UNIWERSYTET IM. ADAMA MICKIEWICZA W POZNANIU
WYDZIAŁ NAUK SPOŁECZNYCH
KOGNITYWISTYKA

Gracjan Popiółkowski

Algorytmy rozpoznające wiek ludzi na podstawie zdjęcia twarzy

Algorithms for Recognizing People's Age from Facial Photos

Poznań 2023

Spis treści

Wprowadzenie

W niniejszym raporcie przedstawiam porównanie wyników klasyfikacji zdjęć twarzy do czterech grup wiekowych: 6 - 20 lat, 25 - 30 lat, 42 - 48 lat i 60 - 98 lat. Dokonam analizy różnych modeli o zróżnicowanych parametrach pod kątem skuteczności oraz problemu przeuczenia. Dodatkowo, zweryfikuję osiągi moich modeli w porównaniu do jednego z popularnych modeli, tj. VGG19.

ROZDZIAŁ 1

Prezentacja danych

1.1. Zbiór danych

Posługiwałem się zbiorem danych "Age Recognition Dataset" udostępnionym przez użytkownika Rashik Rahmana na platformie Kaggle. Zbiór obejmuje ponad 9 tysięcy zdjęć, sklasyfikowanych według 4 grup wiekowych. Zdjęcia to wycinki twarzy o różnych cechach etnicznych, nie pozbawione tła, o rozmiarze 200x200 pikseli.

1.2. Przygotowanie danych

Wszystkie obrazy zostały przeskalowane do rozmiaru 128x128 pikseli. Skorzystałem z generatora danych 'ImageDataGenerator' z biblioteki Keras do dynamicznego generowania rozszerzonych danych treningowych podczas procesu uczenia. Generator został skonfigurowany w taki sposób, aby:

- Skalował wartości pikseli z 0-255 do 0-1.
- Losowo obracał obrazy w poziomie i pionie.
- Rotował obrazy w zakresie -20 do 20 stopni.
- Zarezerwował 20% danych na potrzeby walidacji.

Wszystko to miało na celu zwiększenie różnorodności danych treningowych.

ROZDZIAŁ 2

Algorytmy klasyfikacji obrazów

Algorytmy klasyfikacji obrazów to rodzaj modeli uczenia maszynowego, które uczą się rozpoznawać i przypisywać etykiety do obrazów na podstawie wzorców i cech obecnych na tych obrazach. Algorytmy klasyfikacji obrazów są powszechnie stosowane w różnych dziedzinach, takich jak rozpoznawanie obiektów, diagnostyka medyczna, analiza obrazów satelitarnych i wiele innych. Popularnym narzędziem w tej dziedzinie są konwolucyjne sieci neuronowe (CNN) ze względu na ich zdolność do skutecznego wykrywania wzorców w przestrzeni wielowymiarowej. W mojej analizie skupiłem się na modyfikacji trzech kluczowych parametrów istotnych dla klasyfikacji obrazów:

1. `Liczba filtrów w warstwach konwolucyjnych` - wpływa na wykrywanie bardziej skomplikowanych cech w obrazach.
2. `Rozmiar warstw gęsto połączonych` - zwiększenie liczby neuronów w warstwach gęsto połączonych może pomóc modelowi lepiej dopasować się do danych treningowych.
3. `Dropout` - to technika regularyzacji, która losowo "wyłącza" pewne neurony w trakcie treningu, co może pomóc w zapobieganiu przetrenowaniu modelu.

`VGG19` - to jeden z modeli CNN zaprojektowanych do klasyfikacji obrazów. Nazwa "VGG19" odnosi się do VGGNet, który jest skrótem od Visual Geometry Group Net, a liczba "19" wskazuje na liczbę warstw sieci, w tym 16 warstw konwolucyjnych i 3 warstwy w pełni połączone. Model VGG19 został pierwotnie wytrenowany na dużym zbiorze danych ImageNet, który obejmuje miliony obrazów z tysiącami kategorii.

