UNSUPERVISED FACE RECOGNITION USING ONE-SHOT LEARNING

Thesis

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING - INFORMATION SECURITY

by

RaviRaaja L
(Reg. No.: 16IS18F)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

MANGALORE - 575025

JUNE 2018

UNSUPERVISED FACE RECOGNITION USING ONE-SHOT LEARNING

by

RaviRaaja L

(Reg. No.: 16IS18F)

Under the Guidance of

Dr. Jeny Rajan
Assistant Professor
Department of CSE

THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING - INFORMATION SECURITY

Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
Mangalore - 575025

JUNE 2018

DECLARATION

I hereby *declare* that the Thesis entitled **Unsupervised Face Recognition Using One-Shot Learning** which is being submitted to **National Institute of Technology Karnataka, Surathkal**, in partial fulfilment for the requirements of the award of degree of **Master of Technology** in **Computer Science and Engineering - Information Security** in the department of **Computer Science and Engineering**, is a *bonafide report of the work carried out by me.* The material contained in this report has not been submitted at any other University or Institution for the award of any degree.

16IS18F, RaviRaaja L

....

(Register Number, Name and Signature of the student)
Department of Computer Science and Engineering

Place : NITK, Surathkal
Date :

CERTIFICATE

This is to *certify* that the Thesis entitled **Unsupervised Face Recognition Using One-Shot Learning** submitted by **RaviRaaja L** (Register number: 16IS18F) as the record of the work carried out by him, is *accepted as the P.G Major Project Thesis submission* in partial fulfilment for the requirements of the award of degree of **Master of Technology** in **Computer Science and Engineering - Information Security** in the department of **Computer Science and Engineering** at **National Institute of Technology Karnataka, Surathkal** during the academic year 2017-2018.

....	....
Dr.Jeny Rajan	**Alwyn R. Pais**
Project Guide	**Chairman-DPGC**
Department of CSE,	**Department of CSE,**
NITK Surathkal	**NITK Surathkal**

ACKNOWLEDGEMENT

I take this opportunity to express my deepest gratitude and appreciation to all those who have helped me directly or indirectly towards the successful completion of this project.

I would like to express my gratitude to my guide Dr.Jeny Rajan, Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal for his continuous encouragement and support in carrying out my project work. He has continuously mentored my work and corrected my mistakes, without which it would have been difficult to carry out this project. I would like to thank him for all the help and necessary suggestions that he has always provided me with.

I express my deep gratitude to project coordinator Mr.Mahendra Pratap Singh, Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal for his constant co-operation, support and for providing necessary facilities throughout the M.Tech. program.

I would also like to thank all the supporting staff members of the department of Computer Science and Engineering. Without their timely help and support, it would not have been possible to carry out my project.

Place: Surathkal RaviRaaja L
Date: June 2018

Abstract

The ability to recognize and remember individuals is crucial task and has important implications for the evolution of human social behavior, particularly complex interactions between groups. Recently, the role of mass media in shaping public perceptions at scale has come under increasing scrutiny. Implementation of face recognition at large scale involves in face detection and feature extraction using deep convolutional neural network models for clustering the features. We evaluated the effectiveness of deep convolutional neural network architectures and clustering approaches both qualitative and quantitative metrics. However, the conventional softmax loss of Deep CNNs for the most part does not have potential of segregation. To address this issue modified loss functions are used recent years, for example, large margin softmax loss, large margin cosine loss, center loss. All these improved losses shared same idea; maximizing inter-class variance and minimizing intra class variance. In this project we propose a method to deploy unsupervised face recognition using combination of above improved loss functions and customized Deep CNN training methodology.

Keywords— *Face Recognition, Face Detection, Deep Convolutional Neural Networks, Center Loss, Large Margin Softmax Loss, Large Margin Cosine Loss, Softmax Loss.*

Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Face Recognition is crucial task and has important implications for the evolution of human social behavior, particularly complex interactions between groups. Recently, the role of mass media in shaping public perceptions at scale has come under increasing scrutiny. Notably, many of the questions researchers wish to ask about mass media revolve around the questions of who, or making conclusions based on the identifying unique individuals on screen. The goal of this project is to automatically cluster and recognize unique faces for face verification and recognition where dataset is composed of images from various digital medium with different noise levels, image intensity and contrast. Main objective of this project is to come up with a novel deep convolutional neural network architecture which does on-line training on real time images and also to implement a working model as agile software to perform face verification and face clustering using online training.

1.1 PROBLEM STATEMENT

In the world of face recognition, however, large scale public datasets have been lacking largely due to this factor such as advances in the community remain restricted to Internet giants such as Facebook and Google etc. Facebook [Constine (2017)] and Google [Agarwal (2015)] have implemented supervised face recognition as proprietary tools using massive analytics and private datasets trained on Deep CNNs, tech giants uses existing data on generative models to improve the dataset variance adaptively. In reality, greedy approaches are emphasized on training Deep CNN models. Using very less dataset for each instances of face images and to obtain good recognition accuracy, which is also called as one shot learning problem. In order to do one shot learning with better discriminative impacts, it is necessary to use an objective function which has tendency to discriminate the features to larger extent such that inter class variance is maximized. The problem with available standard dataset includes negligible number of instances of faces to train deep networks. Annotated images with high class imbalance and varying noise levels makes face verification less accurate. Though datasets scarcity problem can be solved by manually adding face images by scrapping the google search results, appended by using Haar cascade approaches [Viola and Jones (2001)], face detection has the problem of translational variance such as face angle exposure. Major issues in this project can be listed as follows designing optimal real time clustering algorithm for grouping similar faces and fine tuning approaches to deploy model as real-time application.

1.2 MOTIVATION

Face images are primary source to understand human behavior can impact the economy of the country. For example, [Denyer (2018)] china has implemented nationwide face recognition system that detects the faces and categorize them according to rules of centralized surveillance system, along side by understanding where people travel, eat, understanding their daily routine patterns helps the government to design better business and industrial policies to improve economy of country. In September 2012, the Internet Archive launched the TV News Archive, a repository of 500,000+ TV news broadcasts aired since 2009 (2017). Closed captions are also provided alongside the video, allowing for richer interactions with the video data. Researchers hope to enable users to ask identity-based queries about this data, such as who appeared on Fox News and discussed immigration the greatest number of times in January. The FBI's Next Generation Identification System uses facial recognition to compare images from crime scenes with a national database of mugshots. Police forces across the United States have been using traditional algorithm-based techniques for several years to predict where crimes are likely to occur. By making face recognition technology which is completely unsupervised and availing it for people as RESTful services will be useful to deploy secure home surveillance system and also useful for government to understand the patterns of people behaviors. The problem is divided into three objectives which are face detection, feature extraction ,face clustering and face verification.

1.3 CONTRIBUTIONS

The major contribution are as follows:

- Deploying deep learning models as agile software is a challenging task. In this project, unsupervised face recognition system using deep convolutional neural networks is deployed with validation accuracy of 98% on real time face images using tensorflow as back end pipeline for production. Implementation of combined discriminative learning objective functions like center loss with softmax [Wen et al. (2016)], triplet loss [Schroff et al. (2015)], Quadruplet loss [Chen et al. (2017)] and margin cosine loss[Wang et al. (2018)] have been implemented to train the model and to produce better discriminative results.

- Implementation of wrapper module using python on MTCNN a novel face detection approach is performed in this project to capture the faces from real time images and align them irrespective of facial orientation and noise levels, given weights for pre-trained MTCNN model. [Xiang and Zhu (2017)]
- Analysis of kmeans++ clustering, DBScan clustering, Guassian clustering models and Rank order clustering with silhoutte scores as inertia.
- Implementation of light weight graph based custom clustering technique has been deployed to solve problem of unsupervised face recognition in large scale.
- Automatic intuitive training of deep convolutional neural network without over fitting the model. [Cogswell et al. (2015)]
- Implementation of data visualization technique t-SNE (Stochastic Neighborhood embedding) to visualize whether embeddings are suitable for clustering and retraining model. [Maaten and Hinton (2008)]

1.4 ORGANIZATION OF THE THESIS

Chapter 1 gives a brief introduction along with formal problem statement in section 1.1 and motivation in section 1.2 and the contributions of this work in section 1.3. Chapter 2 tabulates the literature on research over unsupervised face recognition. Chapter 3 includes the proposed solution ,various deep CNN architecture experimented and Clustering approaches used. Chapter 4 Includes strategy on deployment of Deep CNN for unsupervised face recognition. Chapter 5 includes conclusion and further research ideas to extend the research.

CHAPTER 2

LITURATURE SURVEY

2.1 FACE CLUSTERING

The below table 2.1 lists prior work on face clustering with the face representation and clustering algorithm used as well as the largest dataset size used in terms of face images and the number of subjects. [Ho et al. (2003)] developed variations on spectral clustering by computing the probability that two face images are of the same object. [Zhao et al. (2006)] clustered personal photograph collections by combining a variety of contextual information including time based clustering, and the probability of faces of certain people to appear together in images. They also used identity estimates from a Hidden Markov model, and hierarchical clustering results based on body detection.[Cui et al. (2007)] developed a semi-automatic tool for annotating photographs, which employs clustering as an initial method for organizing photographs. They extracted features from detected faces and color and texture features were extracted, and then spectral clustering was performed and evaluated on a dataset consisting of 400 photographs of 5 subjects. [Tian et al. (2007)] refined this approach by incorporating a probabilistic clustering model which used a junk class that allowed the algorithm to discard clusters that do not have tightly distributed samples. [Zhu et al. (2011)] developed a dissimilarity measure based on two faces being in each others nearest neighbor lists, and perform hierarchical clustering based on the resulting rank-order distance. However, this clustering method was evaluated on small datasets (approximately 1,300 face images).To the best of our knowledge,this study is the first to attempt to do unsupervised identity clustering on a large scale (infinite images). From the literature review, it is clear that most studies used pre-extracted faces from standard database rather than using real time data in my case, model is trained online using clustering output on real time data which follow various noise levels. Moreover, the 128-byte feature extractors we use are significantly smaller than the features employed in other studies allowing for higher efficiency and scalability.

