

# Biometria 1 - Rozpoznawanie twarzy

Kamil Chmurzyński (276939, gr 2, śr 13:15)

Paweł Kotzbach (254540, gr 2, śr 13:15)

Mikołaj Langner (255716, gr 2, śr 13:15)

Kamil Matejuk (250135, gr 2, śr 13:15)

3.04.2024

## 1 Metoda Autoryzacji

Wykorzystując pretrenowany model do rozpoznawania twarzy, wykorzystana zostanie reprezentacja z przedostatniej warstwy modelu (wejścia dla warstwy klasyfikującej w modelu oryginalnym). Dla każdego zdjęcia zapisanego w systemie, zostanie zapisana jego reprezentacja wektorowa w bazie danych.

Następnie autentykacja polegać będzie na znalezieniu wektorów w bazie danych, o odległości od reprezentacji zadanej zdjęcia, poniżej zadanej granicy. Jeżeli podpis podany podczas autentykacji znajduje się pośród podpisów zwróconych wektorów, autentykacja przechodzi poprawnie.

## 2 Dane

Wykorzystano zbiór CelebA, zawierający 202599 kolorowych zdjęć 10177 osób. Po usunięciu osób, do których przypisane było poniżej 10 zdjęć, w zbiorze zostało 193569 zdjęć 8369 osób.



Rysunek 1: Przykład zbioru CelebA

Zdjęcia były różnych rozmiarów, zatem zostały przeskalowane do 256 x 256 pikseli, bez zmiany proporcji, wypełniając pustą przestrzeń kolorem czarnym.

## 2.1 Autorskie dane

Dodatkowo do zbioru danych zostały włączone zdjęcia uczestników projektu.



Rysunek 2: Przykład zbioru dodanych zdjęć

## 2.2 Podział

Na początku ze zbiorami został wydzielony zbiór tzw. autentykacyjny, który nie zostanie użyty podczas finetuningu, natomiast zostanie wykorzystany podczas ćwiczeń z autentykacją. Następnie pozostałe dana zostały podzielone ze stratyfikacją na zbiór treningowy, testowy i walidacyjny - odpowiednio 70%, 15% i 15%. Poniższa tabela przedstawia statystyki zbiorów.

Zbiór	Ilość zdjęć	Ilość osób	Średnia ilość zdjęć na osobę
Treningowy	133772	8269	16.18
Walidacyjny	28666	8269	3.47
Testowy	28666	8269	3.47
Autentykacyjny	2125	100	21.25

Rysunek 3: Statystyki zbiorów

W ramach przeszukiwania przestrzeni hiperparametrów uzyto zbiorów Treningowego, Walidacyjnego i Testowego, zmniejszonych do 100 osób, zachowując proporcje ilości zdjęć na osobę, ze względu na czas obliczeń.

## 3 Model

Wykorzystano model GhostFaceV2 wraz z autorską nakładką na potrzeby testów.

Jako funkcję starty wykorzystano CrossEntropyLoss, z optymalizatorem SGD (Stochastic Gradient Descent).

### 3.1 Modyfikacje

#### 3.1.1 Funkcja aktywacji

Na potrzeby fine-tuningu, dodano warstwę softmax na output klasyfikatora.

#### 3.1.2 Preprocessing danych

Na każdym zbiorze w ostatnim etapie wykonano normalizację wartości w każdym kanale zgodnie z danymi wyliczonymi na wielomilionowym zbiorze ImageNet.

Kanał	Średnia	Odchylenie standardowe
Czerwony (R)	0.4850	0.229
Zielony (G)	0.4560	0.224
Niebieski (B)	0.4060	0.225

Rysunek 4: Normalizacja kanałów kolorów

Dodatkowo wstępne przetwarzanie danych zostało wykonane różne dla zbioru Treningowego, oraz dla zbiorów Walidacyjnego, Testowego i Autentykacyjnego.