W dalszej części tekstu przedstawię porównanie wyników klasyfikacji po zmianie wymienionych parametrów oraz porównam te wyniki do rezultatów VGG19.

ROZDZIAŁ 3

Analiza wyników

3.1. Najprostszy model sieci

`Model 1` - to będzie najprostszy z moich modeli sieci, posłuży nam on jako punkt startowy. Jego specyfikacja:

- `Konwolucyjna Warstwa Conv2D` - w tej warstwie zastosowałem 16 filtrów o rozmiarze 3x3, aktywowane funkcją ReLU. Jest to kluczowy komponent konwolucyjnych sieci neuronowych, odpowiedzialny za ekstrakcję cech z obrazu.
- `Warstwa MaxPooling2D` - warstwa ta wykorzystuje operację max pooling na wynikach poprzedniej konwolucji. Redukuje wymiary macierzy, utrzymując jednocześnie kluczowe cechy obrazu.
- `BatchNormalization` - jest zastosowany po każdej warstwie konwolucyjnej i po warstwie Dense. Pomaga w przyspieszeniu procesu uczenia poprzez normalizację aktywacji w warstwie. Zapewnia także stabilność procesu uczenia.
- `Warstwa Flatten` - jest używana do przekształcenia wielowymiarowej macierzy do jednowymiarowego wektora, co jest wymagane przed przejściem do warstw Dense.
- `Warstwy Dense` - model posiada trzy warstwy Dense. Pierwsza ma 128 neuronów, druga ma 32 neurony, a trzecia to warstwa wyjściowa z 4 neuronami (odpowiadającymi liczbie kategorii). Aktywacja w warstwach gęsto połączonych to ReLU, z wyjątkiem warstwy wyjściowej, gdzie stosuje się funkcję softmax dla problemu klasyfikacji.
- `Optymalizator Adam` - jest adaptacyjnym algorytmem optymalizacji gradientu stosowanym do minimalizacji funkcji straty. Adam jest popularny ze względu na jego efektywność i elastyczność.
- `Dropout` - model nie zawiera funkcji Dropout, więc może być narażony na przeuczenie.

Modele będę porównywał według wielkości `accuracy` i `val_accuracy`, czyli zmiennej oceniające skuteczność klasyfikacji na zbiorze treningowym i walidacyjnym (testowych). `Val_accuracy` mierzy, jak dobrze model generalizuje się na nowych, wcześniej niewidzianych danych. To ważne, aby sprawdzić, czy model nie jest nadmiernie dopasowany (overfitting), co może zdarzyć się, gdy dokładność treningowa jest wysoka, ale dokładność walidacyjna jest niższa. Wyniki w modelu startowym:

- `Accuracy` = 0.7070
- `Val_accuracy` = 0.6461

Czyli skuteczność klasyfikacji na nowych danych wynosi 64,61%. Co jest bardzo dobrym wynikiem przy 4 grupach klasyfikacyjnych, gdzie przyporządkowanie losowe oznaczało by 25%. Pomimo braku funkcji Dropout model nie wygląda również na mocno przetrenowany, pewnie z powodu małej liczby filtrów w warstwach konwolucynych i liczbie neuronów warstw Dense.

3.2. Porównanie 3 modeli ze zmianą jednego parametru

W drugiej części analizy przedstawię wyniki 3 modeli, w których zmieniłem po jednym kluczowym parametrze z modelu startowego.

3.2.1. Model 2 - większa liczby filtrów w Conv2D

W modelu 2 zdecydowałem się zwiększyć liczbę filtrów w warstwach konwolucyjnych z 16 do 32, co powinno przekładać się na wykrywanie bardziej skomplikowanych cech w obrazach, co może poprawić zdolność modelu do rozróżniania różnych grup wiekowych. Skuteczność tego modelu wyniosła:

- `Accuracy` = 0.6887
- `Val_accuracy` = 0.5234

Tutaj różnica między tymi dokładnościami już jest zdecydowanie duża, co świadczy o przeuczeniu modelu. Parametr `val_loss` też jest sporo większy niż w innych modelach i wynosi 1.1579. A dodatkowo skuteczność wcale nie jest lepsza od `Modelu 1`.

