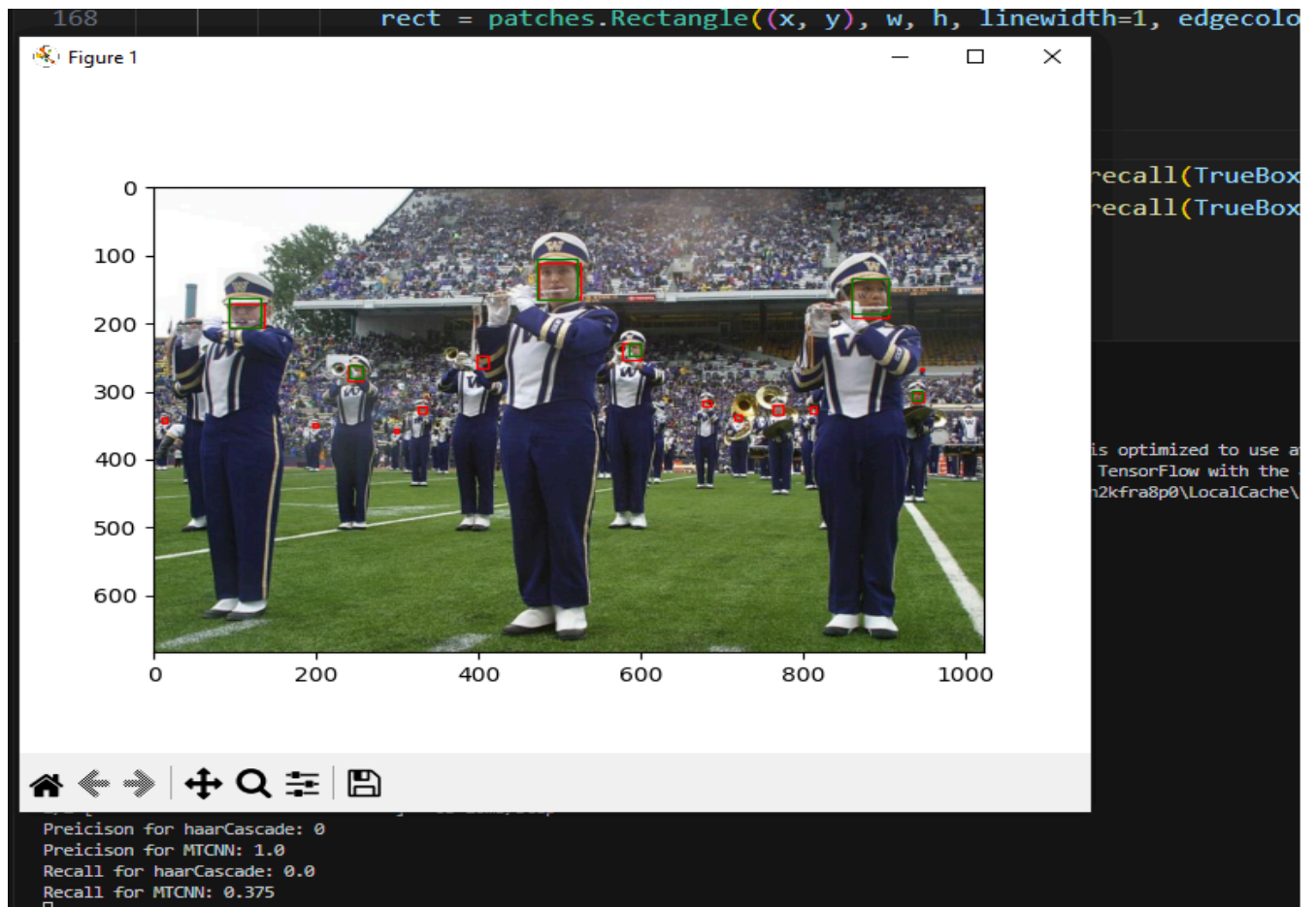
Obserwujemy zaburzenie na tym wykresie ale charakterystyczny rozkład punktów na wykresach jest zachowany (to znaczy aby uzyskać wysokie precision trzeba poświęcić recall i na odwrót). Wykresy potwierdzają wyższość modelu MTCNN nad Haar. Punkty w prawym górnym rogu przedstawiają proste zdjęcia z małą ilością twarzy z którymi model dobrze sobie poradził. Zaburzenia na wykresach wynikają ze specyfiki zbioru WIDER Face, który zawiera wiele zdjęć, które stanowią duże wyzwanie dla modeli detekcji twarzy, co przedstawiają poniższe case studies.