Na zbiorze Treningowym wykonano:

- *RandomResizedCrop* - przycięcie do losowego fragmentu o rozmiarze 128 x 128 pikseli
- *RandomHorizontalFlip* - odbicie w poziomie z prawdopodobieństwem 50%
- *ColorJitter* - losowa modyfikacja jasności, kontrastu i nasycenia, z zakresu od 0.6 do 1.4 dla każdego z parametrów



Rysunek 5: Transformacja treningowa

Na zbiorach Walidacyjnym, Testowym i Autentykacyjnym wykonano:

- *Resize* - przeskalowanie do rozmiaru 128 x 128 pikseli



Rysunek 6: Transformacja Walidacyjna, Testowa i Autentykacyjna

### 3.1.3 Przeszukiwanie hiperparametrów

Podczas optymalizacji wzięto pod uwagę następujące parametry:

- width - model dostępny jest w trzech wariantach (1.0, 1.3, oraz 1.6), opisujących szerokość modelu, tj. ilość neuronów w warstwach ukrytych
- dropout - z zakresu 0%, 25%, 50%
- batch size - z zakresu 16, 32, 64
- learning rate - z zakresu 1e-1, 1e-2, 1e-3, 1e-4
- momentum - parametr optymalizatora SGD, z zakresu 0, 0.9

Model pretrenowany udostępniony został z parametrami domyślnymi: dropout 0%, batch size 32, learning rate 1e-2, momentum 0.9.

Autorom modelu pretrenowanego udało się uzyskać wynik rzędu 75% accuracy na swoim datasecie.

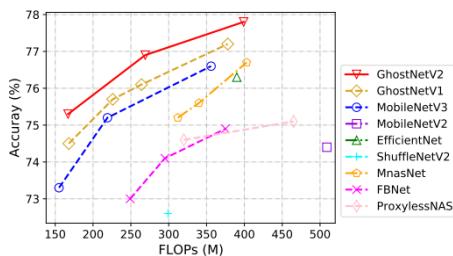


Figure 1: Top-1 accuracy vs. FLOPs on ImageNet dataset.

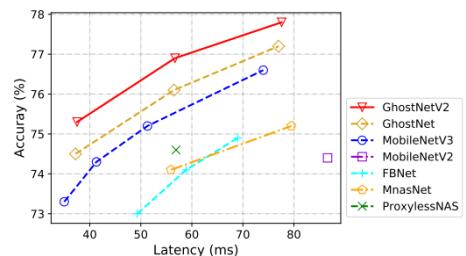
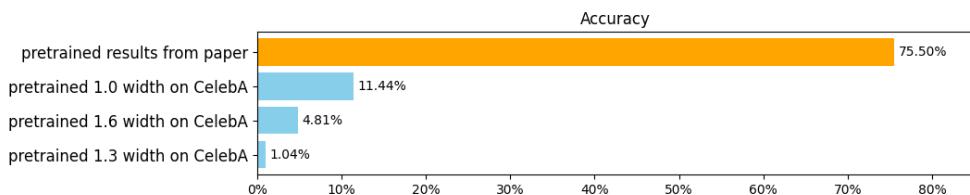


Figure 2: Top-1 accuracy vs. latency on ImageNet dataset.

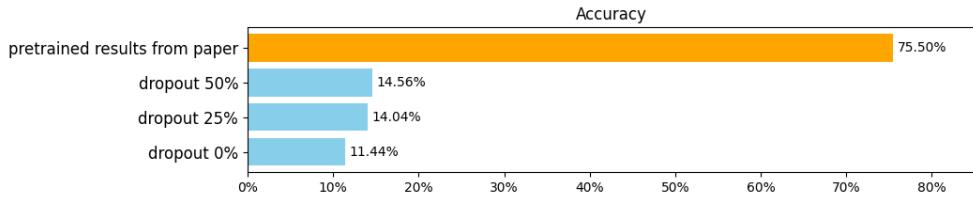
Rysunek 7: Y. Tang et.al "GhostNetV2: Enhance Cheap Operation with Long-Range Attention"

Natomiast porównując wyniki otrzymane dla tych samych hiperparametrów, na zbiorze CelebA, widać potrzebę znalezienia lepszych wartości. Poniżej przedstawiono porównanie różnych zestawów hiperparametrów po finetuningu na zbiorze CelebA, wraz z baseline - wynikiem autorów na ich zbiorze ImageNet.