3.2.2. Model 3 - większa liczby neuronów w warstwach Dense

W przypadku modelu 3 zwiększyłem liczbę neuronów w warstwach Dense z odpowiednio 128, 32 i 4 na 256, 64 i 4, co również powinno skutkować lepszym dopasowaniem do danych treningowych, ale ryzykuję przeuczeniem. Oto wyniki:

- `Accuracy` = 0.7584
- `Val_accuracy` = 0.6153

Tutaj widzimy za to poprawę skuteczności na danych treningowych, jednak skuteczność na danych walidacyjnych jest znowu mniejsza. Parametr `val_loss` też jest stosunkowo duży, więc znowu zachodzi nam przeuczenie.

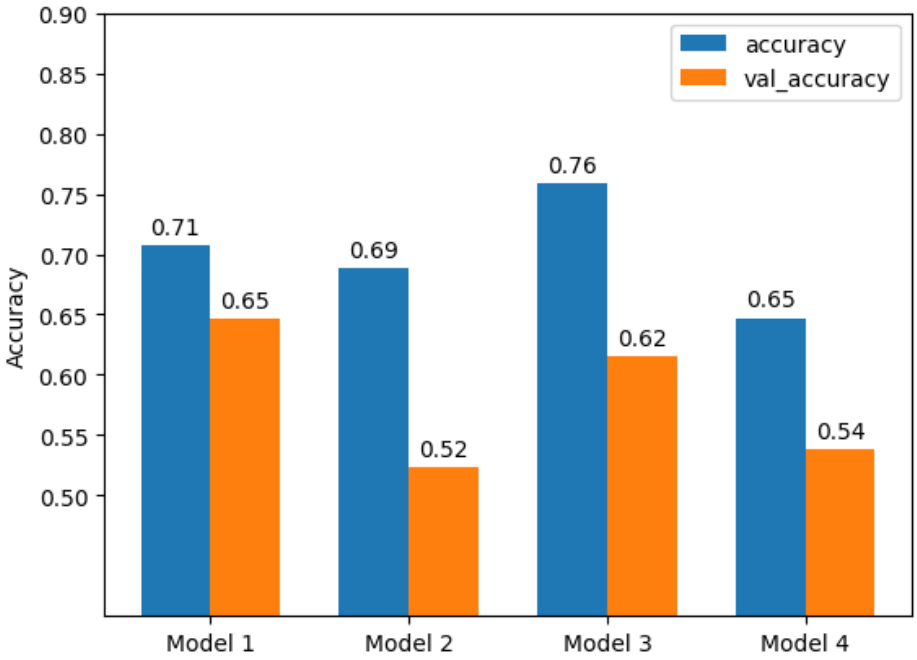
3.2.3. Model 4 - dodanie funkcji Dropout

Ostatecznie wprowadziłem funkcję Dropout (o wartości 0.3) już do modelu podstawowego, by sprawdzić czy już tutaj mogę polepszyć wyniki redukując przeuczenie.

- `Accuracy` = 0.6470
- `Val_accuracy` = 0.5382

Tu wyniki wyszły najsłabsze, więc z pewnością nie tędy droga. Dalej będę testował dodanie funkcji Dropout do bardziej skomplikowanych modeli.

3.2.4. Porównanie 4 pierwszych modeli



Rysunek 3.1: Wykres słupkowy przedstawia porównanie modelu podstawowego 1 i trzech modeli zmodyfikowanych. `Model 2` - większa liczba filtrów, `Model 3` - większa liczba neuronów warstw Dense, `Model 3` - dodana funkcja Dropout.

Ostatecznie widzimy, że `Model 1` wygląda najlepiej pod względem skuteczności na nowych danych. Dalej zobaczymy jak to będzie wyglądało, kiedy zmienimy więcej niż jeden parametr w danym modelu.