Publication	Features	Clustering Methods	#Face images	#Subjects
[Ho et al. (2003)]	Gradient and Pixel Intensity Features	Spectral Clustering	1386	66
[Zhao et al. (2006)]	2DHMM + contextual	Hierarchical Clustering	1500	8
[Cui et al. (2007)]	LBP, clothing color + texture	Spectral Clustering	400	5
[Tian et al. (2007)]	Image + Contextual	Partial Clustering	1147	34
[Zhu et al. (2011)]	Learning based descriptor	Rank order Clustering	1322	53
Vidal and Favaro [Li et al. (2015)] [Zhang et al. (2014)]	Joint subspace learning	Similarity based clustering	2432	38

Table 2.1: Papers on Face clustering

2.2 SUPERVISED FACE DETECTION AND RECOGNITION

Semi supervised learning approaches is used to solve problem at first stage by obtaining embeddings from Deep CNN, so it is necessary to analyze the supervised approaches which uses deep convolutional neural networks which works well for face classification. Thus the experiments can be performed from selected optimal model. Below table. 2.2 list the prior work on face image classification. In 2015, Google released technique to train models for classification problems on discriminating features,called FaceNet. FaceNet follows Inception V3[Szegedy et al. (2015)] deep neural network architecture and have trained using the technique called as siamese network [Koch et al. (2015)], where two networks are used for training images which are identical and also share same weights. Objective function used is triplet loss where triplets of input images are used for training. Usage of this objective function results in extending distance or similarity between the two faces, such that if given two faces are different distance is increased and else reduced 4.10. Inception architecture is highly preferred to obtain embeddings for clustering because using multiple features from multiple filters improve the performance of the network. Other than that, there is another fact that computational complexity on making convolution operations is less costly, makes the inception architecture better than others. All the architectures prior to inception, performed convolution on the spatial and channel wise domain together. By performing the 1x1 convolution, the inception block is doing cross-channel correlations, ignoring the spatial dimensions. This is followed by cross-spatial and cross-channel correlations via the 3x3 and 5x5 filters. Thus inception based models are decided to be used to solve the problem of unsupervised face recognition. Same year Oxford visual Geometry Group performed experiments on conventional CNN model with triplet loss this results in better classification accuracy than facenet. In 2016 CMU have implemented preprocessing techniques to feed into same inception variants like architecture [Amos et al. (2016)] and trained in distributed way to make the weights used for deploying face recognition in mobile applications in supervised way. Thus Google and Oxford has access to extensive amount of private face dataset, training and validation accuracy of the model was more compared to Open face [Amos et al. (2016)].[Chen et al. (2017)] introduced objective function that made use of four images such as anchor, positive, negative and more-negative to train the DCNN architecture similar to triplet loss but this process have very high time complexity compared to training a model with triplet loss. Behalf of face recognition and face clustering, major problem also lays in face detection because conventional face detection approaches performs well only on standard database. For real time data set [Zhang et al. (2016)] introduced new deep convolutional neural network approach named as multi task CNN which is considered as state of art face detection algorithm, thus in this project some more preprocessing techniques are added like usage of laplacian filter to remove blur.

Publication	Feature Extraction Methods	Objective function	AUC
Google Brain [Schroff et al. (2015)]	Siamese Network and Inception V3	Triplet Loss	0.98
Deep Face [Parkhi et al. (2015)]	Alex Net without Local Response Normalization and other variations [Simonyan and Zisserman (2014)]	Triplet loss	0.99
[Amos et al. (2016)]	Affine Transformation and DCNN	Triplet Loss	0.973
[Chen et al. (2017)]	Margin-based hard negative mining	Quadruplet Loss	0.98
[Zadeh et al. (2017)]	Preprocessing and Correlation Network	Land mark detection approach	NIL
[Zhang et al. (2016)]	Multi Task cascade convolutional Neural Network	Face Detection and orientation	NIL

Table 2.2: Papers on Supervised Face Classification and Land mark detection

CHAPTER 3

PROPOSED SOLUTION

3.1 OVERVIEW

Semi supervised learning is the technique opted as first step to solve the problem of unsupervised face recognition. Semi supervised learning with respect to face recognition includes steps such as training standard face images with annotations. As mentioned earlier traditional softmax loss function will not make much impact in discriminating th facial features, usage of appropriate objective function and on training real-time cropped face images using deep convolutional neural network, which is also called as supervised learning. Dataset is trained until optimized validation accuracy is obtained. Cropped face images irrespective of its image features when passed into trained deep CNN to obtain embeddings and clustering those embeddings outputted from the Deep CNNs resulted as labels, with which respect to labels medoid is computed to represent the entire unique cluster. Face verification is performed by a forward pass of cropped face which has to be verified into Deep CNN to obtain embeddings and compare the confidence with all medoids (minimal thresholded confidence value obtained previously). Thus to improve the embeddings patterns to follow principle of high inter class variance and low intra class variance, deep CNN is re-trained in a supervised mode with obtained labels to get model tuned and to produce densely packed embeddings.

The widely used softmax loss in the training process often bring large intra-class variations, and feature normalization is only exploited in the testing process to compute the pair similarities. To bridge the gap, we impose the modified intra-class cosine similarity between the features and weight vectors, which is deployed with combination of softmax loss larger than a margin in the training step, and extend it from four aspects. First, we explored the effect of a hard sample mining strategy. To alleviate the human labor of adjusting the margin hyper-parameter, a self-adaptive margin updating strategy is proposed. Then, a normalized version is given to take full advantage of the cosine similarity constraint. Furthermore, we enhance the former constraint to force the intra-class cosine similarity larger than the mean inter-class cosine similarity with a margin in the exponential feature projection space. Cosine similarity and usage of center loss resulted in almost same accuracy and by comparing memory constrains center loss with softmax loss is used to trained the final model.

3.1.1 Reason for using Deep CNN

Recent year Deep CNNs are giving best results for the problem of image classification. Thus image-net challenge is considered as standard for ranking deep neural network model with respect to Top 5 classification accuracy. [Krizhevsky et al. (2012)] Deep convolutional neural networks are used for face recognition because faces image have high intraclass correlation. Intraclass correlation coefficient (ICC) [Koch (1982)] is an inferential statistic that can be used when quantitative measurements are made on units that are organized into groups. It describes how strongly units in the same group resemble each other. While it is viewed as a type of correlation, unlike most other correlation measures it operates on data structured as groups, rather than data structured as paired observations. In other words the Intraclass Correlation (ICC) assesses rating reliability by comparing the variability of different ratings of the same subject to the total variation across all ratings and all subjects.

There are many standard DCNN architecture available to perform classification problem on images, thus with respect to results of image-net challenge 2017,Squeeze and Excitation blocks are suggested to be used at deep layers of the network. Thus experiments are performed on different neural network architectures like ResNet50 [He et al. (2016)], VGG16 [Simonyan and Zisserman (2014)],Inception V3 [Szegedy et al. (2015)], Xception [Chollet (2016)] etc., because of data scarcity and available dataset has different noise distribution ,intensity and contrast variation. The best results are obtained from usage if inception-Resnet v2 [Szegedy et al. (2016)].

3.2 DEEP CNN ARCHITECTURE

3.2.1 Pure Inception Blocks

Older Inception models used to be trained in a partitioned manner, where each replica was partitioned into a multiple sub-networks in order to be able to fit the whole model in memory. However, the Inception architecture is highly tunable, meaning that there are a lot of possible changes to the number of filters in the various layers that do not affect the quality of the fully trained network. In order to optimize the training speed, we used to tune the layer sizes carefully in order to balance the computation between the various model sub-networks. In contrast, with the introduction of Tensor-Flow ,current models can be trained without partitioning the replicas. This is enabled in part by recent optimizations of memory used by back-propagation, achieved by carefully considering what tensors are needed for gradient computation and structuring the computation to reduce the number of such tensors. We have been relatively conservative about changing the architectural choices and restricted our experiments to varying isolated network components while keeping the rest of the network stable. Not simplifying earlier choices resulted in networks that looked more complicated that they needed to be. In our newer experiments, for Inception-v4 it is decided to shed this unnecessary baggage and made uniform choices for the Inception blocks for each grid size. Please refer to Fig. 3.7 for the large scale structure of the Inception-v4 network and refer Fig. 3.1, 3.2, 3.3, 3.4, 3.5 and 3.6 for the detailed structure of its components. All the convolutions not marked with "V" in the figures are same-padded meaning that their output grid matches the size of their input. Convolutions marked with "V" are valid padded, meaning that input patch of each unit is fully contained in the previous layer and the grid size of the output activation map is reduced accordingly.[Szegedy et al. (2016)]

3.2.2 Residual Inception Blocks

For the residual versions of the Inception networks, Here cheaper Inception blocks than the original Inception is used. Each Inception block is followed by filter-expansion layer (1x1 convolution without activation) which is used for scaling up the dimensionality of the filter bank before the addition to match the depth of the input. This is needed to compensate for the dimensionality reduction induced by the Inception block.

we have tried several versions of the residual version of Inception. Only two of them are detailed here. The first one "Inception-ResNet-v1" roughly the computational cost of Inception-v3, while "Inception-ResNet-v2" matches the raw cost of the newly introduced Inception-v4 network. See Figure. 3.13 for the large scale structure of both variants. (However, the step time of Inception-v4 proved to be significantly slower in practice, probably due to the larger number of layers.) Another small technical difference between our residual and non-residual Inception variants is that in the case of Inception-ResNet, we used batch-normalization only on top of the traditional layers, but not on top of the summations. It is reasonable to expect that a thorough use of batch-normalization should be advantageous, but we wanted to keep each model replica trainable on a single GPU. It turned out that the memory footprint of layers with large activation size was consuming disproportionate amount of GPU memory.