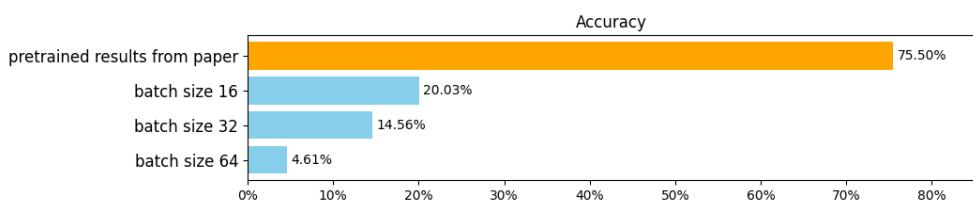
Rysunek 8: Analiza hiperparametru width

Dalsze testy przeprowadzone zostaną na modelu o szerokości 1.0.



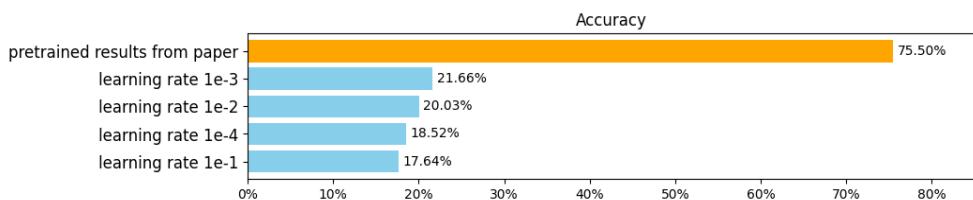
Rysunek 9: Analiza hiperparametru dropout

Najlepszy wynik uzyskano przy 50% dropout.



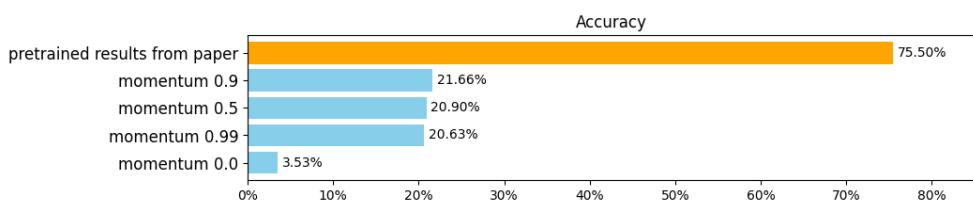
Rysunek 10: Analiza hiperparametru batch size

Najlepszy wynik uzyskano dla partii po 16 próbek.



Rysunek 11: Analiza hiperparametru learning rate

Najlepszy wynik uzyskano dla kroku uczenia 0.001.

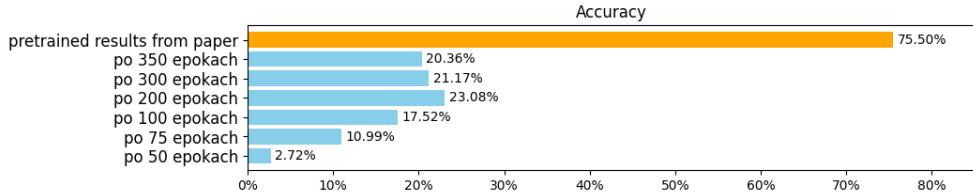


Rysunek 12: Analiza hiperparametru momentum

Rzeczywiście momentum dla optymalizatora SGD równe 0.9 zwraca najlepsze wyniki, tak jak autorzy modelu zastosowali.

W poprzednich etapach zatrzymywano uczenie na 100 epokach, bądź po 10 epokach

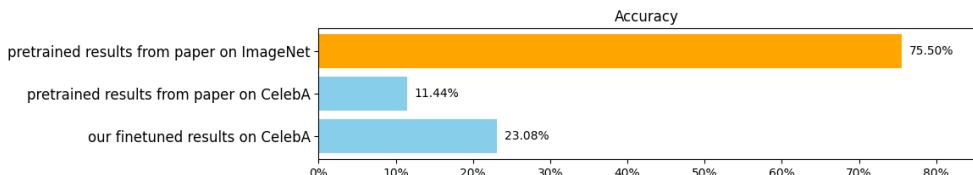
bez poprawy w wartości funkcji straty na zbiorze walidacyjnym (*early stop*). Dla znalezionych najlepszych wartości hiperparametrów, kontynuowano uczenie przez kolejne 350 epok. Najlepsze wyniki osiągnięto po 200 epokach.



