

Final Report

Nurkhan Zhanbayev, Daniil Tsoy (IT-2105)

Machine Learning

Celebrity Face Recognition

Youtube: <https://youtu.be/8YSQ34mBMiE>

GitHub: <https://github.com/OperatorHiKE/Celebrity-recognition-model.git>

Table of Content

Introduction	3
Problem	3
Literature review	3
Current Work	3
Data and Methods	3
Information about the data	3
Machine Learning Models with description	4
Results	4
Implementation and observations	4
Discussion	6
Critical view of results	6
Next steps for future	6

Introduction

Problem

The face recognition of celebrities is the classification task in machine learning. The application of recognition of stars can be found in the identification of real stars against double people who mostly look identical to a real celebrity, but for a computer, it can be established that they are different people.

Literature review

For this project we made some research and decided to use these documents:

- A systematic study of the class imbalance problem in convolutional neural networks <https://arxiv.org/abs/1710.05381>
- Striving for Simplicity: The All Convolutional Net <https://arxiv.org/abs/1412.6806>

Current Work

- Star by Face (<https://starbyface.com/>) - a website that can identify the celebrity by an uploaded photo by a user.

Data and Methods

Information about the data

For the development of a model we will use the dataset from `sklearn.datasets.fetch_lfw_people` called Labelled Faces in the Wild, which we distributed the data by the following:

Number of examples: 3023

The dimensionality of images: (154, 154, 3)

number of unique classes (people): 62

To avoid overfitting we used a data augmentation technique where we used the following methods to reshape the pictures to train the model.

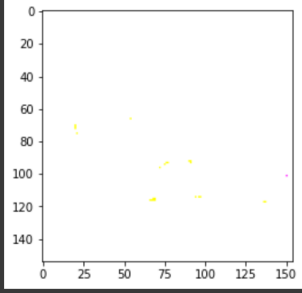
```
datagen = ImageDataGenerator(  
    rotation_range=10,  
    width_shift_range=0.1,  
    height_shift_range=0.1,  
    shear_range=0.05,  
    zoom_range=0.1)
```

```

i = np.random.randint(len(lfw_people.images))
plt.imshow(lfw_people.images[i] * 255)
plt.grid(False)
print(target_names[y[i]])

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Colin Powell

```



This snippet shows how an image is translated to an understandable format for a model for further prediction. It is not clearly seen, but

Machine Learning Models with description

The main Deep Learning model in our project is a Convolutional Neural Network with the following structure:

Model: "model"

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	(None, 154, 154, 3)	0
conv2d (Conv2D)	(None, 154, 154, 12)	336
conv2d_1 (Conv2D)	(None, 77, 77, 12)	588
conv2d_2 (Conv2D)	(None, 77, 77, 16)	1744
conv2d_3 (Conv2D)	(None, 38, 38, 16)	1040
conv2d_4 (Conv2D)	(None, 38, 38, 24)	3480
conv2d_5 (Conv2D)	(None, 19, 19, 24)	2328
dropout (Dropout)	(None, 19, 19, 24)	0
conv2d_6 (Conv2D)	(None, 19, 19, 62)	1550
global_average_pooling2d (GlobalAveragePooling2D)	(None, 62)	0
activation (Activation)	(None, 62)	0
=====		
Total params: 11,066		
Trainable params: 11,066		
Non-trainable params: 0		

