Facial Keypoints Recognition

Tomáš Kalabis

ČVUT - FIT

kalabto2@fit.cvut.cz

December 30, 2022

1 Introduction

Facial keypoint recognition is a problem of computer vision. The objective of this task is to predict keypoint positions on face images (specifically coordinates in the image). These keypoints can be used for face tracking, emotion recognition, bio-metrics etc.

Assignment of task was taken from *Kaggle* competition *Facial Keypoints Recognition* and can be found on website here.

2 Input data

Obtained datasets were enclosed to competition (available here). Data files contains of 4 datasets (*training.csv*, *test.csv*, *IdLookupTable.csv*, *SampleSubmission.csv*). Dataset *training.csv* contains 7049 rows and consists of 30 features describing 15 keypoints of an additional feature *Image* which contains picture saved in textual form. Test dataset contains 1783 rows with an Image to recognize. Other two datasets are used for prediction of test image dataset submitting. *Submissions.csv* dataset can be submitted and evaluated in Kaggle.

List of all keypoints for predictions:

- left_eye_center
- right_eye_center
- left_eye_inner_corner
- left_eye_outer_corner
- right_eye_inner_corner
- right_eye_outer_corner
- left_eyebrow_inner_end
- left_eyebrow_outer_end
- right_eyebrow_inner_end
- right_eyebrow_outer_end
- nose_tip

Figure 1: Sample image of image from dataset with highlighted keypoints

- mouth_left_corner
- mouth_right_corner
- mouth_center_top_lip
- mouth_center_bottom_lip

Sample image with highlighted keypoints is captured on image 1.

All enclosed records in datasets were valid, however there were many missing values. Specifically there are only 2140 fully labeled records out of 7049. And after train / validation split (70%/30%) there is only 1498 left for training.

Due to a lack of data I augmented data to double by mirroring an image and feature values as proposed in [2]. I also tried to preprocess an image with CLAHE normalization. This histogram equalization of grayscale values creates bigger contrasts between them.

Paper [2] also proposes special method for including incomplete data into train / validation dataset. However this was not implemented but it is worth trying it.

3 Methods

The original idea was to train 15 models to each facial keypoint. This would enable usage of whole dataset. However this solution would increase training time so I choose to not go this way. Article [1] tried for this specific problem use Classical CNN, LeNet and VGGNet where the last one was the most successful. However VGGNet is prone to vanishing

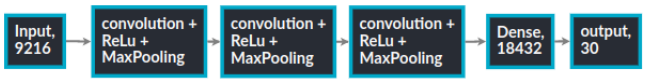

Figure 2: CNN architecture

gradient problem, so I will try ResNet architecture as proposed in [2] instead. I will also try Simple CNN for comparison.

3.1 Simple CNN

Simple CNN has on input 9216 neurons (96×96) representing values of grayscale pixels. Architecture consists of 4 convolution + maxpool layers followed by one dense layer. Whole architecture is captured on figure 2. Loss function was MSE, optimizer was Adam, activation function ReLU, batch size was set to 256 and learning rate 0.01 worked best for me. For training I set 200 epochs (I could thanks to smaller dataset).

3.2 ResNet34

This architecture was big improvement of decreasing validation loss. This is possible because of repeating input in residual block. Training parameters were the same as for the simple CNN 3.1. Whole architecture is captured on figure 3.

4 Performance

In this section are described performances of different models that are captured in table 1. Training of individual models was performed in the Google Colab environment on GPU. Most of the training sessions lasted no more than 30 minutes. For Classical CNN were results poor so after a few trainings I did not further take it into account.

Table 1: Model performance

model	training loss	validation loss
ResNet34 - no prepr. + augm.	2.236	2.690
ResNet34 - prepr. + augm.	0.963	2.301
ResNet26 - prepr. + augm.	0.859	2.078
ResNet18 - prepr. + augm.	**0.589**	**2.053**
ResNet18 - prepr. + augm.	1.793	2.307

4.1 ResNet

ResNet had quite good performance although mostly it should have been ended earlier because of overfitting. Prediction of model is captured on figure 5. Some techniques were tried for decreasing testing loss / overfitting such as LeakyReLU, Dropout (0.2)

