IPR Project 1 Report

Deep Density Clustering of Unconstrained Faces

Name: Divij Singla
Roll No: 210350

Dataset Description

- For Training, **CASIA-WebFace** was used. The dataset is used for face verification and face identification tasks. The dataset contains 494,414 face images of 10,575 real identities collected from the web.
  Dataset Structure: The dataset has two important files 'train.rec' and 'train.idx'.
  train.rec: This contains frames in which photographs are present which can be extracted using mxnet library in python. **(It was a challenging task to do so and learn about its data iterator! as these files are outdated, so I had to downgrade by python version to 3.7)**
  train.idx: This contains the indexing which can be used to find out labels and iterate the dataset using mxnet.

- For Testing, as the model is for unsupervised algorithm as I was not able to utilise the complete CASIA-WebFace dataset due to computational constraints, I first used randomly shuffled CASIA-WebFace only to test the results. Then, I used **Labelled Faces in the Wild (LFW)** Dataset. This dataset was easy to understand because of its vivid descriptions and handy README files. It has classified images in jpg format of different people (by their names). LFW has 13,233 images of 5,749 people.

Implementation Deatils

- **Deep Face Representation**

  1. The following network architecture is used:
  2. The network is first trained on the CASIA-WebFace dataset using SGD for 300 iterations with a standard batch size 128 and momentum 0.9.
  3. The weight decay rates of all the convolutional layers are set to 0, and the weight decay of the final fully connected layer is set to $5 * 10^{-4}$.

  Notes:

  1. The inputs to the networks are $100 * 100 * 3$ RGB images.
  2. Data augmentation is performed by randomly cropping and horizontally flipping face images.

TABLE I
THE ARCHITECTURE OF DCNN MODEL USED IN THIS PAPER.

Name	Type	Filter Size/Ouput/Stride	#Params
Conv11	convolution	3×3 / 32 / 1	0.28K
Conv12	convolution	3×3 / 64 / 1	18K
Conv13	convolution	3×3 / 64 / 1	36K
Pool1	max pooling	2×2 / 2	
Conv21	convolution	3×3 / 64 / 1	36K
Conv22	convolution	3×3 / 128 / 1	72K
Conv23	convolution	3×3 / 128 / 1	144K
Pool2	max pooling	2×2 / 2	
Conv31	convolution	3×3 / 96 / 1	108K
Conv32	convolution	3×3 / 192 / 1	162K
Conv33	convolution	3×3 / 192 / 1	324K
Pool3	max pooling	2×2 / 2	
Conv41	convolution	3×3 / 128 / 1	216K
Conv42	convolution	3×3 / 256 / 1	288K
Conv43	convolution	3×3 / 256 / 1	576K
Pool4	max pooling	2×2 / 2	
Conv51	convolution	3×3 / 160 / 1	360K
Conv52	convolution	3×3 / 320 / 1	450K
Conv53	convolution	3×3 / 320 / 1	900K
Pool5	avg pooling	7×7 / 1	
Dropout	dropout (40%)		
Fc6	fully connection	10548	3305K
Cost	softmax		
total			6995K

3. Given a face image, the deep representation is extracted from the pool5 layer with dimension 320.

- **Parameter Selection**

  The main hyperparameter in the proposed approach: is ϵ for constructing neighborhoods which was optimally chosen as 0.23 by sampling 100 random images from the dataset and calculating the cosine similarity between matched pairs at which the probability distribution is maximised.

Note

In the paper, results are reported when the model was trained on enormous 750K iterations which was not possible in my machine. Each iteration took 3-4 minutes to run, hence I was able to train the model for 300 iterations.

Results

Two measures are adopted for evaluation, NMI and F-measure.

1. **Normalized Mutual Information (NMI)**
   Best NMI achieved : 0.58

2. **BCubed F-measure**
   Best F-measure achieved : 0.63

Both of these scores can be improved significantly using GPU. Where it was required to do 750K iterations, only 300 iterations were able to be performed due to computational constraints.