By omitting the batch-normalization on top of those layers, enable to increase the overall number of Inception blocks substantially. we hope that with better utilization of computing resources, making this trade-off will become unnecessary.[Szegedy et al. (2016)]

3.2.3 Scaling Residual blocks

It is found that if the number of filters exceeded 1000, the residual variants started to exhibit instabilities and the network has just died early in the training, meaning that the last layer before the average pooling started to produce only zeros after a few tens of thousands of iterations. This could not be prevented, neither by lowering the learning rate, nor by adding an extra batch-normalization to this layer. It is found that scaling down the residuals before adding them to the previous layer activation seemed to stabilize the training. In general we picked some scaling factors between 0.1 and 0.3 to scale the residuals before their being added to the accumulated layer activations refer Fig. 3.18. A similar instability was observed in the case of very deep residual networks specifically VGG16 and VGG19, they suggested a two-phase training where the first "warm-up" phase is done with very low learning rate, followed by a second phase with high learning rate. It is found that if the number of filters is very high, then even a very low (0.00001) learning rate is not sufficient to cope with the instabilities and the training with high learning rate had a chance to destroy its effects. It is found it much more reliable to just scale the residuals. Even where the scaling was not strictly necessary, it never seemed to harm the final accuracy, but it helped to stabilize the training.

3.2.4 Loss Functions

As mentioned earlier softmax is not suitable for obtaining discriminative features, different loss functions are experimented with inception-resnet-v2 architecture as follows,

General Softmax Loss

The softmax function or normalized exponential function, is a generalization of the logistic function that squashes a k dimensional vector z of arbitrary real values to k dimensional vector $\sigma(z)$ of real values, where each entry is in range (0,1) and all entries adds up to 1 [Bishop (2006)], the function is given by

$$L_S = -\sum_{i=1}^{m} log(e^{W_{y_i}^{T} x_i + b_{y_i}} / \sum_{j=1}^{n} e^{W_j^{T} x_i + b_j}) \tag{3.1}$$

Triplet Loss

The embedding is represented by $f(x)\varepsilon\mathbb{R}^d$. It embeds an image x into a d-dimensional Euclidean space. Additionally, I constrain this embedding to live on d-dimensional hypersphere, *i.e.* $\|f(x)\|^2 = 1$. This loss is motivated in the context of nearest neighbor classification. Here it is necessary to ensure that an image x_i^a *(anchor)* of specific person is closer to all other images x_i^p *positive* of the same person than it is to any image x_i^n *(negative)* of any other person [Schroff et al. (2015)]. Below represents in terms of equations

$$\|f(x_i{}^a) - f(x_i{}^p)\|_2^2 + \alpha < \|f(x_i{}^a) - f(x_i{}^n)\|_2^2, \forall (f(x_i^a), f(x_i^p), f(x_i^n))\varepsilon\tau \tag{3.2}$$

where α is the margin that is enforced between positive and negative pairs. τ is the set of all possible triplets in the training set and has cardinality n. The loss that is being minimized in then

$$L = \sum_{i}^{N} [\|f(x_i{}^a) - f(x_i{}^p)\|_2^2 + \alpha - \|f(x_i{}^a) - f(x_i{}^n)\|_2^2]_+ \tag{3.3}$$

Generating all possible triplets would result in many triplets that are easily satisfied. These triplets would not contribute to the training and results in slower convergence, as they would still be passed through the network. It is crucial to select hard triplets, that are active and can therefore contribute to improving the model. The following section talks about the different approaches we use for the triplet selection.

- Generate triplets offline every n steps, using the most recent network checkpoint and computing the argmin and argmax on a subset of the data.

- Generate triplets online. This can be done by selecting the hard positive/negative exemplars from within a mini-batch.

Large Margin Cosine Loss

Large margin cosine loss is similar to the softmax loss with one difference in computing the exponential value of $e^{f_{y_i}}$, where

$$f_j = W_j^T x = \left\|W_j\right\| \|x\| cos\Theta_j \tag{3.4}$$

where Θ_j is angle between W_j and x. This formula suggests that both norm and angle of vectors contribute to posterior probability. To have the norm of W as invariable we can use L2 normalization. In testing stage the face recognition score of a testing face pair is usually calculated according to cosine similarity between the two feature vectors. This suggests that the norm of feature vector x is not contributing to the scoring function. Thus, in the training stage, it assumed as $\|x\| = s$. Consequently, the posterior probability merely relies on cosine of angle.

Considering a scenario of binary-classes for example, let θ_i denote the angle between the learned feature vector and the weight vector of Class C_i (i = 1,2). The Normalized softmax loss forces $\cos(\theta_1) > \cos(\theta_2)$ for C_i, and similarly for C_2 , so that features from different classes are correctly classified. To develop a large margin classifier, we further require $\cos(\theta_1) - m > \cos(\theta_2)$ and $\cos(\theta_2) - m > \cos(\theta_1)$ where $m \geq 0$ is a fixed parameter introduced to control the magnitude of the cosine margin. Since $\cos(\theta_i)$ - m is lower than $\cos(\theta_i)$, the constraint is more stringent for classification. The above analysis can be well generalized to the scenario of multi-classes. Therefore, the altered loss reinforces the discrimination of learned features by encouraging an extra margin in the cosine space [Wang et al. (2018)]. LMC can be finally defined as below

$$L_{lmc} = 1/N \sum_{i}^{n} -log((e^{s(\cos(\theta_{y_i,i})-m)})/(e^{s(\cos(\theta_{y_i,i})-m)}) + \sum_{j\neq y_i}^{n} e^{s(\cos(\theta_{j,i}))}) \tag{3.5}$$

Center Loss with Softmax Loss

$$L_c = 1/2 \sum_{i=1}^{m} \left\|x_i - c_{y_i}\right\|_2^2 \tag{3.6}$$

where $c_{y_i}\varepsilon\mathbb{R}^d$,.The above formulation effectively characterizes the intra-class variations. Ideally, the c_{y_i} should be updated as the deep features changed. In other

Networks	k	l	m	n
Inception-v4	192	224	256	384
Inception-ResNet-v1	192	192	256	384
Inception-ResNet-v2	256	256	384	384

Table 3.1: The number of filters of the Reduction-A module for the three Inception variants presented in this thesis. The four numbers in the colums of the paper parametrize the four convolutions of Figure 3.5

words, it is required to take the entire training set into account and average the features of every class in each iteration, which is inefficient even impractical. Therefore, the center loss can not be used directly. This is possibly the reason that such a center loss has never been used in CNNs until now. To address this problem, two necessary modifications are made. First, instead of updating the centers with respect to the entire training set, we perform the update based on mini-batch. In each iteration, the centers are computed by averaging the features of the corresponding classes (In this case, some of the centers may not update). Second, to avoid large perturbations caused by few mis-labelled samples, we use a scalar α to control the learning rate of the centers. The gradients of L_c is computed with respect to x_i and update equation of c_{y_i} are computed as follows:

$$\partial L_c / \partial x_i = x_i - c_{y_i} \tag{3.7}$$

$$\Delta c_j = \sum_{i=1}^{m} (\delta(y_i = j).(c_j - x_i))/(1 + \sum_{i=1}^{m} \delta(y_i = j)) \tag{3.8}$$

where δ (*condition = 1*) if the condition is satisfied, and δ (*condition = 0*) if not, α is restricted in [0,1].Adoption of the joint supervision of softmax loss and center loss to train the CNNs for discriminative feature learning. [Wen et al. (2016)] The formulation is given in below equation.

$$L = L_S + \lambda \times L_c \tag{3.9}$$

3.3 CLUSTERING APPROACHES

For face embeddings obtained from trained neural network are consider as input for clustering faces, here i have analyzed five clustering approaches such as Scalable K-means [Bahmani et al. (2012)], GMM [Cui et al. (2007)], Rank order [Ho et al. (2003)], DBSCAN [Schroff et al. (2015)], DBSCAN + Kmeans [Tian et al. (2007)] and Chinese

Whispers clustering [Biemann (2006)]

3.3.1 K-Means

The k-means clustering algorithm alternates between assigning training examples to the nearest centroid and setting centroid to all the average of all assigned examples. Repeat till convergence:

For every i. set

$$c^{(i)} := argmin_j \left\|x^{(i)} - \mu_j\right\|^2 \tag{3.10}$$

For every j. set

$$\mu_j := (\sum_{i=1}^{m} \{c^{(i)} = j\} x^{(i)}) / (\sum_{i=1}^{m} \{c^{(i)} = j\}) \tag{3.11}$$

Centroids are typically initialized randomly, but we instead use k-means++ [Bahmani et al. (2012)] to choose initial centroids

3.3.2 Guassian Mixture Model(GMM)

We model each training example as originating from one of k Gaussian distributions. Formally, specification of the joint distribution will be $p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$, where $z^{(i)}$ is Multinomial ϕ and $x^{(i)}|z^{(i)} = j$. This latent random variable $z^{(i)}$ can take one of k values and identifies the Guassian from which $x^{(i)}$ was drawn. Parameter initialization done using k-means because random initialization performed poorly in comparison.

The expectation-maximization algorithm is used to estimate parameters. The EM algorithm alternates between (E-step) and estimating the values of $z^{(i)}$ and (M-step) updating the parameters of model based on those estimates, formally,

Repeat till convergence

(E-Step) for each i,j, set

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}; \phi, \mu, \sum) \tag{3.12}$$

(M-step) update the parameters:

$$\phi_j := 1/m \sum_{i=1}^{m} w_j^{(i)} \tag{3.13}$$

$$\mu_j := (\sum_{i=1}^{m} w_j^{(i)} x^{(i)}) / \sum_{i=1}^{m} w_j^{(i)} \tag{3.14}$$

$$\sum := (\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T) / (\sum_{i=1}^{m} w_j^{(i)}) \tag{3.15}$$

3.3.3 Density based clustering(DB-SCAN)

The objective of DBSCAN is to cluster together high-density regions and mark low-density regions as noise. The algorithm takes two parameters: ε, a distance parameter, and *minPts*, the minimum number of points required to form a dense region. For each training example *x*, the algorithm then (1) retrieves all *n* points within an radius of *x* ,(2) either adds all *n* points and their respective clusters to a new cluster if $n \geq minPts$ or marks *x* as noise if *n* ¡ *minPts*. Note that a point marked as noise can still be assigned to a future cluster. As a modification, initialization is done as, each training example as being its own unique assignment. Then instead of labeling a point as noise, simply leave its original assignment unchanged. This ensures that every face eventually gets assigned to some unique identity. Moreover, we iteratively test out various epsilon values for optimization and choose the epsilon value which yields the highest silhouette coefficient as a parameter.