Kolor czerwony - boxy z datasetu

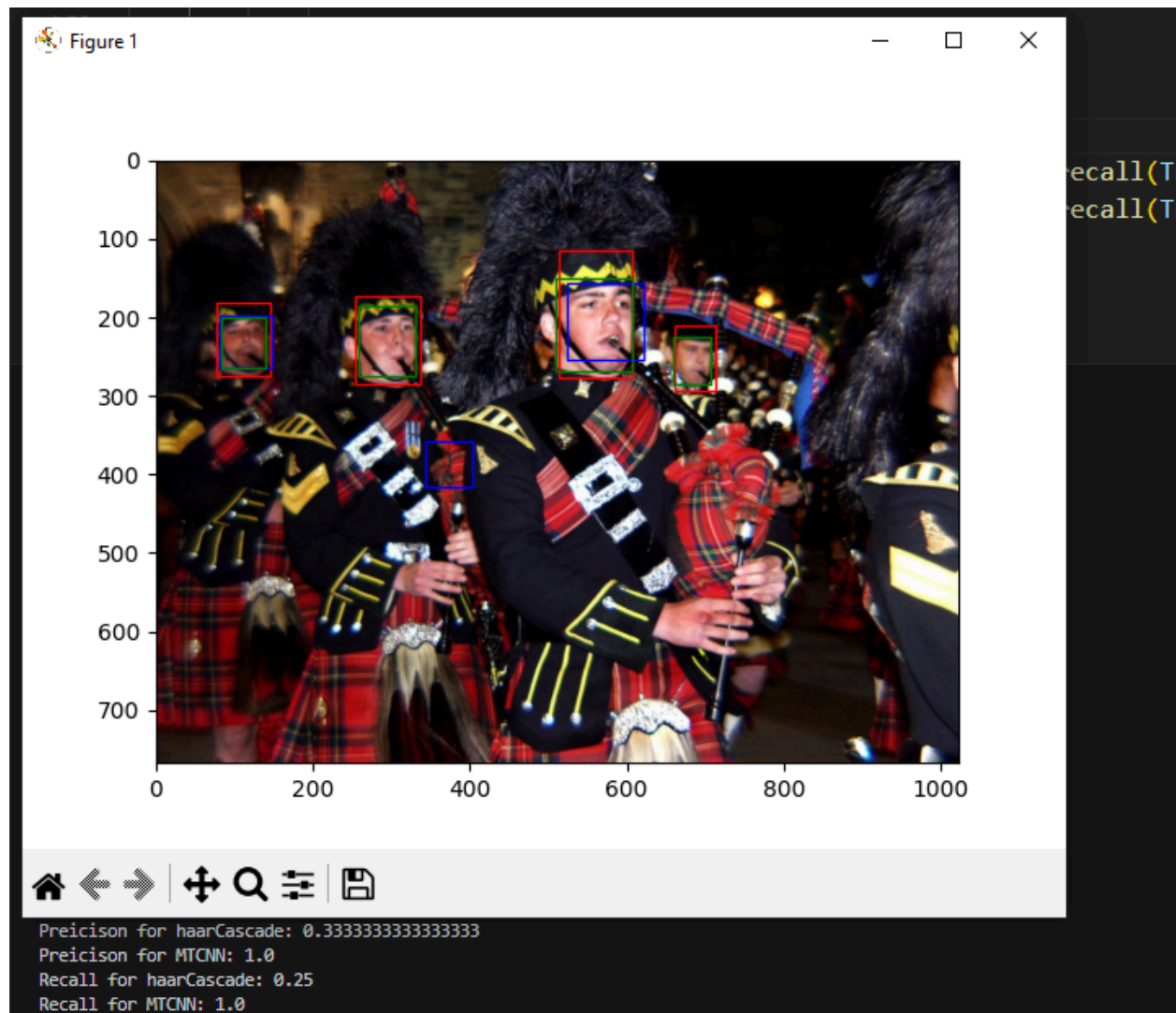
Kolor zielony - MTCNN

Kolor niebieski - HaarCascade

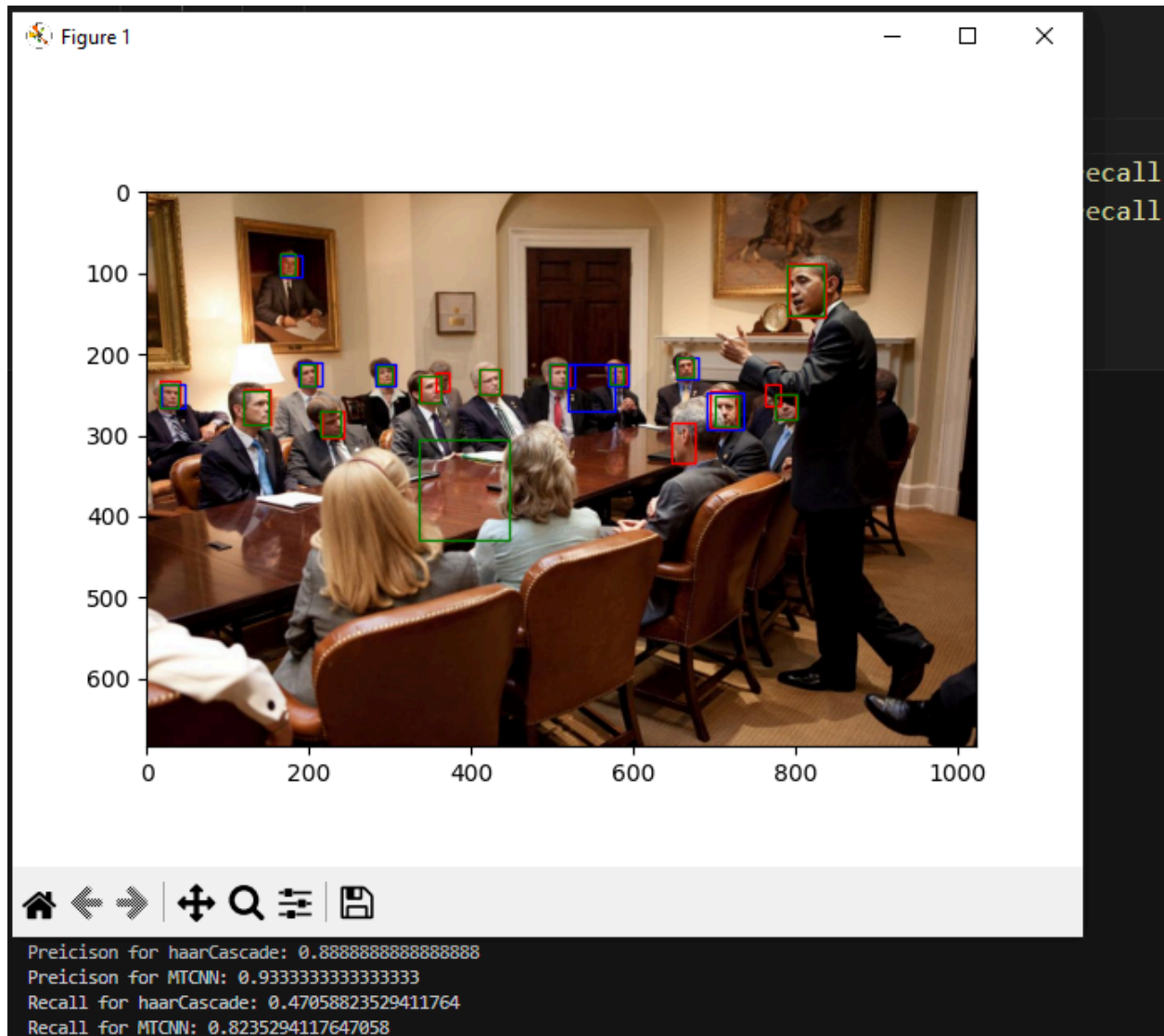
CASE STUDY 1



CASE STUDY 2

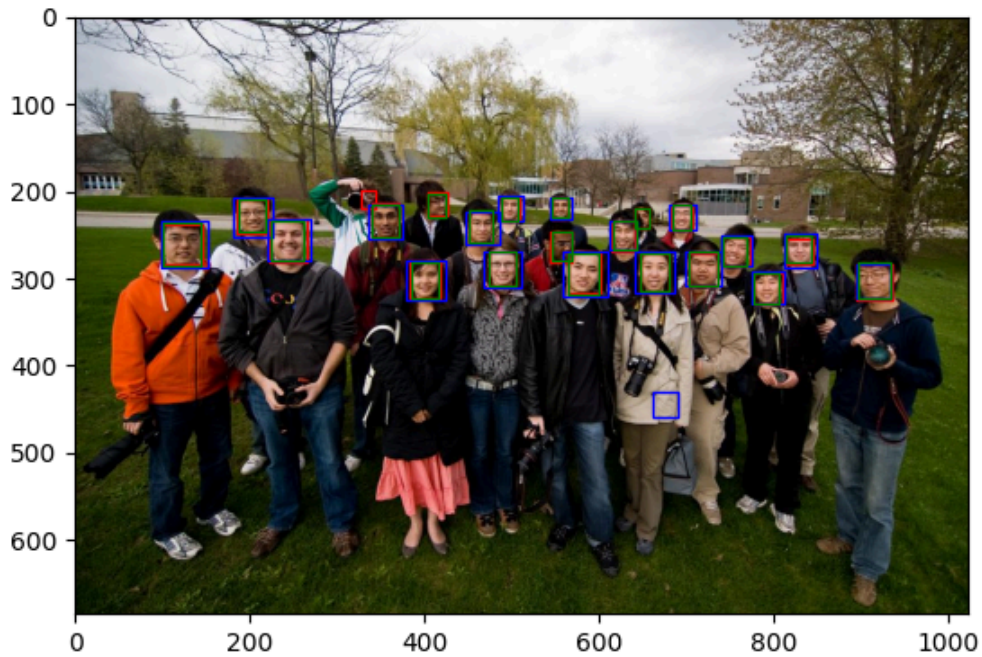


CASE STUDY 3



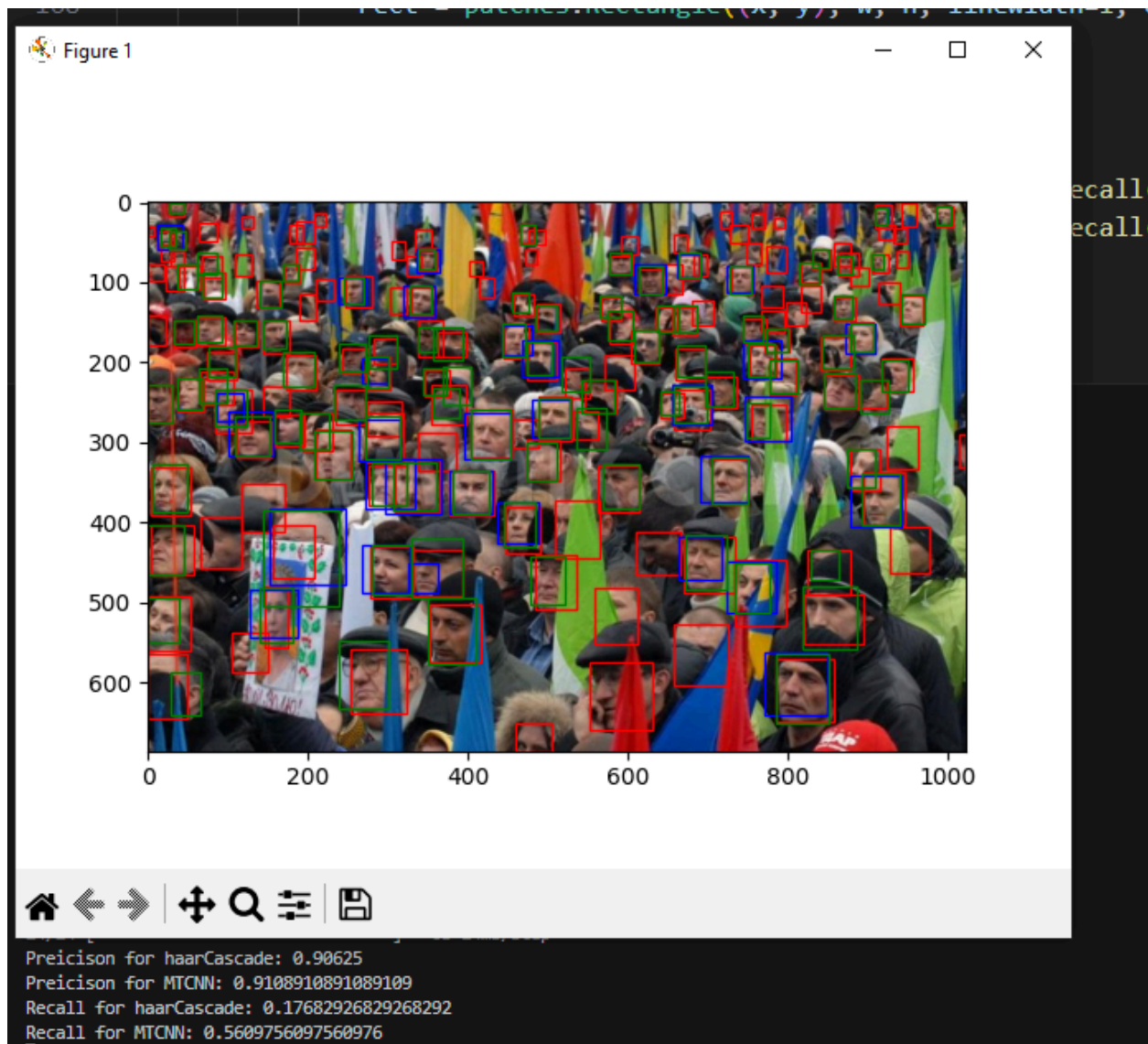
CASE STUDY 4

Figure 1



```
Precision for haarCascade: 0.9473684210526315
Precision for MTCNN: 1.0
Recall for haarCascade: 0.8181818181818182
Recall for MTCNN: 0.9545454545454546
```


CASE STUDY 5



Podsumowanie:

MTCNN przewyższa jednoznacznie HaarCascade pod względem wszystkich metryk jednakże dzieje się to kosztem długości obliczeń co w rezultacie może oznaczać że MTCNN nie nadaje się do detekcji twarzy w kamerach internetowych lub video.

Korelaty atrybutów twarzy z wiekiem osoby

Badania dotyczące jakie cechy twarzy wysoko korelują z wiekiem osoby:

1)

Podsumowując, wyniki tych badań sugerują, że na postrzeganie starzenia duży wpływ ma wygląd okolic ust (objętość i zmarszczki górnej wargi) oraz okolic oczu (kurze łapki i zmarszczki pod oczami, cienie, worki) pod oczami) i jednolitość kolorytu skóry (brązowe plamy, jednolity koloryt skóry).¹

2)

Podsumowanie:

Używamy metody przewidywania regresji przy pomocy maszyny wektorów nośnych (SVM), aby przewidzieć wiek na podstawie wyodrębnionych cech tekstury i punktów orientacyjnych. W porównaniu z cechami tekstury twarzy wyniki przewidywania oparte na punktach orientacyjnych twarzy są lepsze. Sugeruje to, że cechy morfologiczne twarzy zawarte w punktach orientacyjnych twarzy mogą lepiej odzwierciedlać wiek twarzy niż cechy tekstury twarzy. W połączeniu z charakterystycznymi cechami twarzy i teksturami można poprawić skuteczność przewidywania wieku.

Wnioski:

Na podstawie wyników eksperymentów można stwierdzić, że cechy tekstury w połączeniu z charakterystycznymi punktami twarzy są przydatne do przewidywania wieku.²

¹ Nkengne, A., Stamatias, G.N., Bertin, C. (2015). Facial Skin Attributes and Age Perception. In: Farage, M., Miller, K., Maibach, H. (eds) Textbook of Aging Skin. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-1-4939-9888-8_10.