Rysunek 13: Analiza ilości epok

### 3.2 Wynik

Końcowo model douczono na pełnym datasetie treningowym. Ze względu na czas i moc obliczeniową zatrzymano trening po 23 epochach, nie osiągając lepszego wyniku. Finalnie najlepsze accuracy otrzymywano dla hiperparametrów dropout 50%, batch size 16, learning rate 1e-3, momentum 0.9.



Rysunek 14: Porównanie wyników finetuningu

Warto zauważyć, że klasyfikacja 100-klasowa nie jest łatwym zadaniem, dlatego nawet accuracy rzędu 23% jest dobrym wynikiem, zwłaszcza biorąc pod uwagę że będziemy wykorzystywać tylko wyuczone reprezentacje z ostatniej warstwy modelu, nie klasyfikację.

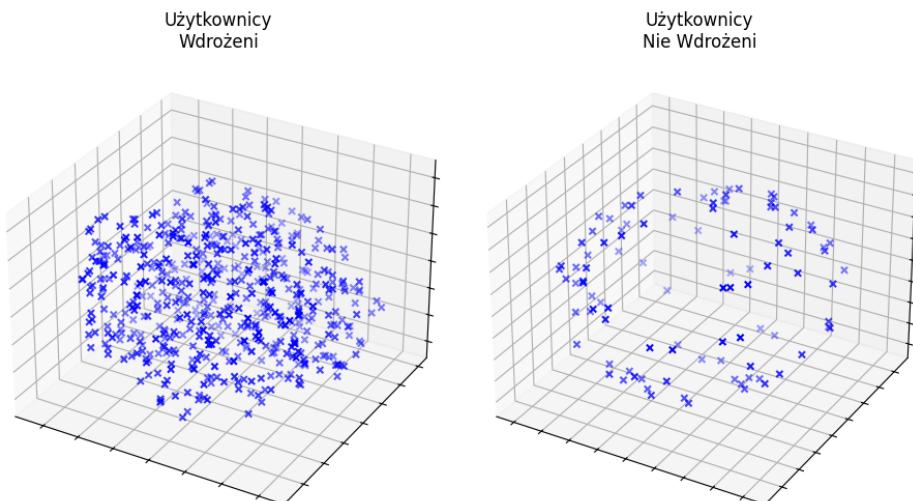
## 4 Testy

### 4.1 Baza danych

W poniższych testach zastosowano zbiór Autentykacyjny użytkowników wdrożonych do bazy (dalej nazywany  $UW$ ) do zainicjalizowania bazy danych. Następnie testowano na innym zbiorze zdjęć użytkowników zdrożonych do bazy ( $UW^*$ ) oraz zbiorze użytkowników nie zdrożonych do bazy ( $UNW$ ).

Zbiór	Ilość zdjęć	Ilość osób	Średnia ilość zdjęć na osobę	Ilość prób podsztycia się
$UW$	1465	80	18.3	0%
$UW^*$	560	80	7.0	50%
$UNW$	100	20	5.0	50%

Rysunek 15: Statystyki zbiorów Autentykacyjnych



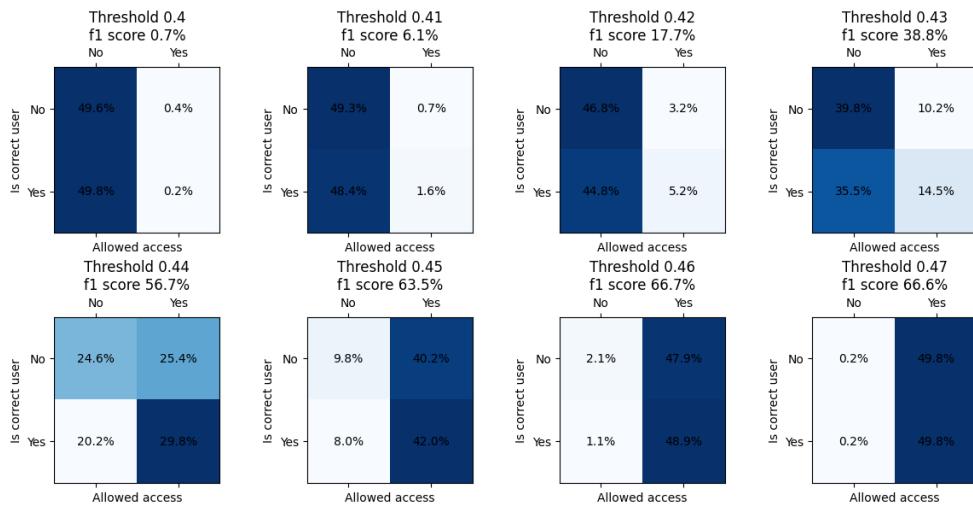
Rysunek 16: Transformacja 3D UMAP Reprezentacji Wektorowej zbiorów