3.3. Dwie inne kombinacje i VGG19

W trzeciej części mojej analizy sprawdzę wyniki przy kombinacji zmian w naszych trzech głównych parametrach oraz przedstawię wyniki pretrenowanego modelu VGG19.

3.3.1. Model 5

Mimo iż modele 2 i 4 nie wypadły najlepiej, zdecydowałem się sprawdzić te wszystkie trzy zmiany w parametrach w jednym modelu, z nadzieją, że skuteczność zrównoważy ewentualne przeuczenie. `Model 5` posiada zwiększoną liczbę filtrów w warstwach konwolucyjnych do 32, liczbę neuronów warstwy Dense do 256, 64 i 4, oraz funkcję Dropout ustawioną na 0.3. Oto wyniki:

- `Accuracy` = 0.6652
- `Val_accuracy` = 0.5839

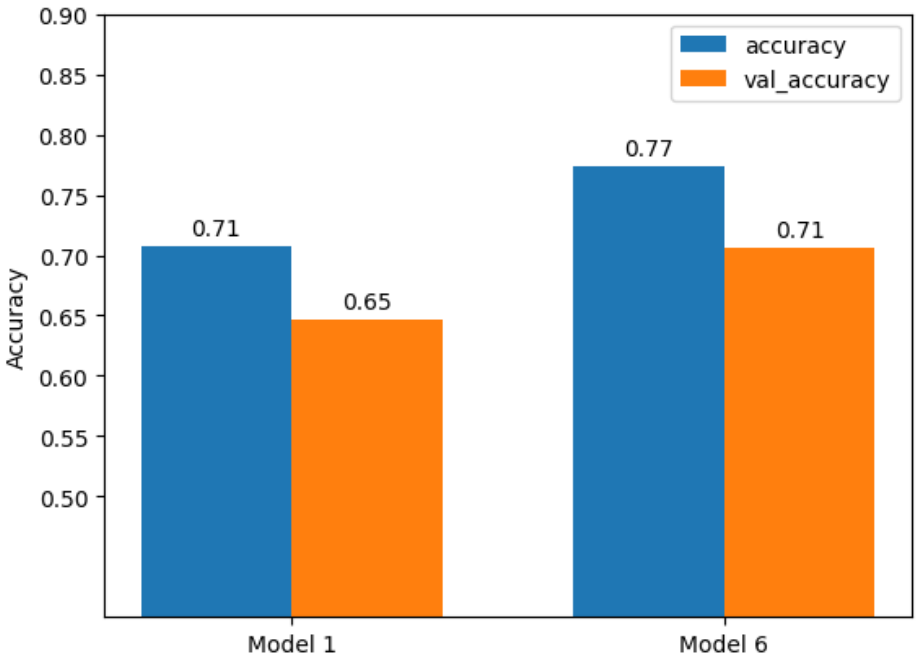
Przeuczenie jest raczej małe, ale jak widać skuteczność nadal jest sporo niższa niż w modelu podstawowym. Próbowałem również zmieniać wartość Dropout z 0.3 na wyższe i niższe, ale wyniki również nie były zadowalające.

3.3.2. Model 6

Ostatecznie przetestowałem model, w którym nie dodaję zupełnie funkcji Dropout, ale zarówno zwiększam liczbę filtrów w warstwach konwolucyjnych do 32, jak i liczbę neuronów warstwy Dense do 256, 64 i 4. Oto wyniki:

- `Accuracy` = 0.7731
- `Val_accuracy` = 0.7056

Wynik ten jest wyraźnie lepszy od `Modelu 1` i ku mojemu zaskoczeniu model również nie wygląda na przeuczony. Niżej zamieszczam wykres dla porównania:



Rysunek 3.2: Porównanie `Modelu 1` z najlepszą kombinacją parametrów jaką udało mi się osiągnąć w `Modelu 6`.