3.3.4 Rank Order Clustering

Rank-Order is a form of agglomerative hierarchical clustering, meaning all training examples initially represent different clusters and are iteratively merged together. In this specific implementation, below following distance metric is used:

$$D(a,b) = d(a,b) + d(b,a)/min(O_a(b), O_b(a)) \tag{3.16}$$

where,

$$d(a,b) = \sum_{i=1}^{min(O_a(b),k)} I_b(O_b(f_a(i)), k) \tag{3.17}$$

f_a is the i-th nearest face to a, $O_b(f_a(i))$ gives the rank of face $f_a(i)$ in face b's neighbor list, and $I_b(x,k)$ is 0 if face *x* is in face b's top *k* nearest neighbors ans 1 otherwise.

The algorithm finds the k-nearest neighbors for each face *a*, computes pairwise distances D(a,b) between *a* and its top *k* neighbors, and transitively merges all (a,b) with distances below a given threshold. It repeats these steps until no further pairwise distances fall below this threshold.

3.3.5 DBSCAN + K-means

In this novel algorithm, we run DBSCAN on top of clusters already produced by k-means. In other words, initialization of each training example using the assignments given by k-means. When running k-means, choose an optimal ε to break apart too-large clusters. For each individual cluster generated by k-means, find optimal epsilon (based on silhouette coefficient) to run DB-SCAN on the cluster. Thus, each DBSCAN run is optimized for each individual cluster and we achieve highly refined and distinct clusters. This provides an automated method to approximate the number of identities or clusters (i.e. k) in contrast to k-means.

3.3.6 Chinese Whispers Clustering

Chinese Whispers is a clustering method used in network science named after the famous whispering game. It has major implication in natural language processing. Chinese Whispers is a hard partitioning, randomized, flat clustering (no hierarchical relations between clusters) method. The random property means that running the process on the same network several times can lead to different results, while because of hard partitioning one node can only belong to one cluster at a given moment. The original algorithm is applicable to undirected, weighted and unweighted graphs. Chinese Whispers is time linear which means that it is extremely fast even if the number of nodes and links are very high in the network.[Biemann (2006)] Here the modified version of this clustering algorithm is done. where embeddings from the neural net is gathered and every embedding is considered as node and edge is created for the nodes which are very closer each other measured using summation of least squared euclidean distance and high correlation between every embeddings in dataset.

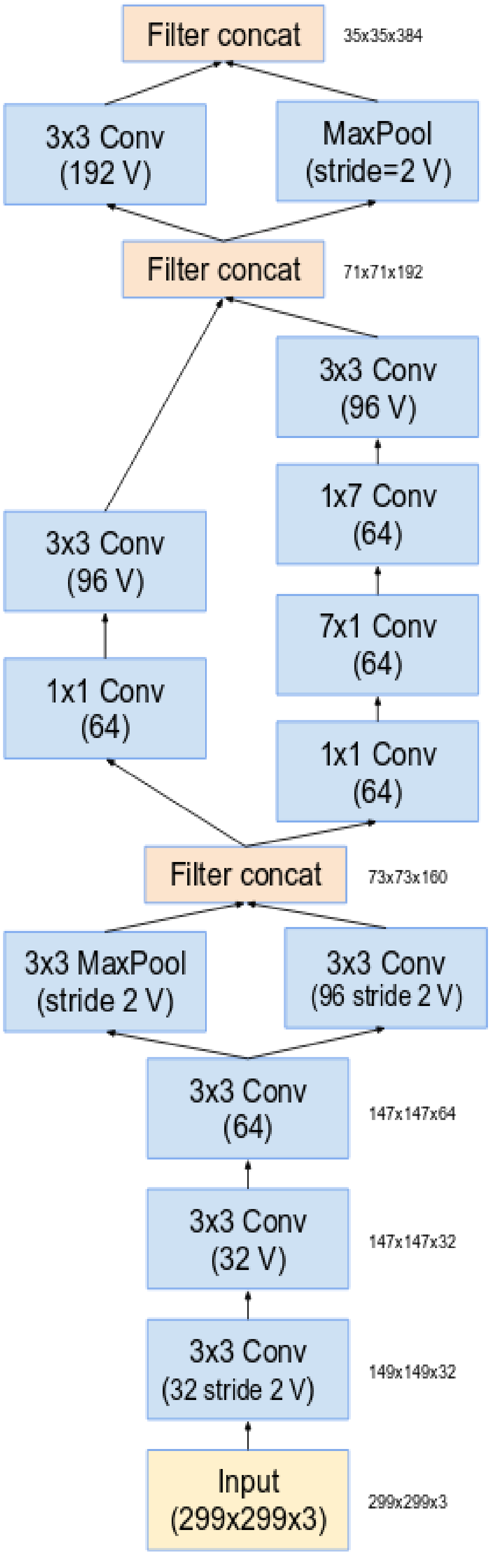

Figure 3.1: The blueprint for stem of the basic Inception-v4 and Inception ResNet-v2 systems. This is the input part of neural network [Szegedy et al. (2017)]. Cf. Figures 3.7 and 3.13

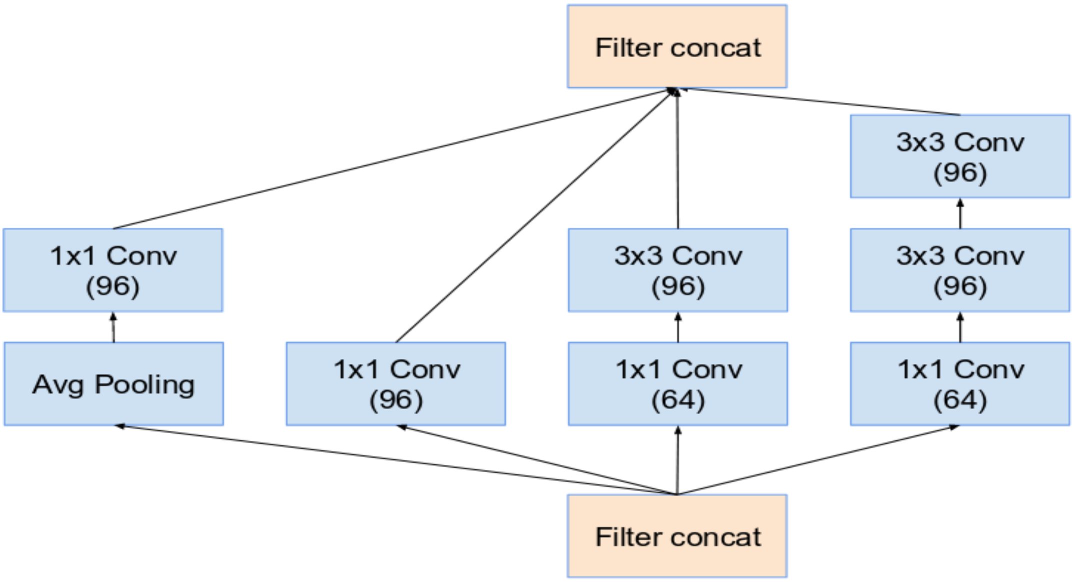

Figure 3.2: The schema for 35 x 35 grid modules of the pure Inception-v4 network. This is the Inception-A block of Figure 3.7. [Szegedy et al. (2017)]

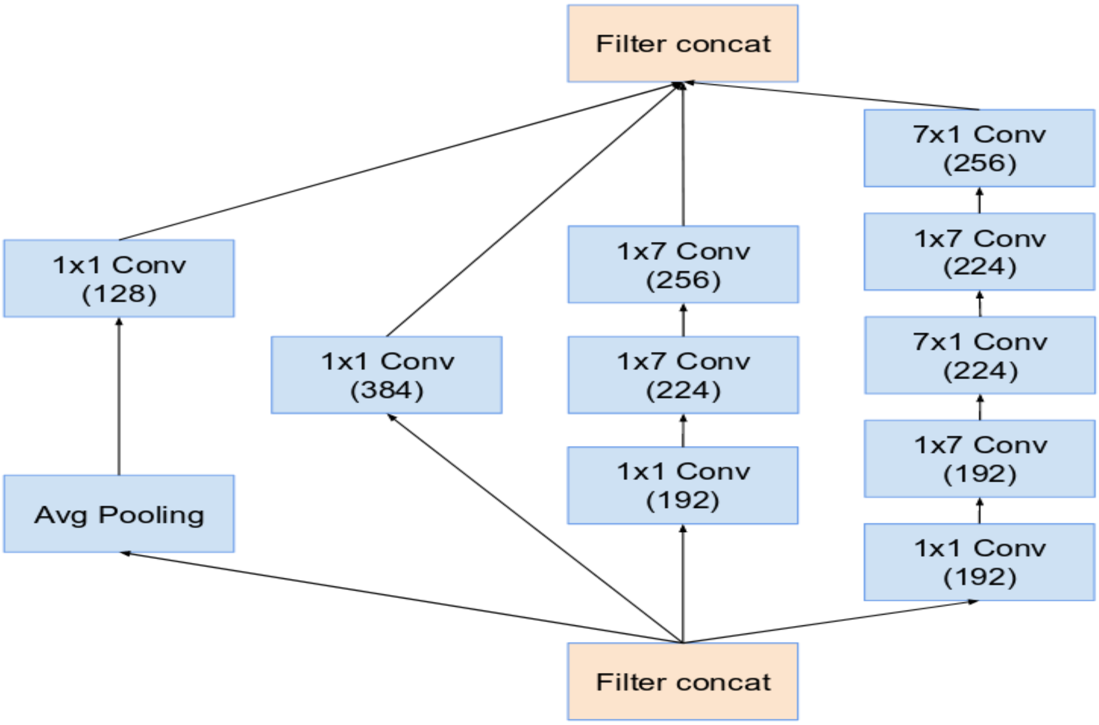

Figure 3.3: The schema for 17x17 grid modules of the pure Inception-v4 network. This is the Inception-B block of Figure 3.7. [Szegedy et al. (2017)]