² Wang M, Chen W. Age prediction based on a small number of facial landmarks and texture features. Technol Health Care. 2021;29(S1):497-507. doi: 10.3233/THC-218047. PMID: 33682786; PMCID: PMC8150531.

Strategia przypisywania klas dla ludzi starszych

Przyjeliśmy konwencję w której do 85 roku każda klasa wiekowa ma długość jeden, a powyżej tego wieku tworzymy 5-letnie przedziały, tzn. 85-90,90-100,105-110. Dane w test secie mają oryginalne etykiety wiekowe (Dla danych do testu i trenowania są oddzielne generatory).

Augmentacja Danych dla modeli i Oversampling

W celu redukcji znaczenia nadmiarowości danych dla zdjęć z niską wartością wieku przyjęliśmy strategię oversamplingu dla starszych grup wiekowych w sposób liniowy dodając dane wygenerowane w ramach augmentacji. W wyniku ograniczeń technicznych platformy google colab nie możliwe było korzystanie z gotowych funkcji, ponadto aby móc dokonać augmentacji danych wymagane było zastosowanie generatorów danych. Poniżej znajduje się kod funkcji która dodaje sampling oraz generator użyty do generowania dodatkowych zdjęć

```
for i in range(len(train_images)):
    print(i)
    image = train_images[i]
    label = train_labels[i]
    image_filename = f'{label}_{i}.jpg'
    image_path = os.path.join(train_output_folder, image_filename)

    image = (image * 255).astype(np.uint8)
    pil_image = Image.fromarray(image)
    pil_image.save(image_path)
    batch_size = int(label / 40)
    augmented_images = datagen.flow(np.expand_dims(image, 0), batch_size=batch_size, save_to_dir=image_path)
    for i in range(batch_size):
        next(augmented_images)
```

```
datagen = ImageDataGenerator(  
    rotation_range=20,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    fill_mode='nearest'  
)
```

W wyniku eksperymentów zrezygnowaliśmy z używania zdjęć z wykrytymi krawędziami za pomocą cv2.Canny gdyż ciężko było ustalić sensowną liczbę dodawanych zdjęć aby nie zaburzać procesu uczenia sieci. Ponadto wydaje się że utk_faces nie stanowi dobrego datasetu dla wykrywania zmarszczek, model knn który operował na liczbie białych pikseli w stosunku do liczby wszystkich pikseli na fragmentach policzków i czoła osiągał gorszy błąd niż zwykły knn (około 22 lata), również sama zależność była delikatnie liniowa, a w samym datasecie nie brakuje zdjęć które stanowią odstępstwo od normy w kwestii ilości krawędzi na zdjęciu dla wieku

W rezultacie dla zdjęcia z grupy wiekowej 80 dodawane są 2 dodatkowe zdjęcia z augmentacji, a dla wieku 40 tylko jedno. Analogiczne przetwarzanie jest stosowane dla test setu aby utrzymać preferowany rozkład danych otrzymany z funkcji train_test_split (w tym miejscu znalazł się błąd w kodzie, który spowodował niezbalansowanie datasetu i bias w stronę ludzi starszych)

Ostatecznie w wyniku tego samplingu wygenerowane zostało około 3300 zdjęć co stanowi ponad $\frac{1}{3}$ zbioru wejściowego, co powinno istotnie balansować zbiór danych.

Architektura własnej sieci CNN

Layer (type)	Output Shape	Param #
conv2d_16 (Conv2D)	(None, 98, 98, 128)	3584
max_pooling2d_16 (MaxPooling2D)	(None, 48, 48, 128)	0
conv2d_17 (Conv2D)	(None, 46, 46, 128)	147584
max_pooling2d_17 (MaxPooling2D)	(None, 22, 22, 128)	0
conv2d_18 (Conv2D)	(None, 20, 20, 256)	295168
max_pooling2d_18 (MaxPooling2D)	(None, 9, 9, 256)	0
conv2d_19 (Conv2D)	(None, 7, 7, 512)	1180160
max_pooling2d_19 (MaxPooling2D)	(None, 3, 3, 512)	0
flatten_4 (Flatten)	(None, 4608)	0
dropout_4 (Dropout)	(None, 4608)	0
dense_4 (Dense)	(None, 512)	2359808
age (Dense)	(None, 1)	513

Jako funkcja kosztu została przyjęta metryka Mean Absolute Error, a problem został potraktowany jako problem regresji

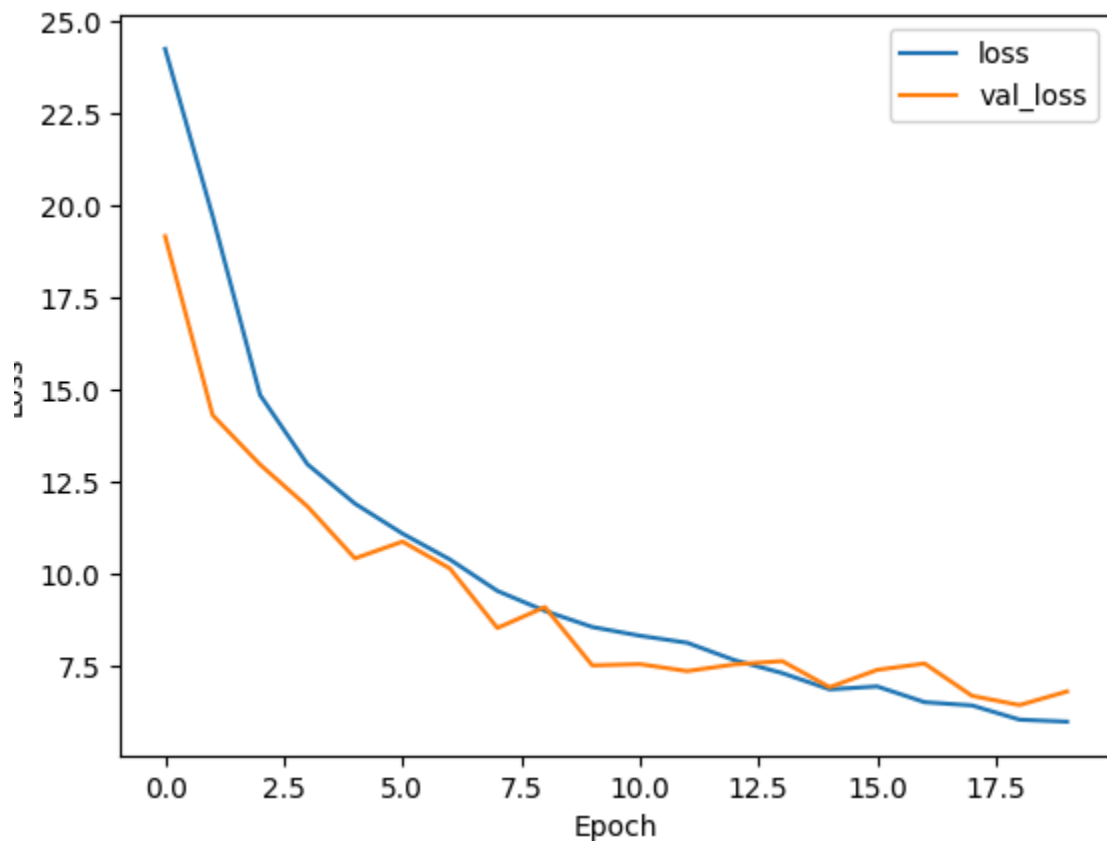
Ustalanie Hyper Parametrów dla własnej konwolucyjnej sieci neuronowej