### 4.2 Skuteczność systemu

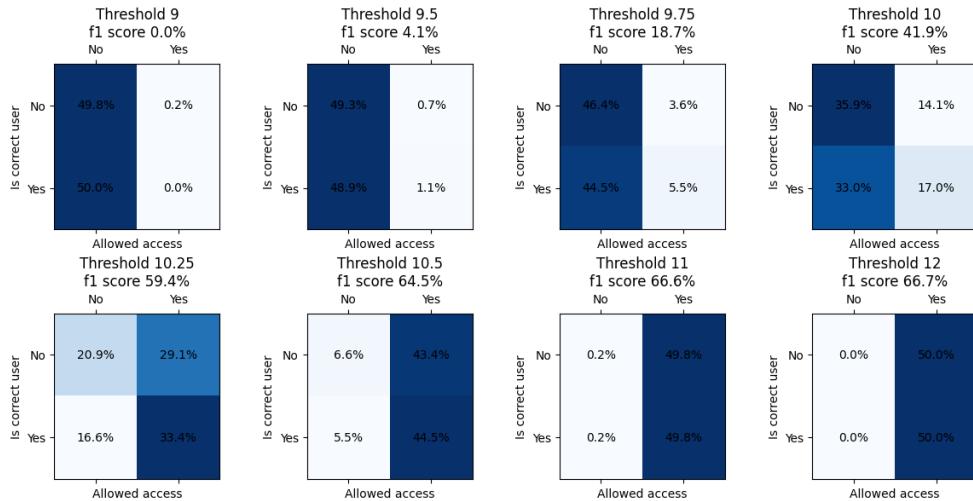
Wykorzystany zostanie tylko zbiór  $UW^*$ , zaiwierający 500 zdjęć, z czego 50% będzie próbami podsztycia się.

#### 4.2.1 Dobór granicy odległości

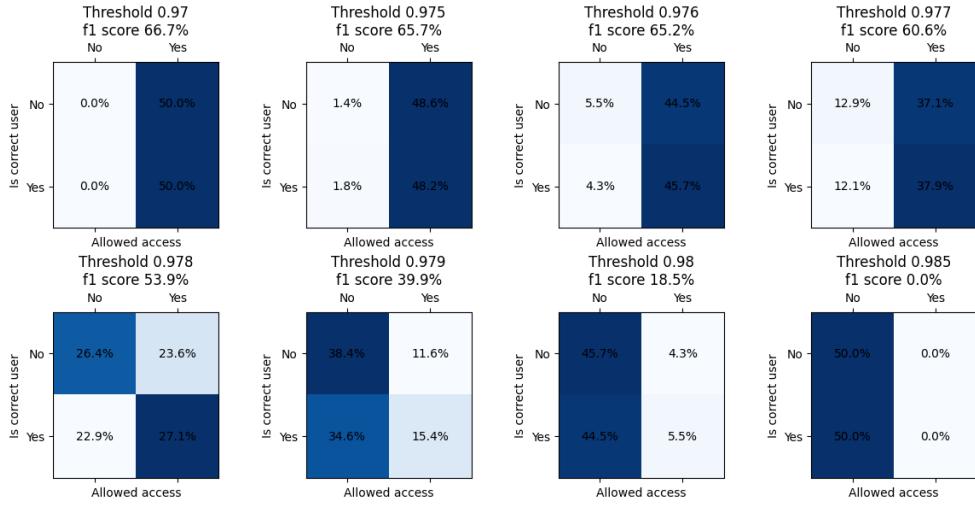
Autentykacja polega na znalezieniu wektorów w bazie danych, o odległości od reprezentacji zadanego zdjęcia, poniżej zadanej granicy. Jeżeli podpis podany podczas autentykacji znajduje się pośród podpisów zwróconych wektorów, autentykacja przechodzi poprawnie. Przeanalizowane zostanie kilka różnych wartości granic, oraz różne metryki odległości.



