Fall 2021

COMP 8740: Neural Networks
Final Project Report

Complete Graph Distance in Facial Keypoint Detection using Deep Learning

Hosneara Ahmed

December 5, 2021

1 Introduction

Object detection is a challenging problem in neural network, even harder than classification problem. However, in case of face detection, it becomes more difficult for various factors such as different orientation of the face image, occlusion, noise, etc. Face detection means to localize a human face, predicting the bounding box over the face.

Facial keypoints detection is a bit different than face detection and a harder problem. Facial keypoints include nose, lip, eyes, eyebrows. The problem involves predicting the coordinates of these keypoints.

For detection problem, transfer learning has become popular and effective. However, it can be a strong backbone for any problem. Transfer learning works in the same way, we human observe and learn things. A model that is pre-trained on a dataset and the weights are learned can be used to solve another problem. No matter if the new dataset is small or much different, it works better than training model from scratch. The reason is, in a deep learning model, the bottom layers are mostly for predicting low level features. As proceeding to top layers, more problem specific features are to be learned. So, if the low level features are already learned with proper weight or at least initialized with something close to that, then it is much easier to learn the later part.

In this project, we have used transfer learning to solve facial keypoints detection problem. Additionally, we have incorporated Complete Graph Distance (CGD) [1] loss function and compared the results with MSE loss.

2 Methodology

2.1 Dataset Description

We have used the facial keypoint dataset from Kaggle [2]. This dataset contains 7049 training data images with associated keypoints and 1783 test images. Images are of shape (96x96x1) that means they have only one channel. The keypoints are: two corners and center of right and left eye, two corners of left and right eyebrows, nose tip, corners and center of lip. However, there are about 68% missing data in the eyes, eyebrows, and lip keypoints.

2.2 Data pre-processing

- **Missing data imputation:** For imputing the missing keypoints, we have used forward filling method which means, the last observed keypoint value will be used to fill out the following missing values.

- **Normalize images:** The images have value range from 0 to 255. To keep them between 0 and 1, we applied max-normalization to all the images.

2.3 Transfer learning

We have used ResNet-50 and DenseNet-121 pretrained models from keras as the backbone of our model. We have finetuned the top layer for the difference in number of last layer features (predicting 30 keypoints for each face image). Also, as both ResNet and DenseNet were trained for imagenet data, those images have 3 channels unlike this data. So we have used 3 filters of 1x1 convolution size to have 96x96x3 sized images.

2.4 Complete Graph Distance (CGD) loss function

Inspired from [1], where authors have used bipartite distance measure along with MSE loss to detect the landmark in spinal X-ray images, we have implemented CGD loss where the distance between each keypoint will be used along with MSE to penalize the model in training.

CGD is a shape-aware loss which calculates the euclidean distance d between each pair of ground truth and predicted coordinates to learn what is the actual distance between each pair of keypoints and the distance between predicted keypoints. So, the face is considered as graph where the keypoints are the nodes.

$$CGD = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}|d_i - d_j| \tag{1}$$

The loss function we used is-

$$MSE + \alpha CGD \tag{2}$$

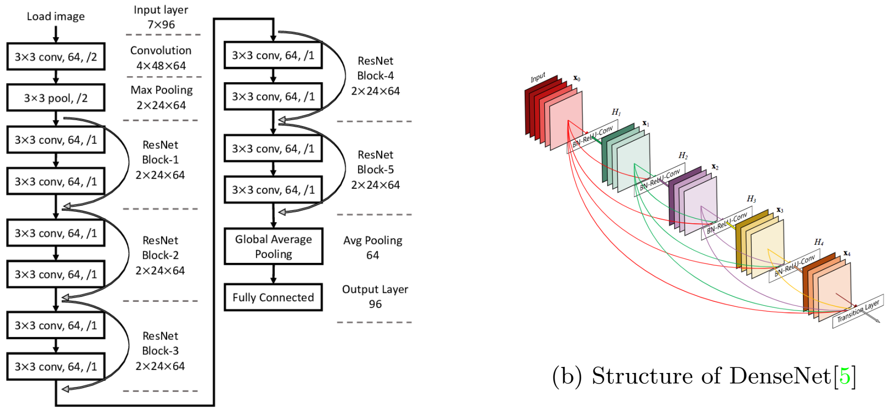
We have used the weight of CGD, $\alpha = 0.01$. Throughout the report, whenever we say CGD, we are referring to $MSE + \alpha CGD$.