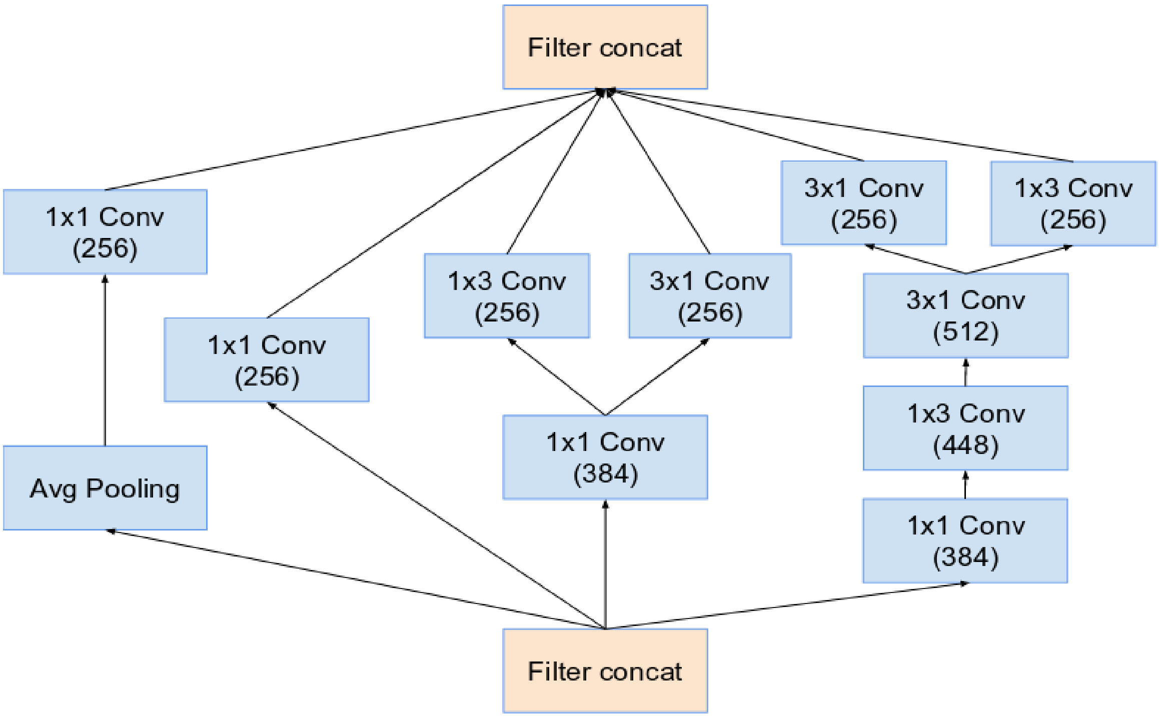

Figure 3.4: The schema for 8x8 grid modules of the pure Inception-v4 network. This is the Inception-C block of Figure 3.7.[Szegedy et al. (2017)]

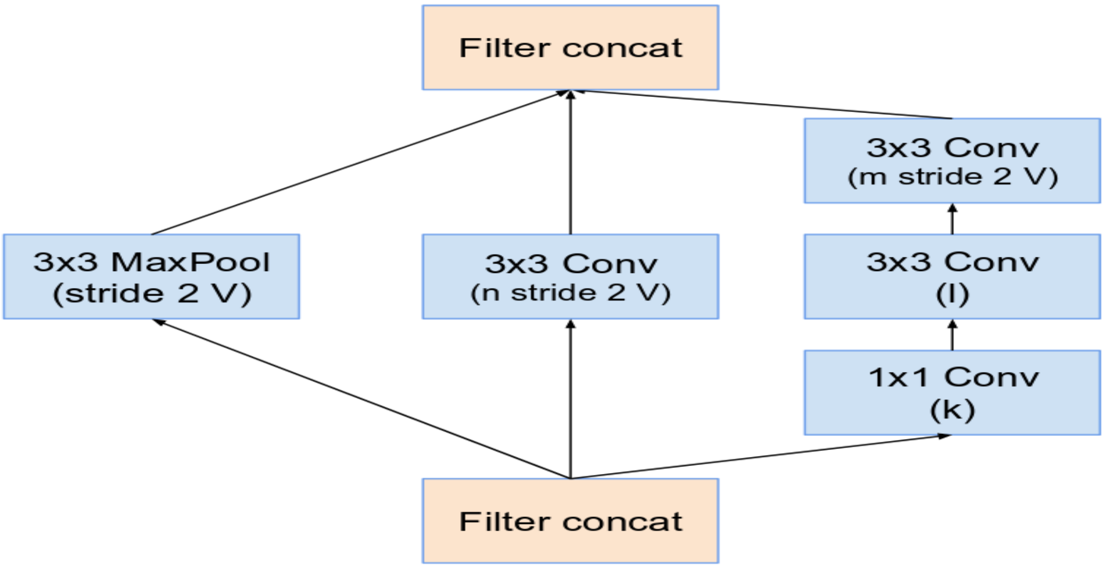

Figure 3.5: The schema for 35x35 to 17x17 reduction module. Different variants of this blocks (with various number of filters) are used in Figure 3.7, and 3.13 in each of the new Inception(-v4, -ResNet-v1, -ResNet-v2) variants presented in this thesis. The k,l,m,n numbers represent filter bank sizes which can be looked up in Table 3.1.[Szegedy et al. (2017)]

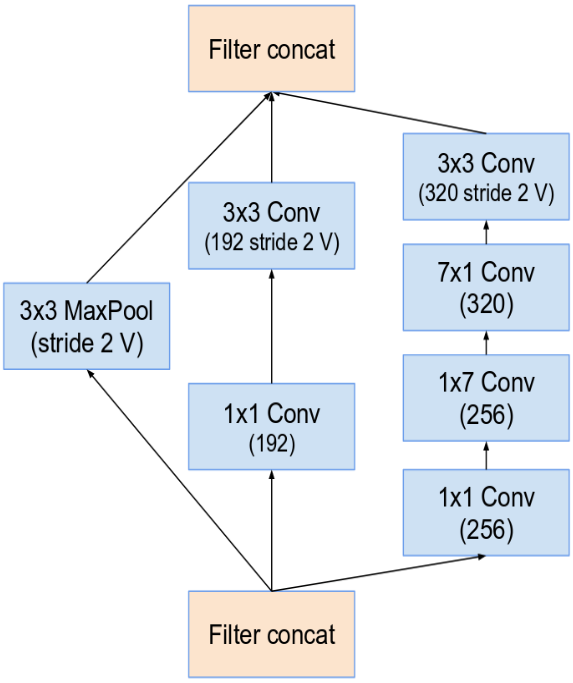

Figure 3.6: The schema for 17x17 to 8x8 grid-reduction module. This is the reduction module used by the pure Inception-v4 network in Figure 3.7. [Szegedy et al. (2017)]

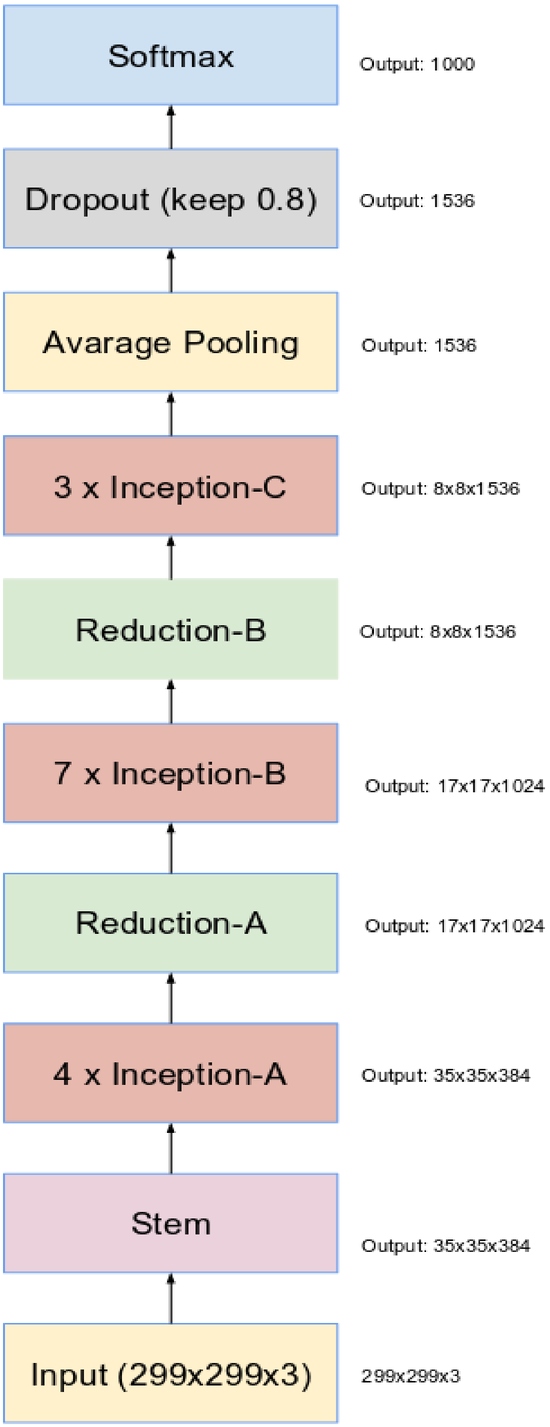

Figure 3.7: The overall schema of the Inception-v4 network. For the detailed modules, please refer to Figures 3.1, 3.2, 3.3, 3.4, 3.5 and 3.6 for the detailed structure of the various components.[Szegedy et al. (2017)]