1. MAE w zależności BatchSize dla epochs 20 i initial learning rate 0.001

32	64	128
7.00	6.89	6.4

Na pierwszy rzut oka warto stosować dla tej konfiguracji batch_size na poziomie 128, co zostało następnie sprawdzone dla epoch o wartości 30 i uzyskany błąd wyniósł 5.96

```
Epoch 30: val_loss did not improve from 5.96081  
61/61 [=====] - 34s 570ms/step - loss: 2.2531 - val_loss: 6.0137
```



Przykładowy trening dla: Training model with learning rate: 0.001, epochs: 20, batch size: 64

```
0s - loss: 6.0622 - mse: 75.2451 - rmse: 8.5874 - mape: 34.0547
5722, saving model to best_regression_instance_model_0.0010_20_64_20240124-032750.h5
378ms/step - loss: 6.0622 - mse: 75.2451 - rmse: 8.5874 - mape: 34.0547 - val_loss: 6.4572 - val_mse: 95.0000
```

Ostateczne Parametry użyte w programie (błąd 5.96 był zapisany w złym formacie)

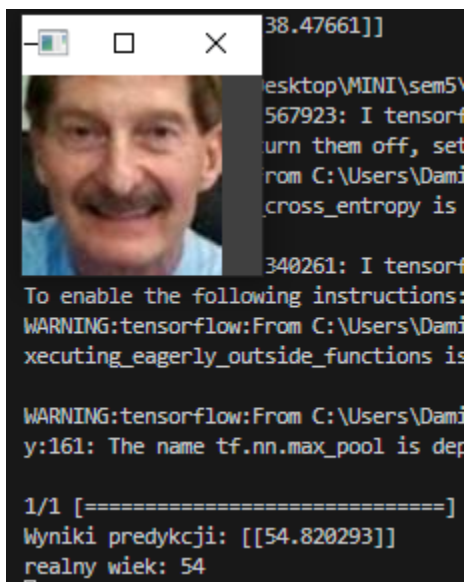
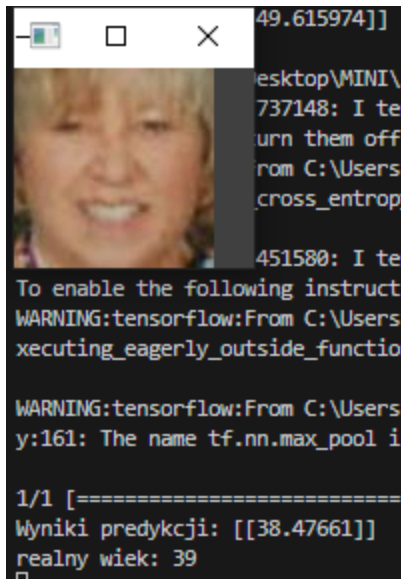
.

```
Epoch 25: val_loss improved from 6.74000 to 6.55927, saving model to best_regression_instance_model_0.0010_25_128_20240124-035410.h5
82/82 [=====] - 60s 743ms/step - loss: 6.1963 - mse: 78.4043 - rmse: 8.8140 - mape: 35.0055 - val_loss: 6.5593 - val_mse: 99.5910 - val_rmse: 9.9800 - val_mape: 52.0694
```

	MAE	MSE	RMSE	MAPE
train	6.19	78.48	8.81	35.085
val	6.5593	99.59	9.9080	52.0694

Case Studies dla własnej sieci z test setu





Widać że model na utk radzi sobie sensownie, niestety w aplikacji obserwujemy bias (przesunięcie) w stronę osób starszych, co wynika z małego błędu przy augmentacji, który sprawia że błąd jest ostatecznie jest liczony na naszą niekorzyść (dodanie osób starszych, likwiduje bias w stronę osób młodych, który niewątpliwie występuje w utk_faces, ale nie uwzględnienie tego w validation set w wyniku błędu w kodzie zwiększa nasz błąd na naszą niekorzyść), żeby polepszyć działanie modelu dla realnych przypadków należałoby zastosować pełny oversampling (co na platformie google colab dostarcza dużych problemów)

Modele detekcji wieku oddzielne dla kobiet i mężczyzn

Kobiety i mężczyźni starzeją się w innym tempie, wiek kobiety koreluje mocniej z innymi czynnikami na twarzy niż wiek u mężczyzn. Dodatkowo tempo fizycznego/widzianego starzenia się różni się w zależności w jakim etapie życia się znajdują. Przykładowo:

Kiedy mężczyźni starzeją się, najbardziej zauważalną zmianą na ich twarzach są głębokie zmarszczki na czole, a także zaczyna się pojawiać fałda pod brodą, niektórzy mężczyźni nawet mogą mieć mniejszą brodę w miarę starzenia się. Brew mężczyźni i ich powieki również opadają, gdy osiągną pewien wiek; dlatego wielu starszych mężczyzn wygląda na zagniewanych i marudnych, mimo że tak się naprawdę nie czują.