3 Deep Learning Architecture

The two baseline models that we have used: ResNet and DenseNet have been very successful in facial keypoint detection problem.

3.1 ResNet-50

ResNet50 [3], a variation of ResNet is a deep CNN network with 48 convolution layers with 1 MaxPool and 1 AveragePool layer that is primarily trained on ImageNet dataset. Unlike ResNet, ResNet50 has skip connections after each 3 convolution blocks as well as 1x1 convolution within the stacks of 3 convolution blocks. 1x1 convolution lets an architecture to change the dimension of the layer easily.



(b) Structure of DenseNet[5]

(a) Structure of ResNet[4]

Figure 1: Architectures of ResNet and DenseNet

3.2 Densenset-121

In a DenseNet model [6] there are connections from all previous layers to the following layers. This lets the model having fewer parameters considering its deep architecture. In DenseNet-121, there are 1 7x7 , 58 3x3, 61 1x1 convolution layers along with 4 average pool and finally 1 fully connected layer.

Both DenseNet and ResNet resolve the problem of vanishing gradient, even though they are very deep neural network. This is because they have simplified the connections by using skip connections (ResNet) and sharing parameter (DenseNet).

4 Experiment and Results

4.1 Experimental Setup

We used the pretrained ResNet-50 and DenseNet-121 from tensorflow.keras. We have compared the models performance based on MSE and CGD. We have also compared the performance with and without image augmentation. The batch size 64, Adam optimizer and decaying learning rate of factor 0.7. I have run the models 200 epochs with early stopping if the training loss does not change significantly in consecutive 20 epochs.

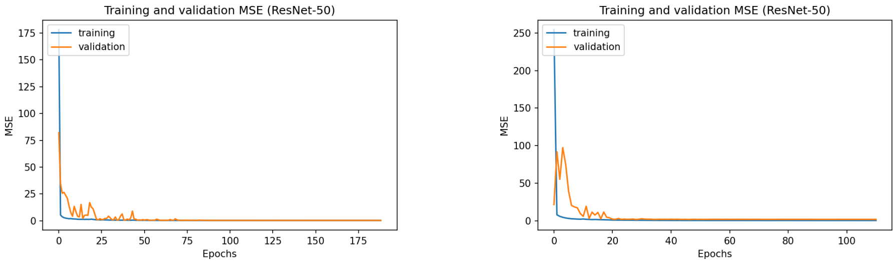
4.2 Image Augmentation

I have experimented with and without using image augmentation. For the augmentation, we have deployed rotation, noise addition, brightness changing [7]. We did not flipped the images as this is a detection problem not classification, so will be difficult for the model to learn localizing the keypoints.

- **Rotation:** We have added rotated images with 20° and $-20°$.
- **Noise:** We have added random normal noise with mean=0 and variance=0.1.
- **Brightness:** We altered brightness by increasing and decreasing by 50%.

4.3 Results

4.3.1 ResNet with MSE loss



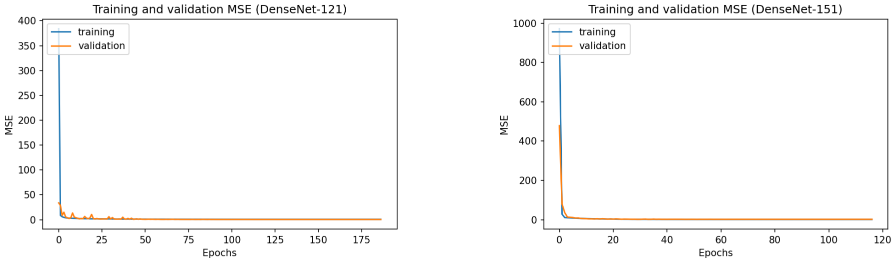
(a) With image augmentation

(b) Without image augmentation

Figure 2: ResNet-50: MSE loss

From figure 2a we can see that, model is performing well from the very beginning of the epochs for both training and validation. On the other hand, when we did not augment image 4b, then the performance is still good, but required a a few more epochs (about 30) to get to a low MSE.

