Facial Feature Localization

Ayushi Vadwala†
Computer Science
California State University -
Sacramento CA USA
avadwala@csus.edu

Jeet Shah
Computer Science
California State University -
Sacramento CA USA
jeetshah@csus.edu

ABSTRACT

Facial feature detection is an important and challenging problem in the fields of machine learning and computer vision. The most important 4 applications of facial feature localization are face recognitions, medical purposes, tracking faces in image and videos and at last face filters. In this contribution we introduce a method to do feature localization more accurately. In this project, we are given a list of 96×96-pixel 8-bit gray level images with their corresponding (x, y) coordinates of 15 facial keypoints. We first adopt Sklearn train test split to randomly split the data set into a training set and a test set, so that we can develop our algorithm on the training set and assess its performance on the test set. The baseline model was Convolutional Neural Network and our aim is to outperform the results of it by using Transfer Learning and Cascaded Convolutional Neural Network. Also the project shows one of the applications that is facial filters on the predicted facial key points.

KEYWORDS

Facial Keypoints, CNN (Convolutional Neural Network), Transfer Learning, Cascaded Convolutional Neural Network.

INTRODUCTION

Face recognition is one of the famous vision challenges. Many researchers have worked on this topic. The convolutional neural networks known to be the state-of-art so far [1], Through this project we want to focus on increasing the accuracy of the of the problem by using transfer learning and cascaded convolutional neural networks.

The dataset we are going to use is provided for Kaggle Competition [2]. Our data set contains of a list of 7049 two-dimensional 8-bit gray level training images.

Facial Key points	
Left eye center	Right eye center
Left eye inner corner	Right eye inner corner
Left eye outer corner	Right eye outer corner
Left eyebrow inner end	Right eyebrow inner end
Left eyebrow outer end	Right eyebrow outer end
Mouth left corner	Mouth right corner
Mouth center top lip	Mouth center bottom lip
Nose tip	

Table 1: 15 Facial Key Points

The images have their respective coordinates of the 15 facial key points as listed in Table 1. The images are represented as a 96 × 96-pixel matrix. The matrix entries are integers from 0 through 255, characterizing the intensity of each of the 96 × 96 = 9216 pixels. In the given training set the first 30 columns gives the (x, y) values of the 15 facial keypoints, and each entry in the last column lists the number representing the pixel matrix of each image. Thus, the training matrix is of the size 7049 X 31.

PROBLEM FORMULATION

The aim of the project is to find the facial features on the gray scale images given as input. The output of the model will be the predicted value of the facial coordinates in (x, y) format which will help in placing the facial filter on the image.

SYSTEM / ALGORITHM DESIGN

The algorithms focused during the course of this project are CNN, Transfer Learning and Cascaded CNN.

3.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) takes input as an image, assign importance to various aspects/objects in the image to differentiate one part from another. The Convolution layer helps in reducing the image size for processing, so that the data which is critical for prediction is not lost. The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. Pooling layer helps in decreasing the computational power requirement for data processing.[8]

[3]The model architecture used for CNN is given below:

Layer 1: Conv Filter = 32, Kernel size = (3,3), MaxPool = (2,2)
Layer 2: Conv Filter = 64, Kernel size = (3,3), MaxPool = (2,2)
Layer 3: Conv Filter = 128, Kernel size = (3,3), MaxPool = (2,2)
Layer 4: Conv Filter = 256, Kernel size = (3,3), MaxPool = (2,2)
Dense Layer: units = 500
Output Layer: units = 30

Table 2: Layers in CNN model

For each neuron in the convolutional layers the activation function was RELU non-linearity (max (0, x)) function.

3.2 Transfer Learning

Transfer learning is a machine learning method where a model developed for some task is reused as a starting point for another task. Transfer learning is used to accelerate the training and improve the performance of the deep learning model.

VGGFace 2 is a pre trained model on face recognition. The dataset contains 3.31 million images of 9131 subjects. Images are downloaded from Google Image Search and has large variations in pose, age, illumination, ethnicity and profession. The whole dataset is split to a training set (including 8631 identities) and a test set (including 500 identities).[9]

This library can be installed via pip:
sudo pip install git+https://github.com/rcmalli/keras-vggface.git

After successful installation, you should then see a message like the following:
Successfully installed keras-vggface-0.6

You can confirm that the library was installed correctly by querying the installed package:
pip show keras-vggface

The model is trained on colored images. It takes input of size 224 x 224 x 3. The shape of the image in the Kaggle dataset is 96 x 96 x 1. To use the VGGFace pretrained model we need to change the shape of data to the 224 x 224 x 3. Once the shape of data is changed the data is ready for the VGG model.

