APL-405 Term-Project
Report 3

Low-Resolution (LR) face recognition using two-branch DCNN

1. Introduction

In the recent years, face recognition has shown good performance in different areas that too in challenging conditions such as occlusion, variation in pose, illumination, and expression. While there are already many face recognition tools that have been developed for recognizing high quality face images in controlled conditions, there are very few studies focused on face recognition in real world applications such as surveillance systems with low resolution (LR) faces. There may be cases where high resolution (HR) probe images can't be obtained due to some reason in that case these tools can't be used. So, to deal with the issue of recognizing LR probe face images when a gallery of high-quality images there are three standard approaches already prevailing to address this problem.

- In first approach we convert LR probe images into HR images and then compare features between that images and gallery HR images.
- In this we convert gallery HR images into LR images and then performing our method and match the features of both.
- In this approach we transform the HR & LR images into a common space and then match the feature of both images. And backpropagate the distance between the features to train our data set so that we can get lesser differences as much as possible between the features of these two HR & LR images. So, we use third one approach and find a non-linear transformation using two deep convolutional neural networks to find common space between low & HR face images. This proposed offers high recognition accuracy compared to others.

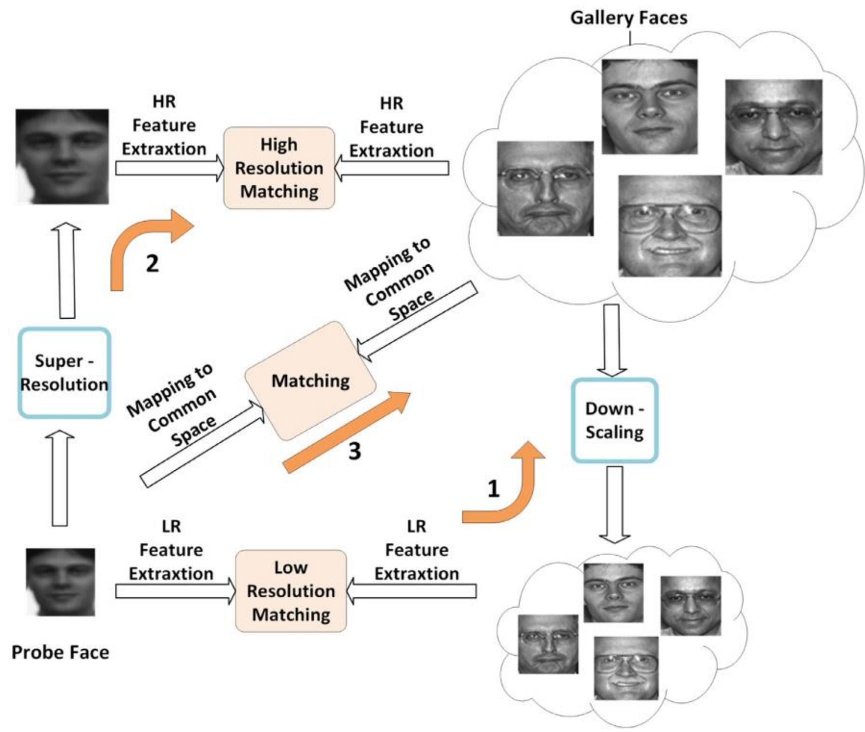

Fig. 1. Three general approaches for LR face recognition.

2. Problem Statement

Our aim is to make a novel coupled mapping for LR face recognition

So, for the mapping we try to find two nonlinear functions F_H and F_L for HR and LR images respectively that outputs a 4,096-dimensional feature vector which represents all the features of that image.

$$\phi_i^h = F_H(I_i^h) \quad (1)$$

$$\phi_i^l = F_L(I_i^l) \quad (2)$$

where $I_i^h \in R^{M\times M}$ and $I_i^l \in R^{N\times N}$ that $N < M$

And to test which of HR gallery images resembles LR image the most we can calculate features of all HR gallery images (F_H) and then we can see which HR image's feature are closest to probe image features which can be mathematically written as: -

$$k = \arg\min_j \{d_{i,j}\}_{j=1}^{N_G} \quad (3)$$

$d_{i,j}$: distance between feature of probe image and j-th gallery image

So here k-th HR gallery image will match probe image the most.

3. Methodology

So, to get a nonlinear coupled mapping two deep convolutional neural networks (DCNNs) can be used to extract features from both LR probe images and HR gallery images separately and project them into a common space. After passing both LR and HR images through DCNNs, gradient we use gradient descent algorithm to minimize the distance between the features of image pairs and hence updating the weights of DCNN of LR image by backpropagation of the error. Data used for training contains pairs of LR and HR images of same person may be in different condition such as pose, expression.