Rysunek 17: Porównanie wyników systemu dla różnych granic odległości euklidesowej



Rysunek 18: Porównanie wyników systemu dla różnych granic odległości manhattańskiej



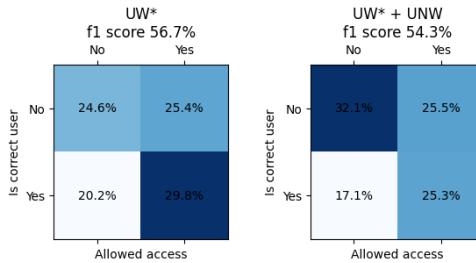
Rysunek 19: Porównanie wyników systemu dla różnych granic odległości cosinusowej

Jak widać niezależnie od metryki odległości, model nie potrafił wygenerować reprezentacji na tyle unikalnych, by stworzyć bezpieczny system, nie przepuszczający oszustów. Maksymalnie osiągnięto f1-score na poziomie 60%, jedynie można wybrać czy preferowane jest nieogłoszenie zarejestrowanych użytkowników, czy logowanie niezarejestrowanych użytkowników.

W dalszych testach zostanie wukorzystana odległość euklidesowa, z granicą 0.44.

### 4.3 Skuteczność systemu z niedodanymi użytkownikami

Do zbioru  $UW^*$ , zaiwierającego 500 zdjęć, zostanie dodany zbiór  $UNW$ , zawierający 100 zdjęć osób nie będących w bazie danych. Nadal 50% próbek będzie próbami podszycia się.