First, we can load the VGGFace model without the classifier by setting the 'include_top' argument to 'False', specifying the shape of the output via the 'input_shape'. After that adding a couple of dense and dropout layers to get the result in the desired output pattern.

3.3 Cascaded Convolutional Neural Network

The cascaded convolutional neural network has a Main network and four classification branches. Here in the model we have used much less filters in comparison with CNN. As a result, the whole network is very light compared with a conventional network. [10]

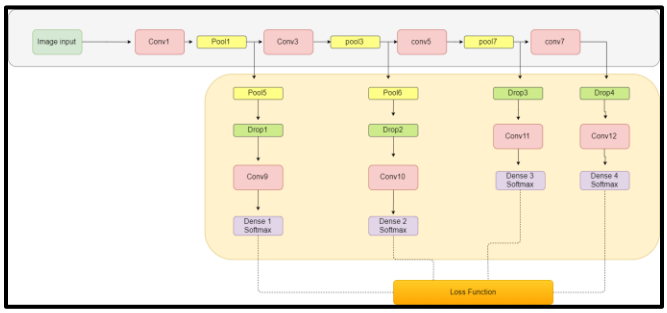

Figure 1: Cascaded CNN Architecture

With the proposed convolutional network cascade for facial point detection on the first level highly robust initial estimations are provided, while shallower convolutional networks at the following levels finely tune the initial prediction to achieve high accuracy. The Architecture of our proposed model is shown in Figure 1 and the filters on each layer are mentioned in Table 3.

Main Network
Conv1 Filter = 64, Kernel size = 4
Pool1 = (2,2)
Conv3 Filter = 64, Kernel size = 4
Pool3 = (2,2)
Conv5 Filter = 32, Kernel size = 2
Pool7 = (2,2)
Conv7 Filter = 16, Kernel size = 4
Classification Branch
Pool5 = (2,2)
Drop1 = 0.25
Conv9 Filter = 32, Kernel size = 2
Dense1 = 20
Pool6 = (2,2)
Drop2 = 0.25
Conv10 Filter = 32, Kernel size = 4
Dense2 = 10
Drop3 = 0.25
Conv11 Filter = 16, Kernel size = 4
Dense3 = 10
Drop4 = 0.25
Conv12 Filter = 8, Kernel size = 4
Dense4 = 10

Table 3 : Cascaded CNN Architecture

EXPERIMENTAL EVALUATION

4.1 Methodology

The given problem is from Kaggle, the test file provided by them does not have the key point values, as the problem is for a competition thus only image data is given. So, for testing our models we spilt the training into train and test parts.

The statistics about the key point suggests that this dataset, only 2140 images are "high quality" with all keypoints, while 4909 other images are "low quality" with only 4 keypoints labelled which are the outliers. Figure 2 shows the graph analysis of the features. The model training will be done only on the "high quality" data

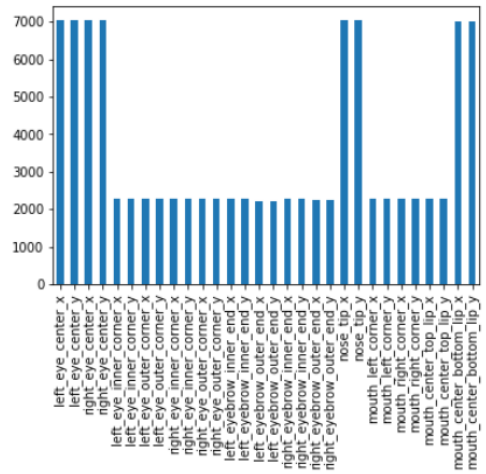

Figure 2: Analysis of training dataset's value counts

After doing the train-test split the train dataframe has 1712 images and test dataframe has 428 images. We also predicted facial features for test data given by Kaggle and stored the feature location in another output file. Models used for training the dataset are Convolutional Neural Network (CNN), Transfer learning using VGGFace model and Cascaded Convolutional Neural Network (CCNN). At last Regression chart is drawn to see the predicted facial feature coordinates versus actual coordinates.

4.1.1 Convolutional Neural Network

For CNN we tried different parameter tuning the best results we got was for the model explained above. Other experiments are mentioned in the table 3. All the models were run for 500 epochs and it resulted in the respective RMSE mentioned.