4.3.2 DenseNet with MSE loss



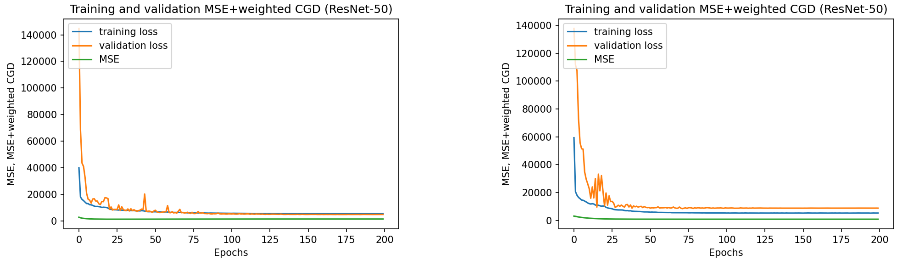
(a) With image augmentation

(b) Without image augmentation

Figure 3: DenseNet-121: MSE loss

From figure 3a and 3b, we can see the keypoint detection performance with respect to MSE loss is very good. Even though initially the MSE loss is very high, eventually it did take only a few epochs to go down.

4.3.3 ResNet with CGD loss



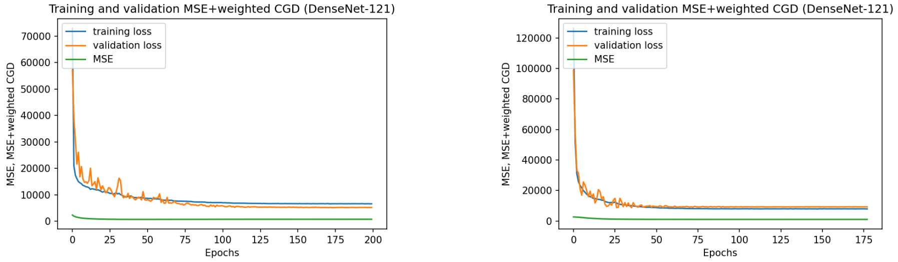
(a) With image augmentation

(b) Without image augmentation

Figure 4: ResNet-50: CGD loss

Coming to CGD loss comparing to MSE from figure 4a and 4b we have kept the MSE as metric and observed the CGD loss. However, we can see CGD cannot outperform MSE, rather loss is pretty high in both with image augmentation and without image augmentation.

4.3.4 DenseNet with CGD loss



(a) With image augmentation

(b) Without image augmentation

Figure 5: DenseNet-121: CGD loss

In comparison of CGD in case of DenseNet-121 5a and ResNet-50 4a, we see the loss is lower for CGD in DenseNet than with the CGD in ResNet-50 when image augmentation is used. Also, the validation loss in DenseNet is much smoother than ResNet when image augmentation is not added.

4.4 Facial Keypoint Prediction on Test Images

4.4.1 ResNet-50 with MSE loss

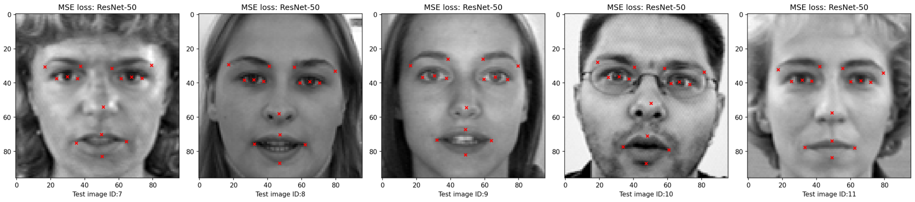


Figure 6: ResNet-50: Keypoint prediction (with image augmentation)

4.4.2 DenseNet-121 with MSE loss

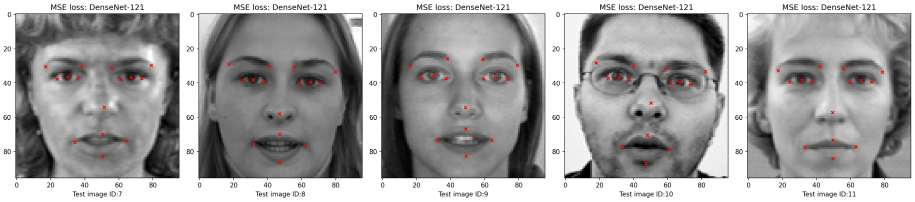


Figure 7: DenseNet-121: Keypoint prediction (with image augmentation)

In the test images, we can see the ResNet-50 and DenseNet-50 with MSE loss are performing very good in keypoint detection.