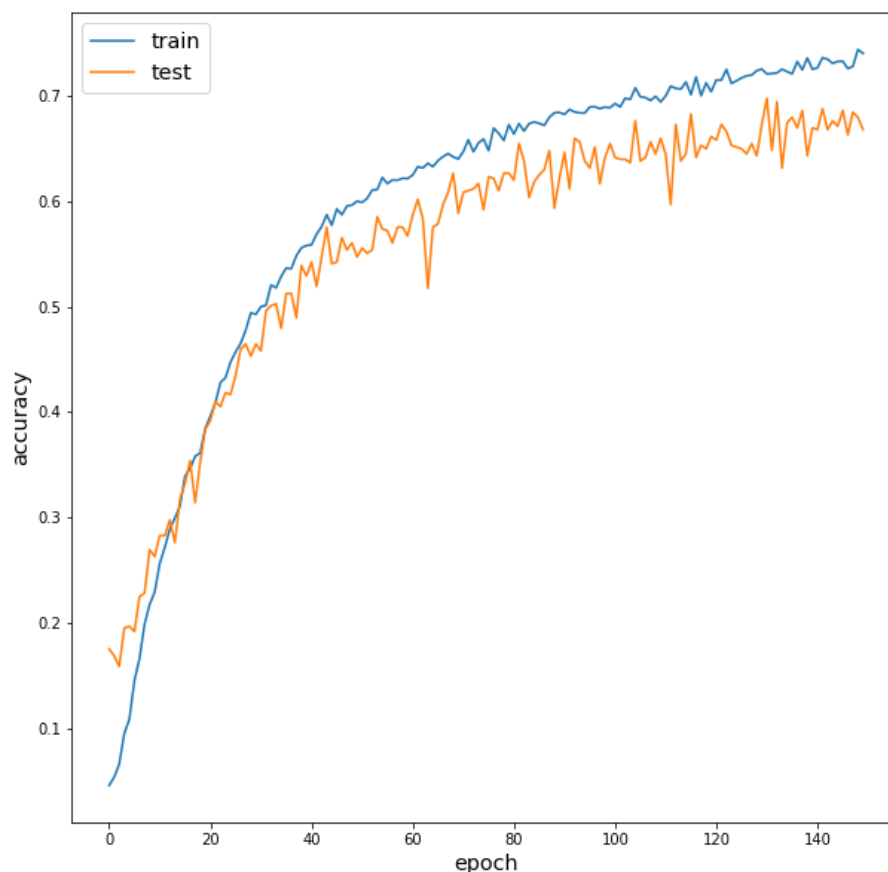
Our model consists of 9 layers. The first 6 of which are convolutional layers with filters from 12 to 24, (2, 2) core size and steps in every second convolutional layer. After that, we drop out 50% of all neurons to prevent overtraining. Then, we add the last convolutional layer to match the number of classes available.

In total, we have 11.066 parameters that our model will use for the recognition of celebrities.