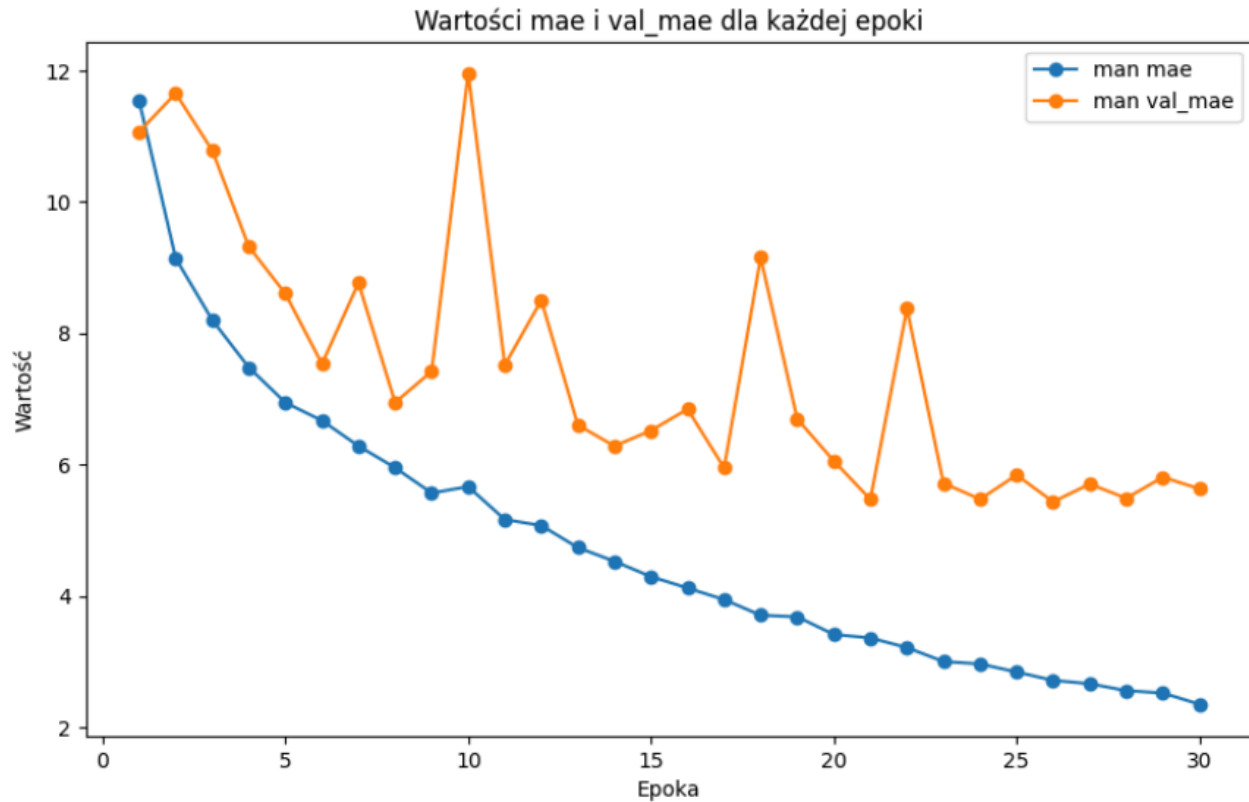
U kobiet zmarszczki nie są tak głębokie jak u mężczyzn, zazwyczaj to tylko drobne linie wokół oczu i ust, zwane popularnie "liniami uśmiechu" lub "nóżkami wrony". Niektóre kobiety także mogą rozwinąć fałd pod brodą, ale zazwyczaj skóra wokół szyi opada, co skutkuje tzw. "kurczakiem" na szyi.³

Dlatego zdecydowaliśmy się na wytrenowanie dwóch modeli. Jeden model służy do predykcji wieku na podstawie zdjęcia twarzy mężczyzny, drugi model służy do predykcji wieku na podstawie zdjęcia twarzy kobiety. Następnie łączymy te dwa modele przy predykcji wieku losowo wybranej twarzy. Dla danego zdjęcia odgadujemy płeć, następnie używamy wyspecjalizowanego modelu dla danej płci. Takie rozwiązanie pozwala na lepszy wynik, ponieważ modele wyspecjalizowane dla jednej płci mają lepsze wyniki niż modele ogólne, w dodatku zgadywanie płci ma bardzo dużą skuteczność.

Modele wyżej wymienione to UTK_faces_man.h5 i UTK_faces_woman.h5. Oba to modele regresyjne z wejściem będącym obrazem rozmiaru (64, 64). Oparte są na zmodyfikowanej architekturze ResNet-50, dokładny opis modelu znajduje się w pliku model_resnet_summary.txt.

³ How Do Faces Of Men And Women Age Differently? - <https://kopelsonclinic.com/how-do-faces-of-men-and-women-age-differently/>