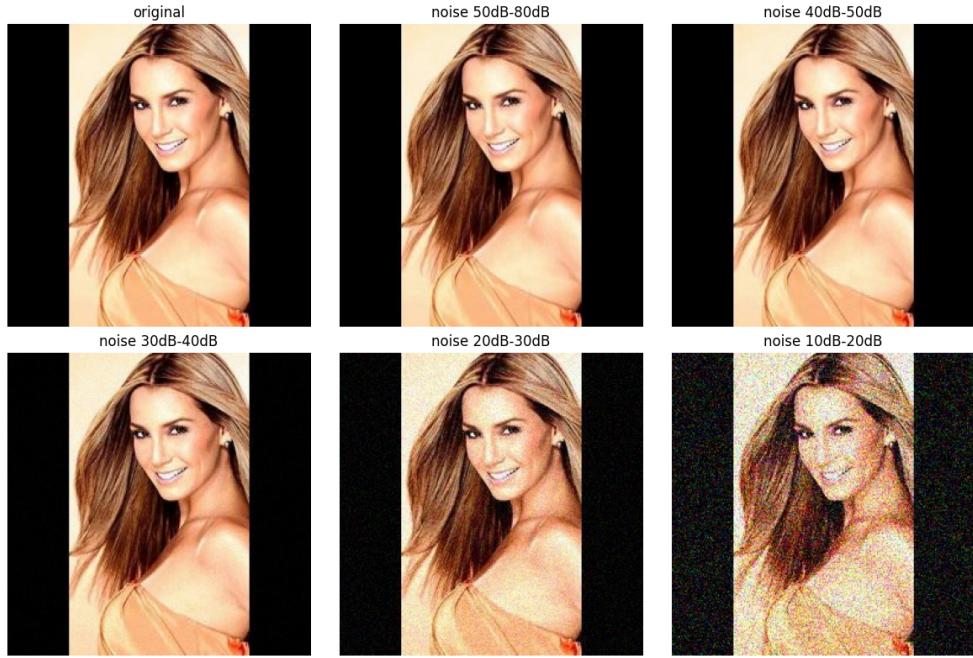


Rysunek 20: Porównanie wyników systemu po dodaniu zbioru UNW

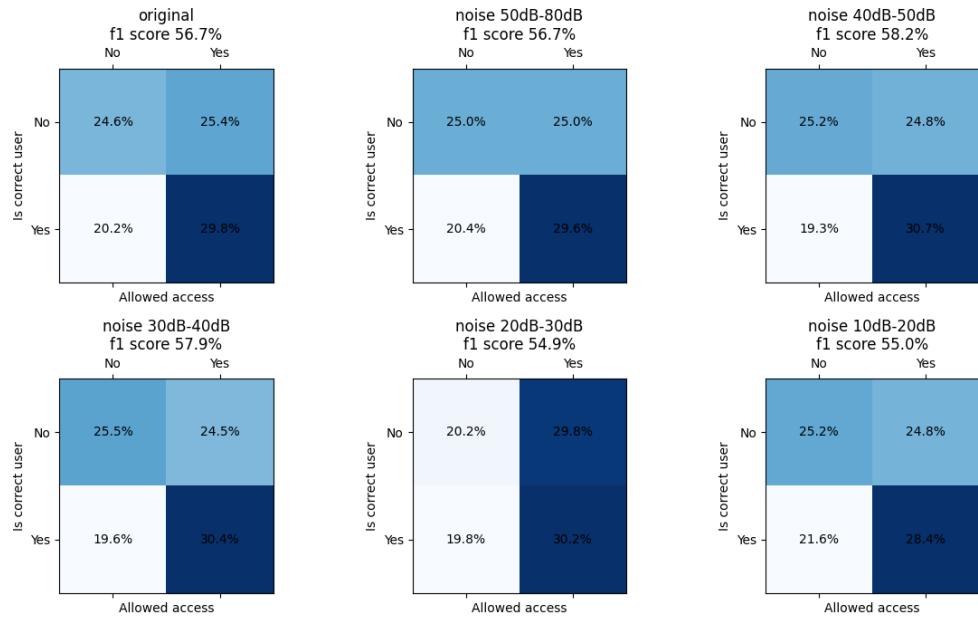
System poprawnie nie udzielił dostępu dla większej ilości próbek (32.1% vs 24.6%).

### 4.4 Odporność systemu na szum

Poniżej przedstawiono przykładowy efekt zastosowania szumu Gausowskiego w zadanej skali PSNR na zdjęcie.



Rysunek 21: Efekt dodania szumu do zdjęcia w zadanych grancach



Rysunek 22: Porównanie wyników systemu po dodaniu szumu

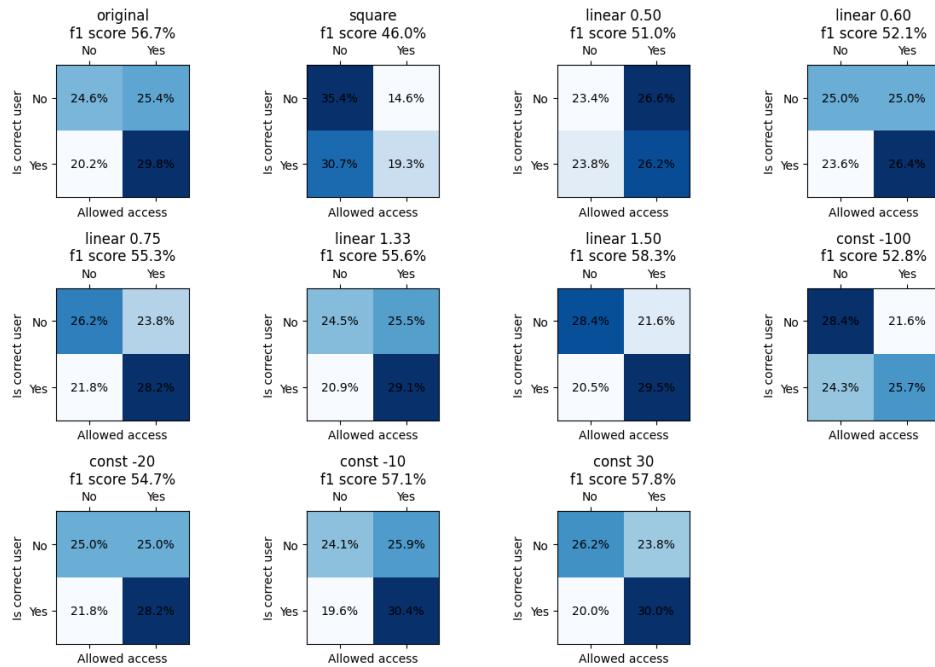
Dodanie szumu nie wpłynęło znacząco na wyniki. Dopiero znaczny szum ( $< 20dB$ ) spowodował delikatne zmniejszenie się f1-score.