CNN					
Optimization	Kernel No	Kernel Size	Activation	R2 Score	RMSE
Adam	32	(3,3)	Relu	0.46	2.22
	64	(3,3)	Relu		
	128	(3,3)	Relu		
	256	(3,3)	Relu		
Adam	32	(2,2)	Relu	0.37	2.3
	32	(2,2)	Relu		
	64	(2,2)	Relu		
	64	(2,2)	Relu		
sgd	32	(3,3)	Relu	NAN	NAN
	64	(3,3)	Relu		
	128	(3,3)	Relu		
	256	(3,3)	Relu		

Table 4 : CNN experiments

4.1.2 Transfer Learning

For transfer learning we tried different we added different dense layers at last. The results of the transfer learning with different dense layer is given in table 4.

Transfer learning		
Dense layers	Number of nodes	RMSE
2	1024,30 (125 epochs)	6.63
2	1024,30 (10 epochs)	7.69
1	30 (50 epochs)	9.72

Table 5 : Transfer learning experiments

4.1.3 Cascaded Convolutional Neural Network

For Cascaded CNN we tried the model for 1000, 100 and 50 epochs the best results we got was for the model explained above.

- For 100 epochs RMSE is 1.74 and R2 score is 0.67.
- For 50 epochs RMSE is 1.82 and R2 score is 0.64.

4.2 Analysis of Results

4.2.1 Convolutional Neural Network

For our best model we plotted the regression chart. The model follows the trend of the result but there are some fluctuations in the result.

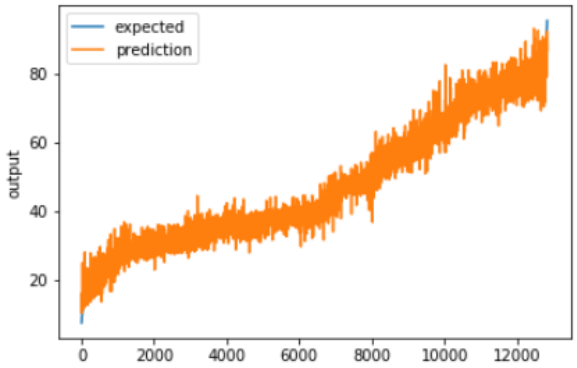

Figure 3: Regression chart of CNN for predicted facial feature coordinates versus actual coordinates

4.2.2 Transfer Learning

After analysis of transfer learning in comparison with other two model we found out that VGGFace is not giving us good results. The negative transfer refers to scenarios where the transfer of knowledge from the source to the target does not lead to any improvement, but rather causes a drop in the overall performance of the target task. There can be various reasons for negative transfer, such as cases when the source task is not sufficiently related to the target task, or if the transfer method could not leverage the relationship between the source and target tasks very well. Figure 4 shows our best RMSE for transfer learning.

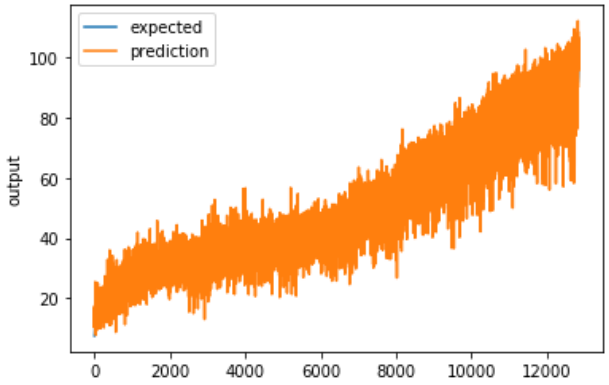

Figure 4: Regression chart of Cascaded CNN for predicted facial feature coordinates versus actual coordinates

4.2.3 Cascaded Convolutional Neural Network

For our best model we plotted the regression chart. The model follows the trend of the result but there are some fluctuations in the result.

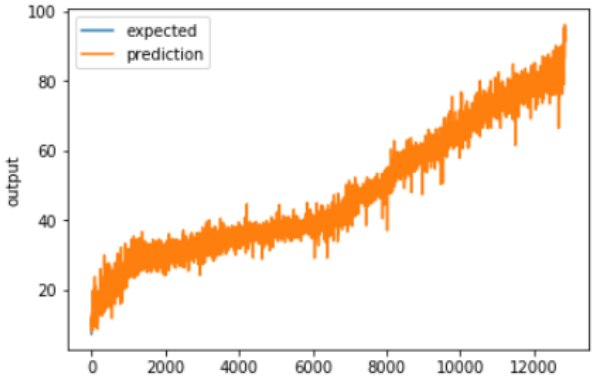

Figure 5: Regression chart of Cascaded CNN for predicted facial feature coordinates versus actual coordinates

4.2.4 Applying facial filters on the predicted values

One of the many applications of Facial Feature Detection is to apply facial filters on the image. Here for example we tried to place a moustache on the facial image using the mouth center top lip key point detected by our models.