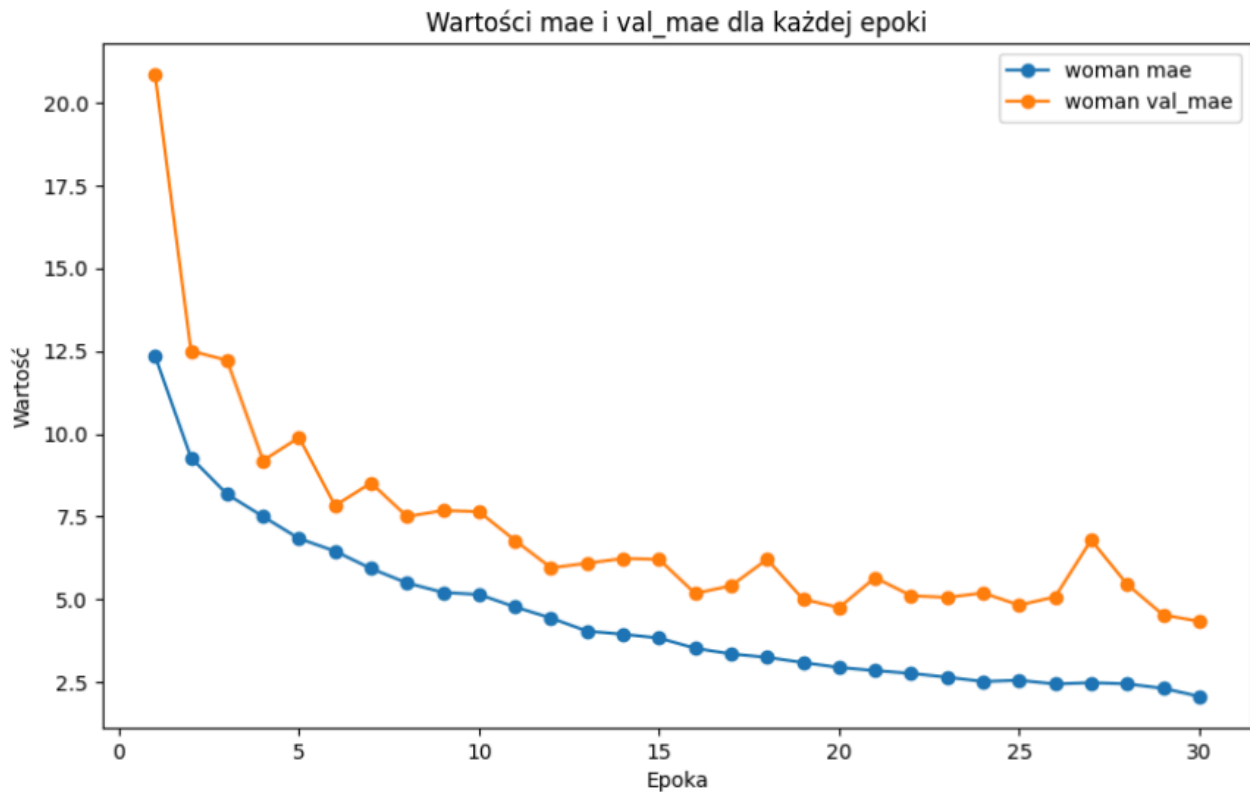
Wykres treningowy UTK_faces_man



Model był ćwiczony na 30 epokach, batch_size=32, shuffle=True. Dane podzieliłiśmy w proporcjach 80/20 (80% procent dane treningowe, 20% dane testowe). Model był ćwiczony na data set UTK-faces. Aby użyć funkcji stratify przy oddzielaniu danych, wiek osób starszych niż 85 lat zmieniłem na 85. W testach jednak uwzględniłem prawdziwy wiek osób.

```
636 - mse: 18.4722 - accuracy: 0.0649 - val_loss: 5.4721 - va
405 - mse: 17.0802 - accuracy: 0.0673 - val_loss: 5.8403 - va
7132 - mse: 15.7576 - accuracy: 0.0659 - val_loss: 5.4328 - v
623 - mse: 15.1160 - accuracy: 0.0631 - val_loss: 5.7042 - va
562 - mse: 13.8357 - accuracy: 0.0653 - val_loss: 5.4869 - va
210 - mse: 13.4204 - accuracy: 0.0668 - val_loss: 5.8082 - va
498 - mse: 11.7539 - accuracy: 0.0662 - val_loss: 5.6307 - va
```

Wykres treningowy UTK_faces_woman



Tutaj tak samo model był ćwiczony na 30 epokach, batch_size=32, shuffle=True. Dane podzieliśmy w proporcjach 80/20 (80% procent dane treningowe, 20% dane testowe). Aby użyć funkcji stratify przy oddzielaniu danych, wiek osób starszych niż 85 lat zmieniłem na 85. W testach jednak uwzględniłem prawdziwy wiek osób.

```
val_loss: 4.8258 - val_mae: 4.8258 - val_mse: 58.9648 - val_accuracy: 0.0560
val_loss: 5.0762 - val_mae: 5.0762 - val_mse: 66.3827 - val_accuracy: 0.0569
val_loss: 6.7911 - val_mae: 6.7911 - val_mse: 112.1016 - val_accuracy: 0.0569
val_loss: 5.4549 - val_mae: 5.4549 - val_mse: 81.1499 - val_accuracy: 0.0186
- val_loss: 4.5295 - val_mae: 4.5295 - val_mse: 52.5187 - val_accuracy: 0.0551
- val_loss: 4.3369 - val_mae: 4.3369 - val_mse: 52.9462 - val_accuracy: 0.0347
```

Testowanie modelu

```
Check_MAE(model_man, test_ages_man, test_images_man)
Check_MAE(model_woman, test_ages_woman, test_images_woman)
```

```
105/105 [=====] - 1s 13ms/step
105/105 [=====] - 1s 12ms/step
MAE: 3.411360709385263
MSE: 31.224657453643275
105/105 [=====] - 1s 13ms/step
105/105 [=====] - 1s 13ms/step
MAE: 2.625916027926428
MSE: 20.247629053703626
```

Model przy tej samej architekturze, tym samym sposobie ćwiczenia, oparty na tym samym data-secie otrzymały różne wyniki dla danych testowych. Model dla kobiet okazał się mieć większą skuteczność i to w stopniu znaczącym (o 0.786 w statystyce MAE). Jest to możliwe spowodowane innym sposobem starzenia się kobiet, niż w przypadku mężczyzn.

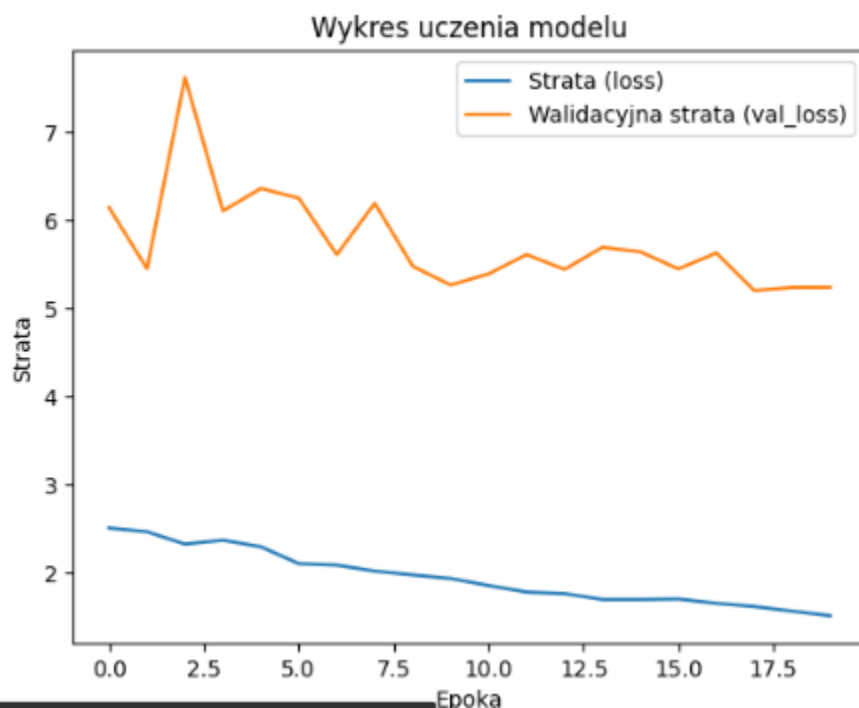
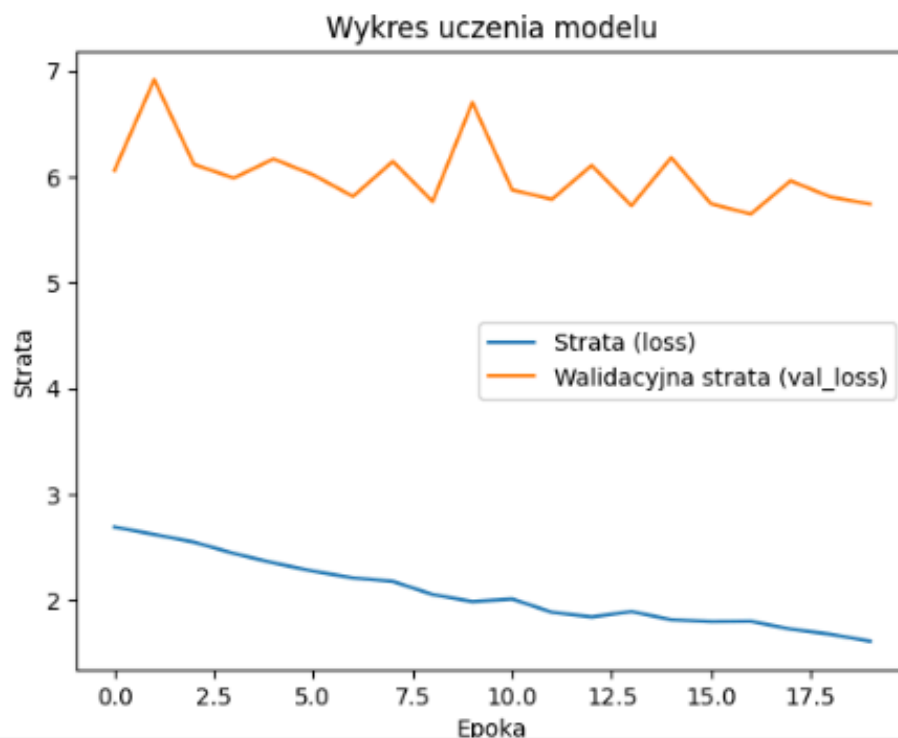
Wyniki te są nieaktualne, niefortunnie przy pobieraniu danych zostawiłem metodę `np.shuffle` o której zapomniałem. Po wytrenowaniu modelu odłączyło mi się GPU na google colab i musiałem pobrać znowu dane które się przetasowały. Więc wyżej wymienione wyniki są niepoprawne. Jednak modele osiągnęły podczas trenowania `val_mae = 4.33` dla kobiet i `val_mae = 5.43` dla mężczyzn.