For HR images we used VGG-16 network architecture in which out of three fully connected layers last two are dropped. And it is used with pretrained weights. This network is called Feature Extraction CNN (FECNN).

For LR images we first train a 5-layer convolutional network to improve resolution of images this network is called SRNet (Super Resolution Network) and then on this network FECNN is added to extract features from LR image. This complete network is called Super-Resolution Feature-Extraction CNN (SRFECNN).

3.1 Forward Propagation: For training from pairs of HR and LR images we first pass HR image to FECNN and LR image to SRFECNN to get feature vectors of both images

3.2 Loss Function: Now to decrease the gap between feature vectors we used Mean Squared Error (MSE) loss function which needs to be minimized to get more similar feature vectors.

$$\mathrm{MSE} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}}_{\text{test set}} (\underbrace{y_i}_{\text{predicted vaue}} - \underbrace{\hat{y}_i}_{\text{actual value}})^2$$

3.3 Back Propagation: For back propagation and updating weights and biases we use gradient descent algorithm and Adam's optimizer.

**Gradient descent is a first-order iterative optimization algorithm used to update weights to decrease loss function and finding a local minimum for the loss function. In this we take repeated steps in the opposite direction of the gradient of the function to achieve the minimum value.

$$\Theta_j = \Theta_j - \underset{\substack{\uparrow \\ \text{Learning Rate}}}{\alpha} \frac{\partial}{\partial \Theta_j} J(\Theta_0, \Theta_1)$$

4. Progress

So, to implement the proposed two DCNN's approach for LR face recognition we used TensorFlow library in Python, google colab with GPU accelerator for quicker training. We collected the training data from 300-W, I-BUG, AFW and HELEN dataset and testing data from LFW, MBGC dataset. We extracted data from images, preprocessed it thorough data pipeline and saved it as TfRecords file. Training was done on dataset with batch size of 64 and for 10 epochs.

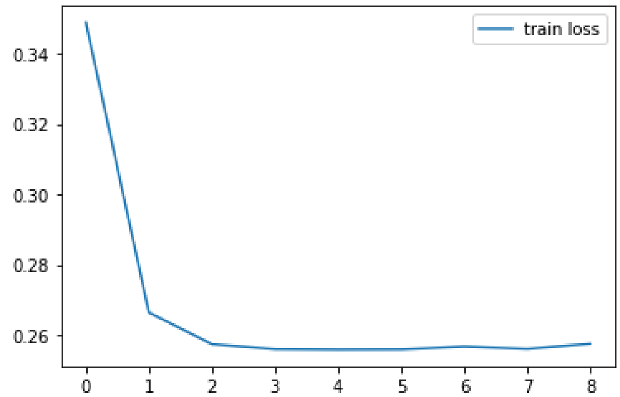

Fig 1 – Train Loss

5. Work Remaining

- Right now, we are also using a constant learning rate. We have to reduce learning rate of all layers to fine-tune the weights obtained in the first two training phases. However, the learning rate of first layers of FECNN is less than last layers of it, because in a specific problem, last layers of a DCNN have more discriminant information about the problem and the first layers of it have more general features that can change sparsely.
- Add comments and make code a bit cleaner.

Statement of contribution:

Bhavesh Gupta (2019ME10783) – coded ML part, trained on SRNet and SRFECNN, tested the model, wrote section 3, 4 and 5

Ashutosh Khandal (2019ME10781) – collected data, coded the pre-processing part, shifted on google colab wrote section 1 and 2

Reference

- Dong, C. , Loy, C. C. , He, K. , & Tang, X. (2014). *Learning a deep convolutional network for image super-resolution* (pp. 184–199) .
- Freeman, W. T. , Pasztor, E. C. , & Carmichael, O. T. (20 0 0). Learning low-level vision. *International Journal of Computer Vision, 40* (1), 25–47 .
- Le, V. , Brandt, J. , Lin, Z. , Bourdev, L. , & Huang, T. (2012). Interactive facial feature localization. In *Computer vision–ECCV* (pp. 679–692) .
- Parkhi, O. M. , Vedaldi, A. , & Zisserman, A. (2015). Deep face recognition. *British Machine Vision Conference, 1* (3), 6 .
- Sagonas, C. , Tzimiropoulos, G. , Zafeiriou, S. , & Pantic, M. (2013). *300 faces in-the-wild challenge: The first facial landmark localization challenge* (pp. 397–403) .
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for largescale image recognition. CoRR abs/ 1409.1556. .
- Zeiler, M. D. , & Fergus, R. (2014). Visualizing and understanding convolutional net-works (pp. 818–833) .