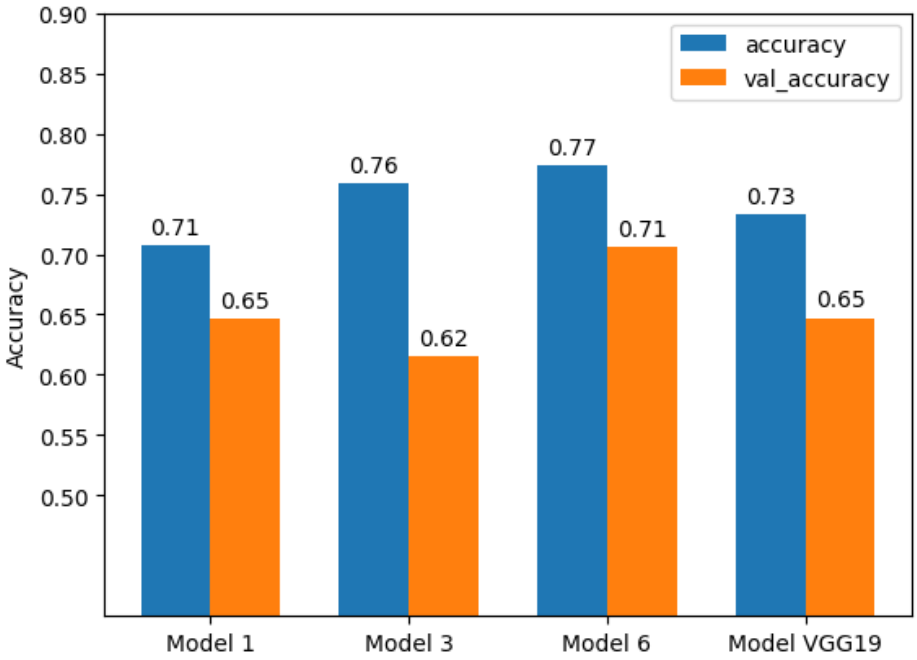
3.3.3. Model VGG19

`Model VGG19` - wcześniej opisany pretrenowany model o sporej skuteczności klasyfikacji obrazów. Przy naszym zadaniu dopasowania wieku do zdjęcia twarzy, uzyskał następujące wyniki:

- `Accuracy` = 0.7327
- `Val_accuracy` = 0.6472

Wypada on nieco lepiej od naszego podstawowego modelu, ale wyraźnie gorzej od najlepszego jakiego udało się osiągnąć.

3.3.4. Wykres porównujący 4 kluczowe modele



Rysunek 3.3: Porównanie najlepiej prosperujących modelów: `Modelu 1` - podstawowego, `Modelu 3` - ze zwiększoną liczbą neuronów Dense, `Modelu 6` - najlepszego jakiego udało mi się uzyskać i pretrenowanego `Modelu VGG19`.

Podsumowanie

Reasumując, w mojej pracy użyłem różnych modeli sieci do klasyfikacji zdjęć twarzy na grupy wiekowe: 6 - 20 lat, 25 - 30 lat, 42 - 48 lat i 60 - 98 lat. Sieci neuronowe poradziły sobie z tym na prawdę dobrze. Najlepszy wynik, jaki udało się osiągnąć, to:

- `Accuracy` = 0.7731
- `Val_accuracy` = 0.7056

Losowe rozmieszczenie przy 4 grupach daje nam 25% skuteczności. `Modelu 6` uzyskał wynik na danych testowych równy 70% poprawnej klasyfikacji zdjęcia do grupy wiekowej. Co uważam, że jest na prawdę dobrym wynikiem. Zdjęcia te nie są pozbawione tła, więc zawierają sporo szumu informacyjnego w perspektywie algorytmu, który analizuje piksel po pikselu. Są też w niskiej rozdzielczości, posiadając silniejszy sprzęt można by wytrenować algorytmy na większych obrazach, czyli większej ilości danych, z których modele mogłyby wyciągać wzorce świadczące o wieku.

Bibliografia

- https://www.statsoft.pl/textbook/stneunet.html#artificial