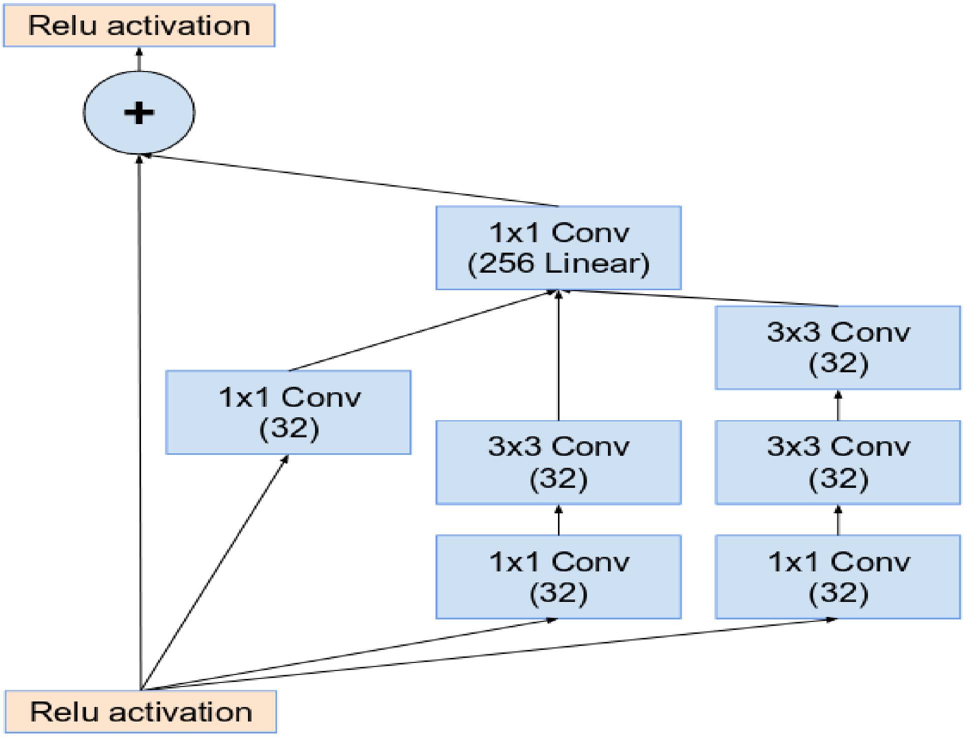

Figure 3.8: The schema for 35x35 grid (Inception-ResNet-A) module of Inception-ResNet-v1 network,[Szegedy et al. (2017)]

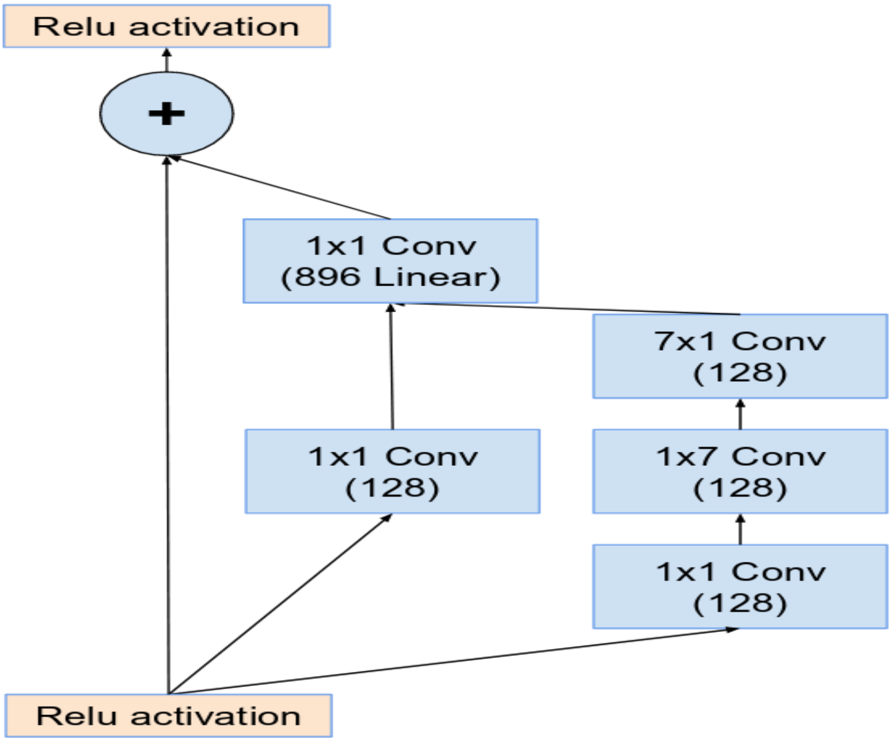

Figure 3.9: The schema for 17x17 grid (Inception-ResNet-B) module of Inception-ResNet-v1 network.[Szegedy et al. (2017)]

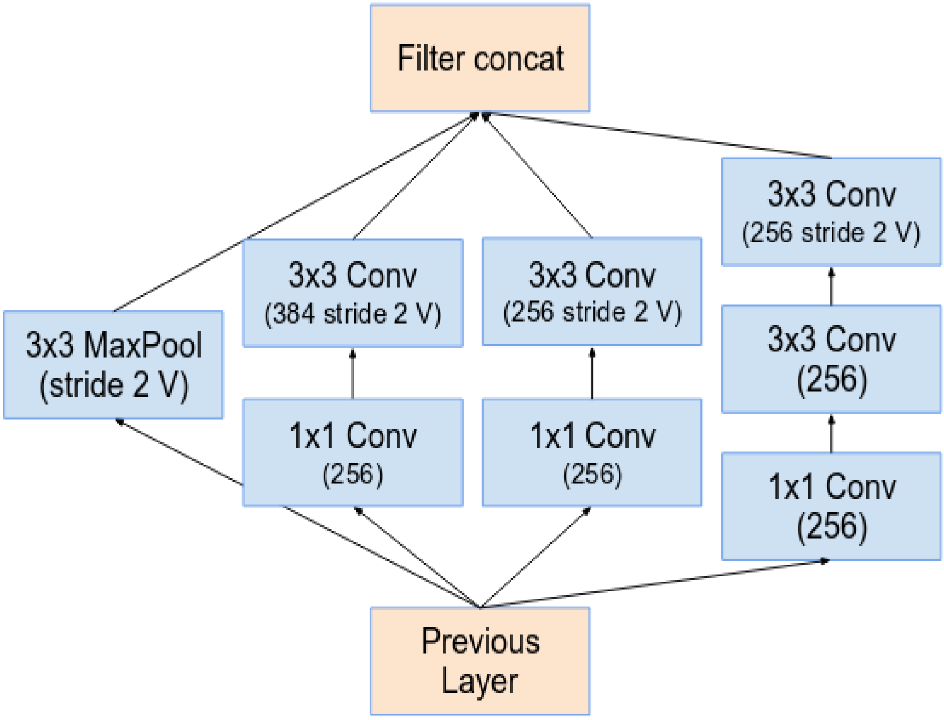

Figure 3.10: “Reduction-B” 17x17 to 8x8 grid-reduction module.This module used by the smaller Inception-ResNet-v1 network in Figure 3.13. [Szegedy et al. (2017)]

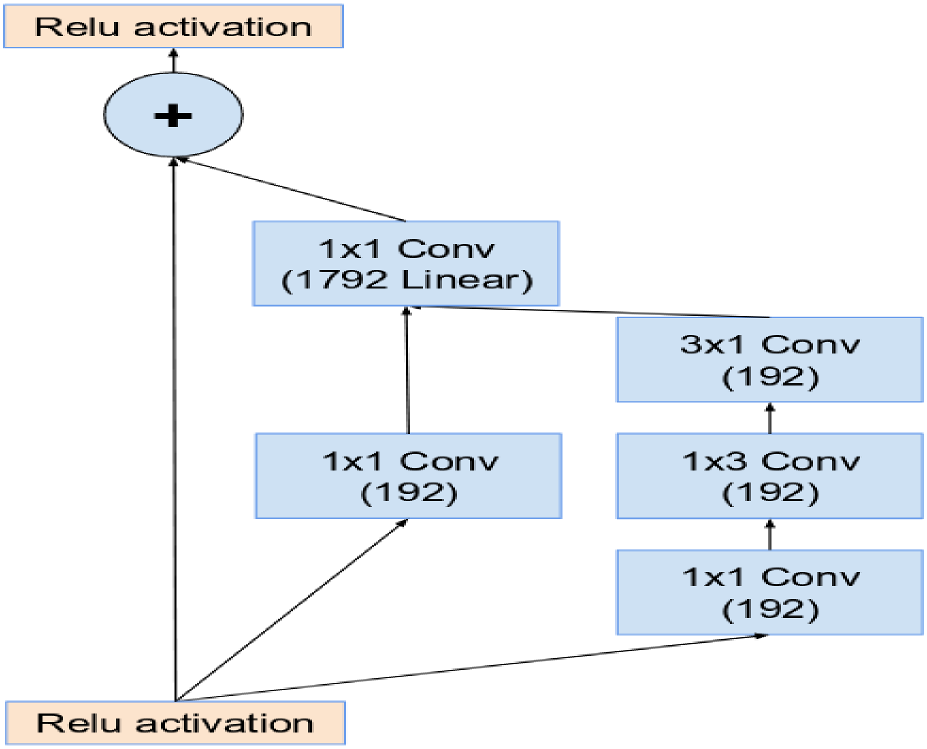

Figure 3.11: The schema for 8x8 grid (Inception-ResNet-C) module of Inception-ResNet-v1 network.[Szegedy et al. (2017)]