## 4.5 Odporność systemu na poziom luminacji

Poniżej przedstawiono przykładowy efekt edycji luminacji na zdjęcie. Każde ze zdjęć zostało przekonwertowane do *HSL*, następnie zedytowany ostał ostatni kanał *L*, a zdjęcie przekonwertowane spowrotem do *RGB*.



Rysunek 23: Efekt edycji luminacji zdjęcia w sposób kwadratowy, liniowy, lub o stałą



Rysunek 24: Porównanie wyników systemu po edycji luminacji

Edycja luminacji w większym stopniu wywarła wpływ na wyniki, ponieważ w większym stopniu zmieniła zdjęcie wejściowe.

Zmiana kwadratowa znacznie pogorszyła jakość zdjęcia, co widać w największym spadku f1-score.

Pomiędzy liniową zmianą luminacji a wynikami systemi widać zależność liniową - im większa luminacja tym lepiej radzi sobie system.

Zmiana o stałą pogorszyła wynik dla ujemnej stałej (zmniejszając luminację), oraz nie zmieniła wyniku przy stałej dodatniej (zwiększenie luminacji), prawdopodobnie poprzez wprowadzone artefakty.

## 5 Podsumowanie

Autorzy udostępnili wytrenowany model osiągający accuracy 75% na datasecie ImageNet. Na datasecie CelebA jednak osiągnął on jedynie 11% accuracy.

W ramach projektu pretrenowany model douczono na biorze CelebA, osiągając accuracy 23%. Niestety wynik ten nie przełożył się na tworzenie odpowiednich reprezentacji na ostatniej wartwie ukrytej do zadania autentykacji na podstawie twarzy.

Osiągnięto efekt, w którym model prawidłowo działa w 54.4% przypadkach (29.8% prawidłowo zalogował, 24.6% prawidłowo odrzucił). Dodatkowo w 25.4% przypadków model zalogował użytkownika podszywającego się pod kogoś innego, oraz w 20.2% przypadków odmówił dostępu użytkownikowi posiadającemu konto.

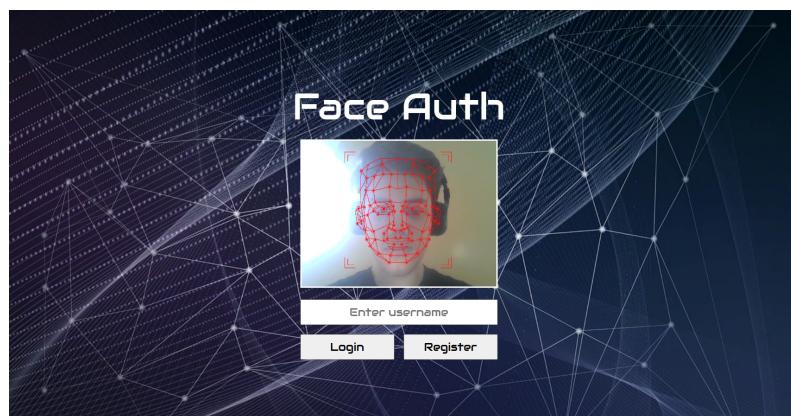
## 6 Dodatki

### 6.1 Kod źródłowy

Kod źródłowy można znaleźć w serwisie [github.com](https://github.com). Wstępnie architektura została przystosowana do przetestowania każdego z zaproponowanych modeli (ArcFace, DeepFace, GhostFace, InsightFace), natomiast finalnie ograniczono analizę tylko do modelu GhostNet ze względu na dostępność pretrenowanych wag modelu.

### 6.2 Aplikacja

Do interakcji z modelem, oraz aby zaprezentować użycie w realnym środowisku, utworzono stronę internetową do logowania za pomocą rozpoznawania twarzy.



Rysunek 25: Interface strony internetowej