Figure 6: Applying Facial Filter on the images using one of the feature locations predicted

4.2.5 Comparison of 3 different models :

Model	RMSE	R2
Convolutional Neural Network	2.22	0.46
Transfer learning	6.63	-3.8
Cascaded Convolutional Neural Network	1.74	0.67

Table 6 : Model comparison table

Though Cascaded Convolutional Neural Network was trained for just 100 Epochs and Convolutional Neural Network for 500 Epochs, Cascaded Convolutional Neural Network is giving better results which is shown in above table. Moreover, Cascaded Convolutional Neural Network is time efficient than Transfer Learning because to run just 1 Epoch Cascaded CNN was taking just 5 Seconds while Transfer Learning was taking 5 Minutes. So, for this dataset Cascaded CNN is best and time efficient model.

RELATED WORK

Significant progress on facial feature localization has been achieved in recent years. Many researchers used Support Vector Regression (SVR) [1] and Convolutional Neural networks [3] which gives them good RMSE. Hao Zhang in their paper used 4 CNN layers to solve the facial feature detection problem and their method resulted in the RMSE of 3.3 without dropout layer and 3.2 with dropout layer.

Furthermore, cascading convolutional neural network architectures proposed by Sun.Y et.al. [4] have been used to avoid local minima caused by ambiguity and data corruption in difficult image samples

due to occlusions, large pose variations, and extreme lightings. This approach shows solid results.

S. Longpre et.al [5] mentioned in their future work that Transfer learning using VGGFace can further increase the accuracy of the model. VGGFace is a pretrained model, finetuned over the last few layers. This could also be a contender for an effective ensemble model

Ran Gao and Qi Liu [6] experimented with different CNN architecture including the LeNet-5, VGGNet and a 14-layer network. They also experimented with various optimization algorithms and other techniques including dropout, data augmentation to increase the prediction accuracy on the test set.

Moreover, Shenghao Shi [7] has applied ten different methods to the Kaggle facial keypoints detection data set. CNN has the best RMSE with only 1.972, but it's time consuming. Although decision tree does not have low RMSE, the it takes advantage of interpretability. Linear regression methods are easy enough to train but are not recommended for real world applications. Neural network and KNN both get surprising result and need further investigation.

CONCLUSION

A wide spectrum of learning models are available for Facial Feature Localization. The three models applied to detect facial keypoints and their performance in the paper are CNN, Transfer Learning and Cascaded CNN. Transfer learning through VGGFace 2 did not give satisfactory results. The CNN gave good results, but Cascaded CNN gave better results than CNN. Our method of Cascaded CNN significantly improves the prediction accuracy of state-of-the-art method CNN. Thus, it can be said that Cascaded CNN is best suited for the Kaggle's facial keypoint detection dataset.

WORK DIVISION

Name: **Ayushi Vadwala**

Tasks performed:

- Removed rows with any null values and Removed duplicate rows
- Feature Normalization
- Implemented model (CNN) and Parameter Tuning for the same.
- Transfer Learning and Cascaded CNN.
- Worked on Report

Name: **Jeet Shah**

Tasks performed:

- Split the data into train and test data.
- Implemented facial filters
- Transfer Learning and Cascaded CNN
- Worked on presentation
- Prediction for the Test data and compared actual and predicted result.

LEARNING EXPERIENCE

- CNN is the state of the art, but it takes time in learning. The accuracy given by CNN is good.
- In transfer learning a pretrained model is trained on different images and it is taken as a starting point for the current model.
- It is difficult to up sample the images from 96 x 96 to 224 x 224. For transfer learning the size of the images needs to be changed so that the pretrained model can accept the images from our dataset.
- Cascaded CNN gives surprising results. The RMSE and R2 score of the cascaded CNN is better than the CNN.
- Any facial filters can be applied on the face if the facial feature location is known.

REFERENCES

[1] Esmaeili, A., Khosravi, K., & Mirjalili, S. (2015). Facial Keypoint Detection.
[2] https://www.kaggle.com/c/facial-keypoints-detection
[3] Zhang, H., Chen, J., & Agarwal, N. Facial Keypoints Detection.
[4] Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3476-3483).
[5] Longpre, S., & Sohmshetty, A. (2016). Facial Keypoint Detection.
[6] Gao, R., & Liu, Q. (2018). Facial Keypoints Detection with Deep Learning. JCP, 13(12), 1403-1410.
[7] Shi, S. (2017). Facial Keypoints Detection. arXiv preprint arXiv:1710.05279
[8] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[9] http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/
[10] Lu, K., Chen, J., Little, J. J., & He, H. (2017). Light cascaded convolutional neural networks for accurate player detection. arXiv preprint arXiv:1709.10230.