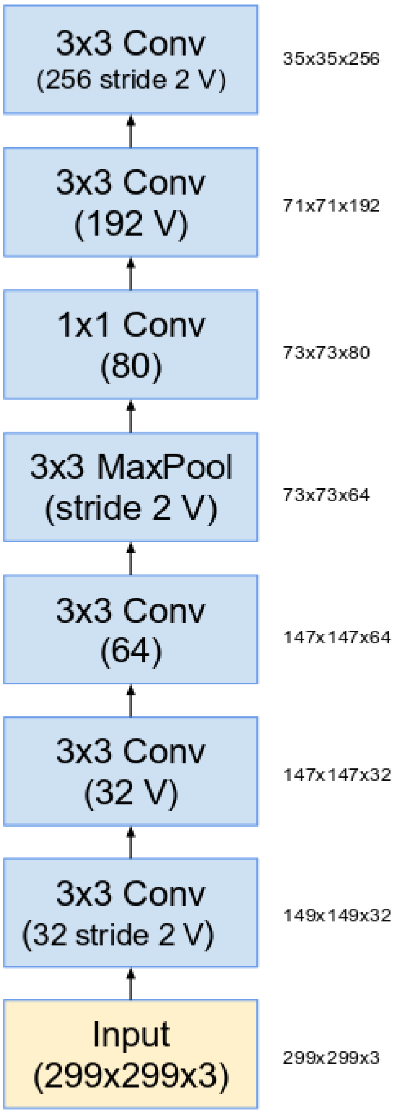

Figure 3.12: The stem of the Inception-ResNet-v1 network.[Szegedy et al. (2017)]

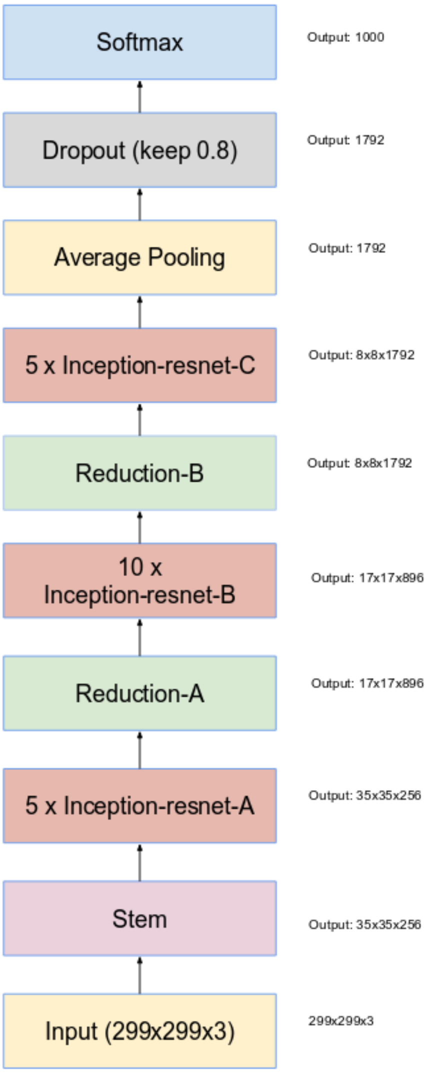

Figure 3.13: Schema for Inception-ResNet-v1 and Inception-ResNet-v2 networks. This schema applies to both networks but the underlying components differ. Inception-ResNet-v1 uses the blocks as described in Figures 3.12, 3.8, 3.5, 3.9, 3.10 and 3.11. Inception-ResNet-v2 uses the blocks as described in Figures 3.1, 3.14, 3.5, 3.15, 3.16 and 3.17. The output sizes in the diagram refer to the activation vector tensor shapes of Inception-ResNet-v1.[Szegedy et al. (2017)]

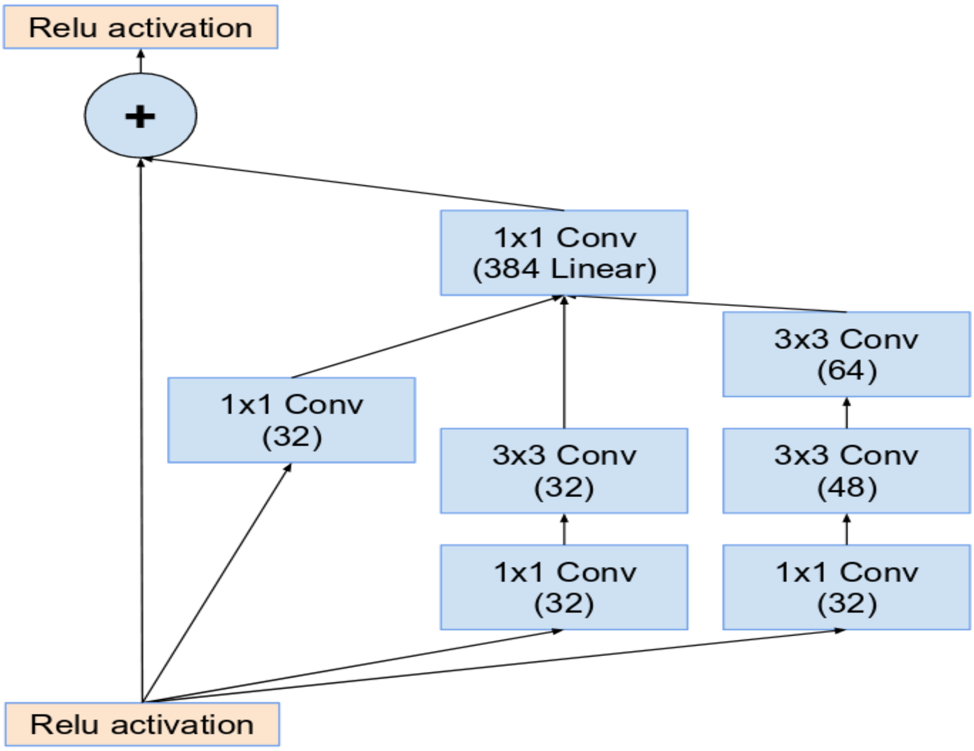

Figure 3.14: The schema for 35x35 grid (Inception-ResNet-A) module of the Inception-ResNet-v2 network.[Szegedy et al. (2017)]

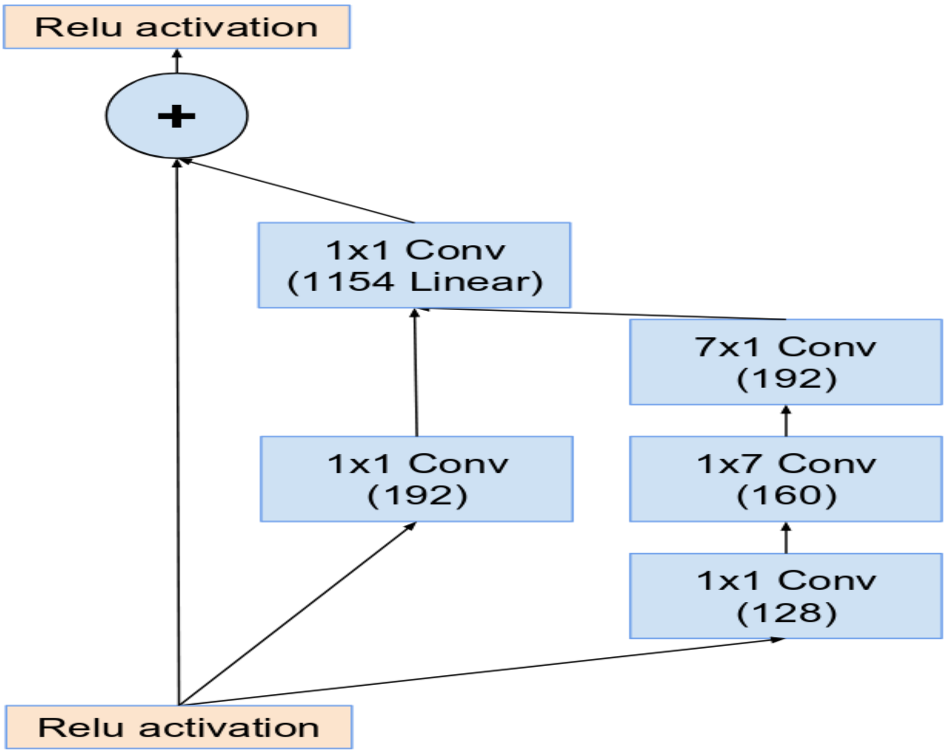

Figure 3.15: The schema for 17x17 grid (Inception-ResNet-B) module of the Inception-ResNet-v2 network.[Szegedy et al. (2017)]

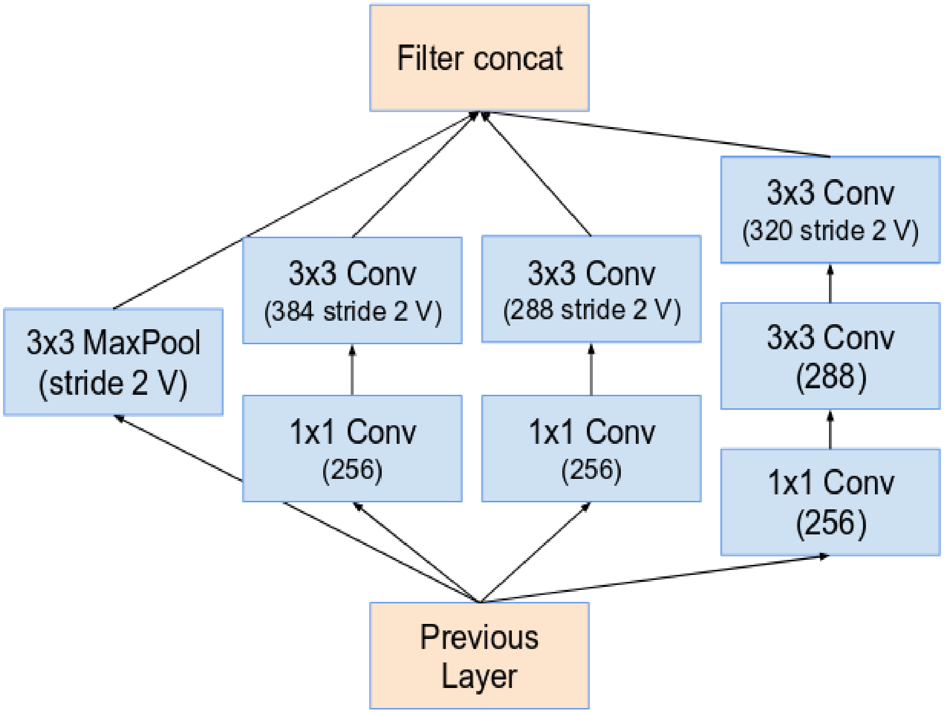

Figure 3.16: The schema for 17x17 to 8x8 grid-reduction module. Reduction-B module used by the wider Inception-ResNet-v1 network in Figure 3.13. [Szegedy et al. (2017)]