4.4.3 ResNet-50 with CGD loss

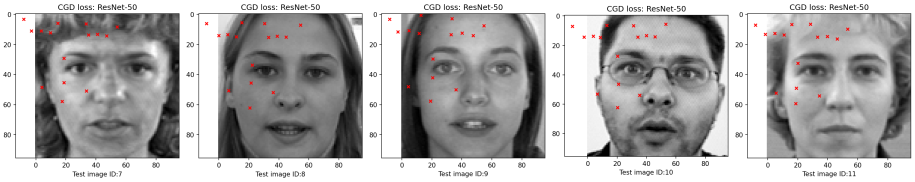


Figure 8: ResNet-50: Keypoint prediction (with image augmentation)

4.4.4 DenseNet-121 with CGD loss

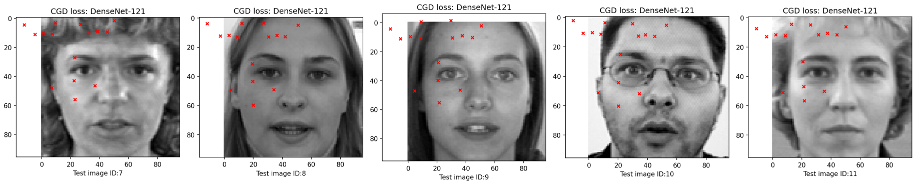


Figure 9: DenseNet-121: Keypoint prediction (with image augmentation)

However, these models are far from predicting the keypoints while penalizing with CGD loss.

4.5 Discussion and Comparison

From the results, we can see the the models are performing well while penalizing with MSE loss than CGD loss. The reason could be the nature of the dataset that we have used. In [1], the spinal X-ray images were used where the distance between the landmarks resembles the bipartite nature of a graph.

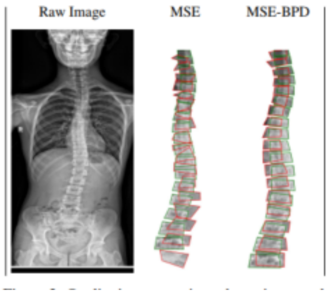


Figure 10: Spinal X-ray image with BPD loss

From figure 10 we see, there is a clear structure that can be learned using BPD loss.

However, in case of facial keypoint detection data, there is no fixed structure of distance that can be learned due to the face orientation, noise, etc. For example, the distance between the eye centers may differ if the face pose changes. So it is hard to learn the distance structure between the facial keypoints.

5 Conclusion

In this project we have deployed transfer learning with different losses. We have also explored CGD loss which is inspired from [1] to see how it performs in the facial keypoint detection dataset. However, due to the varying structure of face images, models trained with CGD could not perform better than MSE. In future, we can explore with different image dataset like spinal X-ray images to see if models penalized with CGD can detect or localize better.

6 Appendix

Code: https://github.com/Hosneara/Neural-Network-Project/tree/master

References

[1] Abdullah-Al-Zubaer Imran, Chao Huang, Hui Tang, Wei Fan, Kenneth M. C. Cheung, Michael To, Zhen Qian, and Demetri Terzopoulos. Bipartite distance for shape-aware landmark detection in spinal x-ray images, 2020.

[2] Yoshua Bengio. Facial Keypoints Detection. https://www.kaggle.com/c/facial-keypoints-detection/, 2017. [Online; accessed on 7-Dec-2021].

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[4] Seunghyoung Ryu. Short-Term Load Forecasting based on ResNet and LSTM. https://www.researchgate.net/figure/The-structure-of-ResNet-12_fig1_329954455/, 2018. [Online; accessed on 7-Dec-2021].

[5] Pytorch Team. DENSENET. https://pytorch.org/hub/pytorch_vision_densenet/, 2021. [Online; accessed on 7-Dec-2021].

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[7] Luong Quang Dung. Facial Keypoint Detection - CNN + Augmentation. https://www.kaggle.com/lqdisme/facial-keypoint-detection-cnn-augmentation/, 2021. [Online; accessed on 7-Dec-2021].