Model detekcji płci

Do połączenia wyżej wymienionych płci posłużyliśmy się dwoma modelami. Jednym modelem jest gotowy model z biblioteki python deepface, posłużyliśmy się nią w GUI, model osiąga wynik 97.44% accuracy. Drugim modelem użytym do łączenia, szczególnie do testowania był własnoręcznie stworzony model oparty na data secie UTK faces. Architektura to również przerobiony Resnet50, ćwiczony na 30 epochs, batch_size=8. Model osiągnąć wynik 0.9427% accuracy.

```
Epoch 24/30
1670/1670 [=====] - 107s 64ms/step - loss: 0.0430 - accuracy: 0.9856 - val_loss: 0.2219 - val_accuracy: 0.9427
Epoch 25/30
1670/1670 [=====] - 108s 64ms/step - loss: 0.0399 - accuracy: 0.9857 - val_loss: 0.2596 - val_accuracy: 0.9311
Epoch 26/30
1670/1670 [=====] - 107s 64ms/step - loss: 0.0364 - accuracy: 0.9871 - val_loss: 0.2426 - val_accuracy: 0.9361
Epoch 27/30
1670/1670 [=====] - 109s 66ms/step - loss: 0.0323 - accuracy: 0.9891 - val_loss: 0.2740 - val_accuracy: 0.9373
Epoch 28/30
1670/1670 [=====] - 84s 50ms/step - loss: 0.0350 - accuracy: 0.9877 - val_loss: 0.2544 - val_accuracy: 0.9410
Epoch 29/30
1670/1670 [=====] - 84s 50ms/step - loss: 0.0314 - accuracy: 0.9887 - val_loss: 0.2468 - val_accuracy: 0.9401
Epoch 30/30
1670/1670 [=====] - 84s 51ms/step - loss: 0.0310 - accuracy: 0.9884 - val_loss: 0.2969 - val accuracy: 0.9359
```

Ponowne trenowanie modeli dla kobiet i mężczyzn wraz z końcową metryką



Modele były trenowane 2x20 epok z batch_size ustawionym na 32. Pierwszy Wykres przedstawia model dla mężczyzn, a drugi dla kobiet. Niestety nie udało się odtworzyć poprzednich udokumentowanych wyników (4,3 i 5,3, kolejno dla kobiet i mężczyzn), a ostateczne wyniki to 5,2 i 5,6 w tej samej kolejności, oba modele to resnet50 z wagami

imagenet z funkcją kosztu mae, i dla odmiany w stosunku do poprzednich używany był cały dataset UTKFace który ma około 20tys zdjęć tak aby była odpowiednia ilość zdjęć dla obu płci. Na lepsze wyniki ma szansę zdecydowanie model dla kobiet, który zapewne przy większej ilości prób i skorzystaniu z grid searcha osiągnąłby poziom poniżej 4 błędu mae. Wydaje się, że może być kilka powodów nie odtworzenia wyników:

1. Korzystanie z całego UTK Face, a nie fragmentów datasetu
2. Korzystanie z wag Imagenet
3. W poprzednich modelach używany był croppart 2 który mógł poprawiać wynik, gdyż ten nowszy choć zawiera więcej zdjęć to nie wszystkie są w nich scropowane

Poprzednich modeli nie możemy przepuścić gdyż po generowaniu setu przez `train_test_split` z seedem 42 były celowo shufflowane w kodzie i błąd nie byłoby reprezentatywny. Oba modele wytrenowane później były osiągane na tym samym datasetcie dzielonym na seedzie 42, więc wynik można odtworzyć.

Co ciekawe ostateczny MAE całego modelu, z wykorzystaniem sieci do rozpoznawania płci wynosi 5.01.

```
[[5.014361]]
```

Być może wynika to z tego że oba te modele były trenowane z użyciem grup wiekowych 105-110 itd i w wyniku pośpiechu nie było to zmieniane w testowym zbiorze, a do metryki MAE finalnej użyte były oryginalne etykiety.

```

n = len(test_images)
for img, age in zip(test_images, test_labels):
    imgr64 = cv2.resize(img, (50, 50))
    imgr64=np.expand_dims(np.array(Image.fromarray(imgr64).resize((50, 50))), axis=0)
    img=np.expand_dims(np.array(Image.fromarray(img).resize((224, 224))), axis=0)
    g = model3.predict(imgr64)
    if(g==0):
        age_predict= model1.predict(img)

    else:
        age_predict= model2.predict(img)
    print(age_predict[0][0])
    print(age)
    abs = np.abs(age-age_predict[0][0])
    sum=sum+abs
mae = sum/n

print(mae)

```

Oto kod który oblicza finalną metrykę i symuluje pełny model z użyciem modelu do detekcji płci.(Nie udało się utworzyć finalnego modelu z użyciem concatenate). Można więc oczekiwać że z użyciem poprzednich modeli generowany by był finalny błąd rzędu 4.5.

Wnioski

Wydaje się że podejście z podziałem na płeć ma sensowne zastosowanie, widoczna jest różnica między dwoma modelami, jeden z nich osiąga istotnie lepsze wyniki na tej samej architekturze modelu(w nowych o około .5 , w starych modelach o 1), żeby uczciwie porównać istotę tego podejścia należy go porównać z analogicznie trenowanym modelem na całym datasetcie. Być może sensownym podejściem w celu poprawienia wyników byłoby tworzenie różnych architektur dla obu płci. Warto zauważyć, że modele do detekcji płci osiągają dość duże accuracy i to właśnie to może w dużej mierze decydować o jakości tego podejścia, tzn czy poprawa uzyskana z lepszego modelu dla kobiet nie jest zniwelowana przez błąd sieci do detekcji płci i skorzystanie z niepoprawnego modelu.