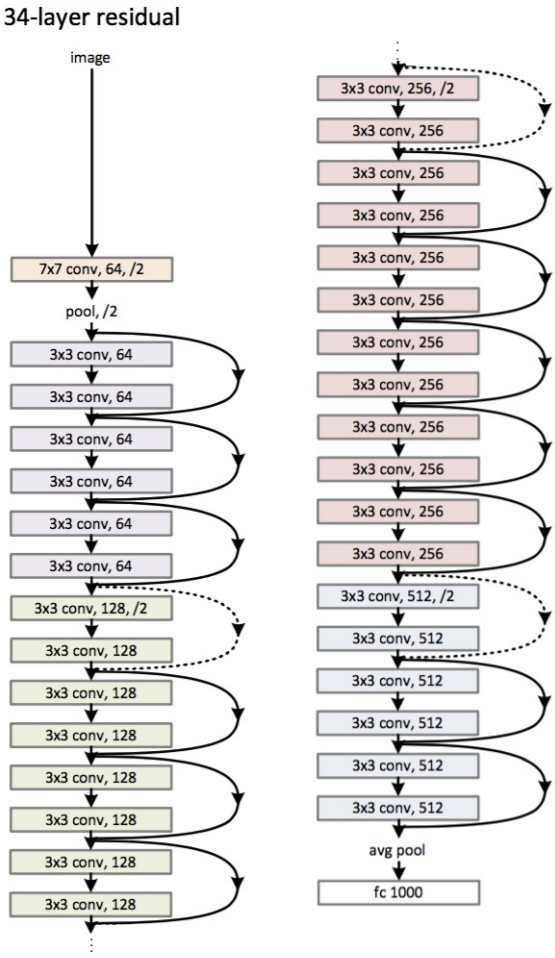

Figure 3: ResNet34 architecture

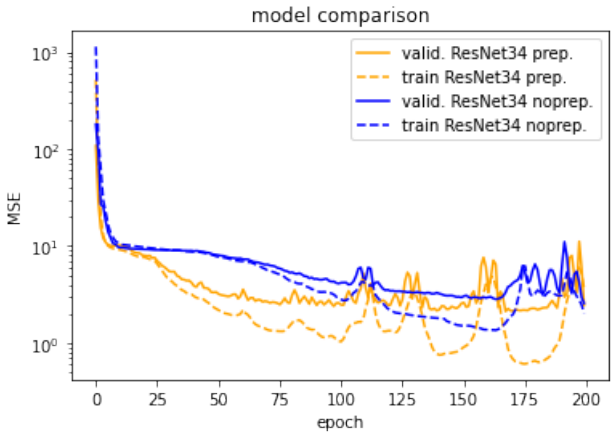

Figure 4: ResNet34 training

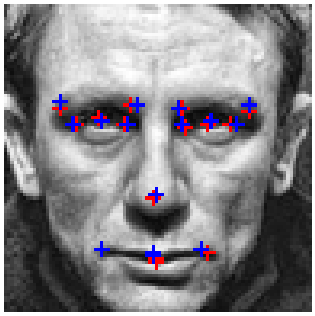

Figure 5: prediction of keypoints (blue) and actual targets (red)

and different learning rate, however they didn't have notable improvement. Also from figure 6 can be seen that preprocessing and augmentation doesn't significantly (however slightly) improve performance, although they look more stable. For improving results I also tried to reduce network size which also slightly helped. That is probably due to excessive ammount of trainable parameters in bigger networks. Also on figure 4 are captured Resnet34 with and without preprocessing with their train and validation loss from which is visible overfitting issue.

As a possible improvements I could see another data augmentation and usage of incomplete data which could be booster to the network because approx. 3000 training data is not enough. Overall I would say that results are satisfactory but could be better especially issues with overfitting.

5 Conclusion

The task of facial kypoints recognition had satisfactory results and could be further used in tasks such as emotion recognition. The best model was little above 2 MSE loss in validation dataset and after submitting testing dataset (ResNet18 without dropout and with preprocessing - possibly best)

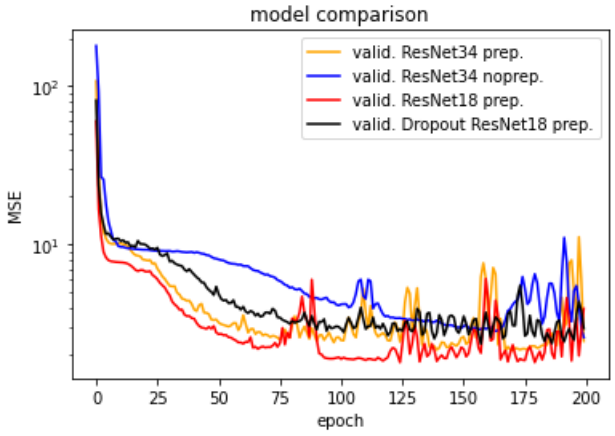

Figure 6: Comparison of models

to Kaggle I had my model evaluated to approx 3 MSE loss. On the other side many improvements could be done for decreasing overfitting such better preprocessing mentioned earlier.

References

[1] Savina Colaco and Dong Seog Han. Facial keypoint detection with convolutional neural networks. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 671–674. IEEE, 2020.

[2] Shaoen Wu, Junhong Xu, Shangyue Zhu, and Hanqing Guo. A deep residual convolutional neural network for facial keypoint detection with missing labels. *Signal Processing*, 144:384–391, 2018.