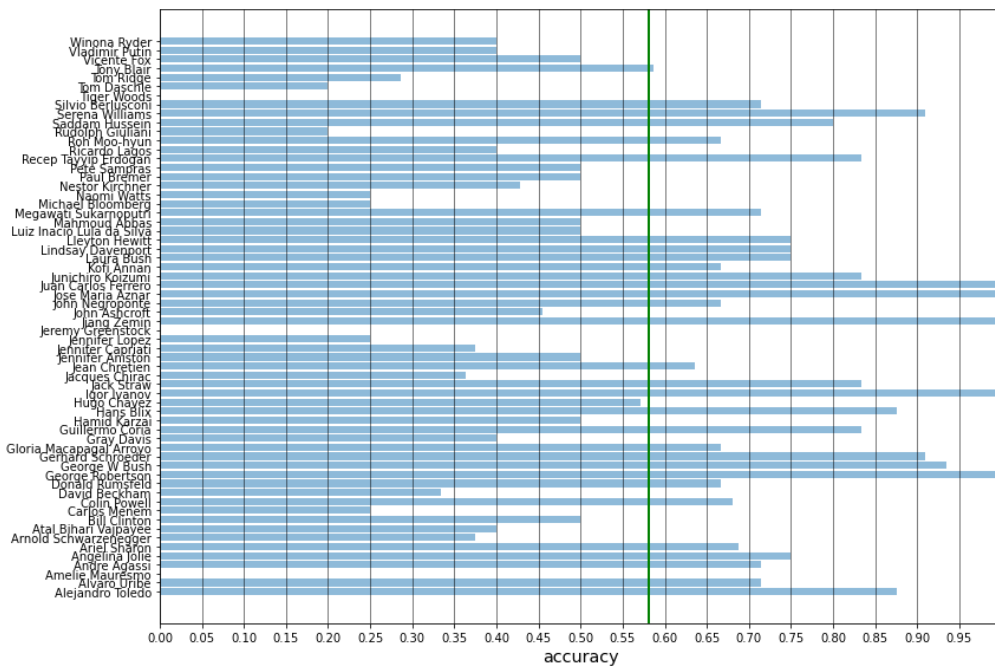
Results

Implementation and observations

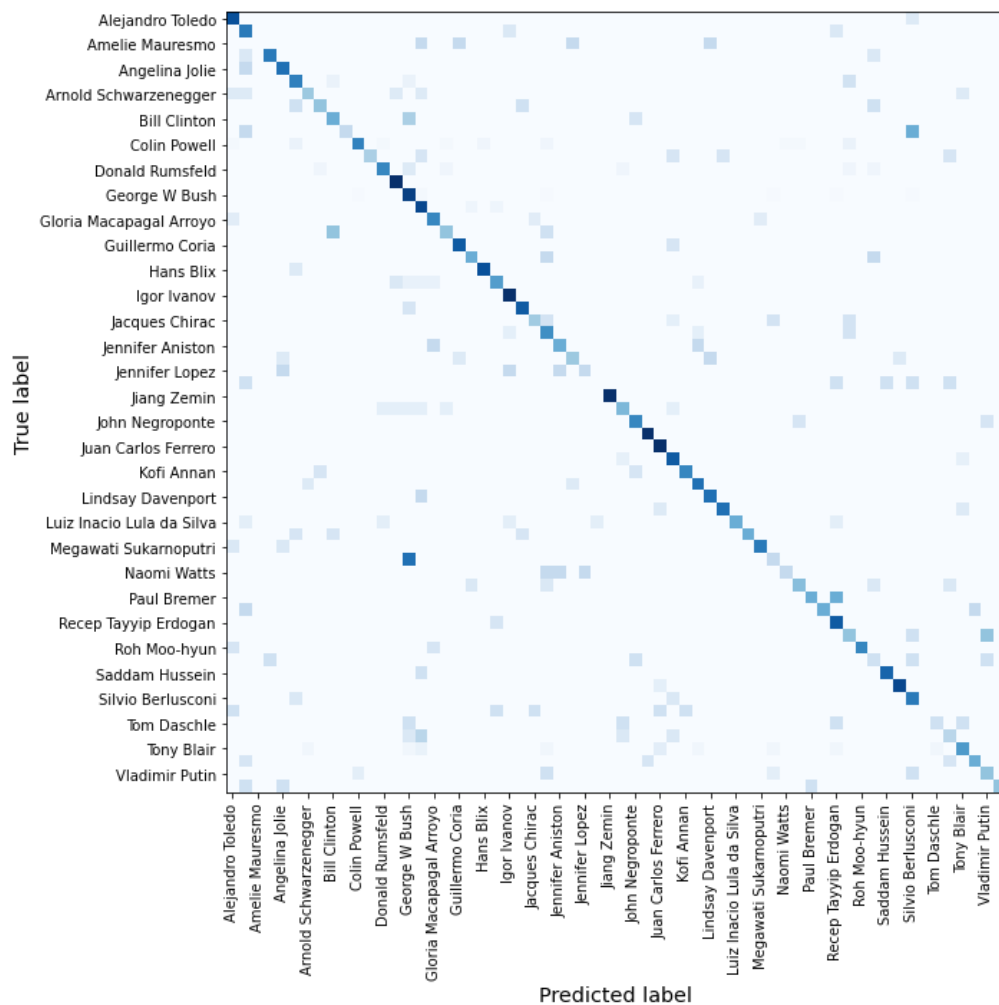
The highest accuracy achieved in epoch 150 is 74%, and the average accuracy in the test is 58.1%



This graph shows the development of accuracy during each epochs of train and test sets.



This graph shows the average accuracy of each target name. The model mostly predicted target names with more than 70% accuracy, but there are names that the model could not predict.



The confusion matrix shows the relationship between true labelled and predicted values. It can be seen that the classification of celebrity models has been a tough task for our model. We calculated that the recall and precision of our models are 60%.

Discussion

Critical view of results

The average accuracy of the model shows that there is needed further development of the model even after the completion of 150 epochs. In industrial use cases, the model can make mistakes by misunderstanding the given photo of a celebrity which can cause trouble. Therefore, there is a need to improve the model. However, by using the free Google Colab sources we can not develop it more due to restrictions of RAM memory and GPU power, otherwise, the development will take more time due to the limit of powers.

Next steps for future

There are some steps to improve our model:

- Add more layers with additional strides and padding
- Increase the number of examples in the dataset or use a different dataset
- Increase the batch size after increasing the power of the machine
- Implement the model in website or app

Sources:

- A systematic study of the class imbalance problem in convolutional neural networks <https://arxiv.org/abs/1710.05381>
- Striving for Simplicity: The All Convolutional Net <https://arxiv.org/abs/1412.6806>
-