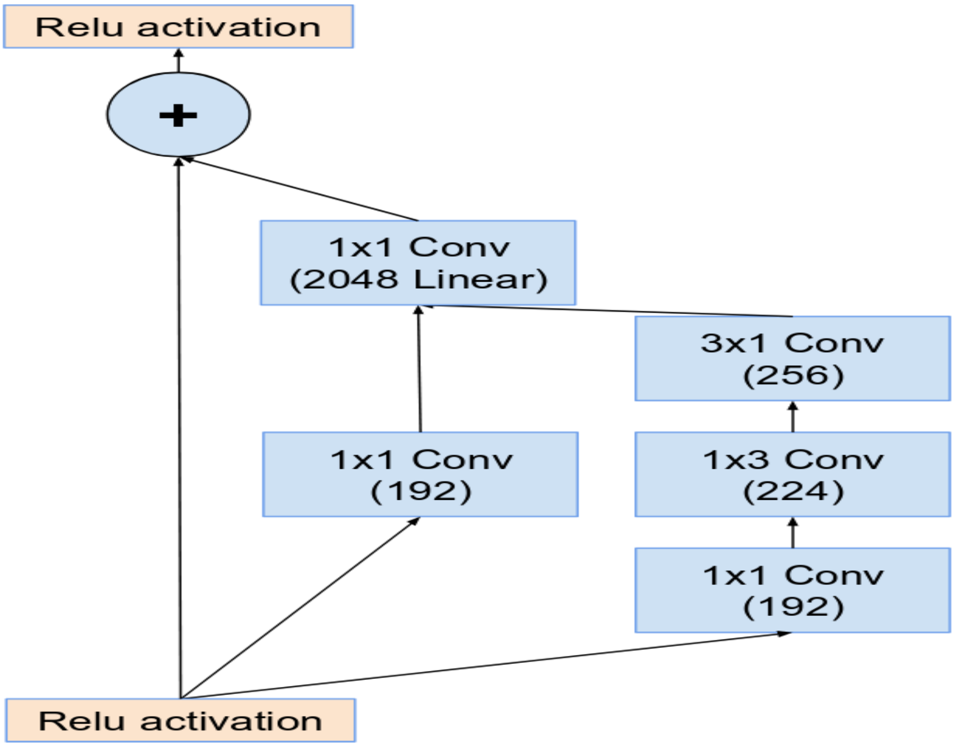

Figure 3.17: The schema for 8x8 grid (Inception-ResNet-C) module of the Inception-ResNet-v2 network.[Szegedy et al. (2017)]

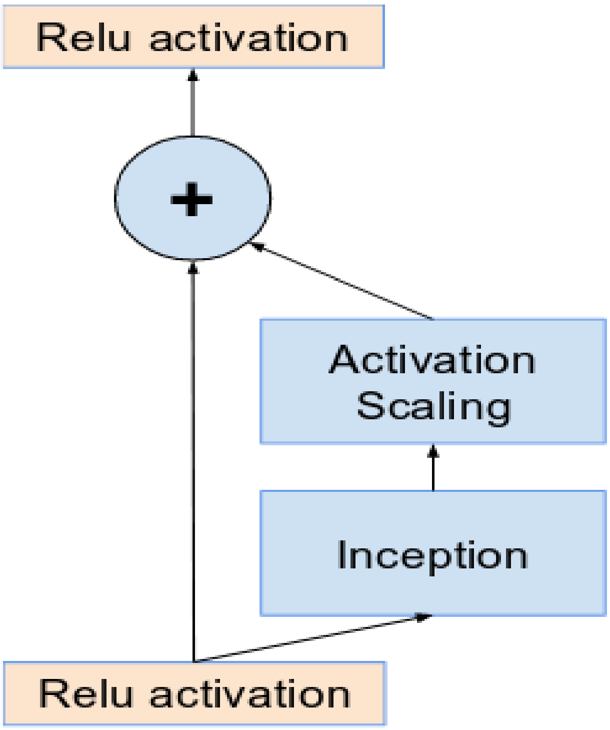

Figure 3.18: The general schema for scaling combined Inception-resnet moduels. I expect that the same idea is useful in the general resnet case, where instead of the Inception block an arbitrary subnetwork is used. The scaling block just scales the last linear activations by a suitable constant, typically around 0.1.[Szegedy et al. (2017)]

CHAPTER 4

IMPLEMENTATION

4.1 SIAMESE TRAINING

Siamese neural network is a class of neural network architectures that contain two or more identical subnetworks. Identical here means they have the same configuration with the same parameters and weights. Parameter updating is mirrored across both subnetworks. Siamese NNs are popular among tasks that involve finding similarity or a relationship between two comparable things. Some examples are paraphrase scoring, where the inputs are two sentences and the output is a score of how similar they are; or signature verification, where figure out whether two signatures are from the same person. Generally, in such tasks, two identical subnetworks are used to process the two inputs, and another module will take their outputs and produce the final output. Siamese architectures are good in these tasks because sharing weights across subnetworks means fewer parameters to train for, which in turn means less data required and less tendency to overfit. Each subnetwork essentially produces a representation of its input. If your inputs are of the same kind, like matching two sentences or matching two pictures, it makes sense to use similar model to process similar inputs. This way you have representation vectors with the same semantics, making them easier to compare. [Bromley (1993)]. To visualize siamese architecture refer Figure.4.8

4.2 DEPLOYMENT OVERVIEW

To understand the complete overview of the unsupervised face recognition project refer fig 4.1. Initial steps involves face detection and preprocessing the face images, training Deep CNN with annotated images, obtaining embeddings from trained neural network, cluster the embeddings and obtain labels and finally, retrain the model for packing embeddings tighter.

4.2.1 Face Detection

Pre-trained multi-task cascaded convolutional neural network (MTCNN) is used to detect and align the faces [Zhang et al. (2014)]. The MTCNN framework uses a cascading structure with three stages of deep convolutional networks that predict face and landmark location. Given an image, it is initially resized to different scales to build an image pyramid, which is given as the input for the three-stage cascaded framework. refer fig 4.2

- In stage 1, a convolutional network called Proposal Network (P-Net) obtains the candidate windows and their bounding box regression vectors. After that, estimated bounding box regression vectors are used to calibrate the candidates. Finally, a non-maximum suppression (NMS) is used to merge highly overlapped candidates.
- In stage 2, all candidates are inputted into a CNN, called Refine Network (R-Net), which further rejects a large number of false candidates, performs calibration with bounding box regression, and NMS candidate merge.
- In stage 3, the network describes the face in more details and outputs five facial landmarks positions along with the aligned and cropped image.

4.2.2 Supervised Face Recognition

Dataset Used

Initially, face images are cropped and passed into deep CNN for training, for training initially standard datasets are used. They are CASIA-WebFace dataset consists of total of 4,53,453 images over 10,575 identities after face detection. Some performance improvement has been seen if the dataset has been filtered before training. The best performing model has been trained initially on the VGGFace2 dataset consisting of 3.3M faces and 9000 classes. Refer figure for Accuracy and loss on training CASIA-webface dataset 4.3, 4.4 and VGG Face2 dataset refer fig 4.5, 4.6. Refer table 4.1 for the training accuracy results.

Algorithm for Training

```
Algorithm 1: Algorithm for Training Deep CNN
    1: Consider positive, anchor and negative images where positive and anchor are
    different instances of same person. Negative and anchor are images of different
    person.
    2: Choosing completely different negative image makes the convergence slower so
    that after N epochs, distance matrix for all images in current mini-batch is
    computed, and least distance with anchor and negative image is chosen. This also
    avoid the false convergence phenomenon of siamese network.
    3: Training is performed by having two instance of Deep CNN model, where anchor
    and positive pairs, anchor and negative pairs are passed for training deep CNN.
    Refer Fig.4.10 results in learning process using triplet loss. Synthetic Gradients are
    used to train model based on only inception resnet v2 architecture. Fig. 4.9 shows
    the new extended architecture of inception resnet v2.
    4: Using pre-trained architecture of inception resnet v2, last layer (containing logits
    (or) vector outputting 128 embeddings) with softmax as activation function is
    removed and output tensor is connected according to the architecture in
    fig. 4.9.where C is Convolutional Layer, P is Pooling Layer, LC is Locally
    Connected Layer and FC is Fully Connected Layer. Newly added layers weights are
    initialized with uniform normal distribution. This network is fine tuned with center
    loss.
    5: Usage of center loss after training with triplet loss made the model to learn the
    facial features and makes convergence of embeddings faster. Visualization of
    embeddings obtained by passing images into trained neural network is performed by
    T-SNE. Refer fig. 4.11
    6: Once visualization of TSNE is satisfactory and can also handles high class
    imbalance problem. The network weights are used for clustering process.
```

4.2.3 Training steps for online training

Inception Resnet v2 is one of the deep network architecture used so that training process on annotated images performed initially will have high time complexity because of the issues in convergence and over fitting of the model during online training. To avoid these issues, for the first time synthetic gradients (SC) has been implemented on CNN to compute next gradient values without actual back propagation step. Synthetic Gradients are introduces and has implications in recurrent neural networks to compute values of gates in problem of language and sequence modeling. In a deep CNN forward pass and backward pass consume lot of time, so to prevent it, a small neural network which are densely connected are used to compute the gradient values. Refer Figure 4.7 for the visual understanding of synthetic gradient computation [Jaderberg et al. (2016)]. Backward pass is also called as update lock. The objective function for the decoupled neural network is norm of squared difference between the predicted gradient and actual back-propagated gradient.

Tuning on real time images is also performed on custom neural network architecture with synthetic gradients, where synthetic gradients are calculated only at the larger filter sized convolutions like 17x17. This assumption is based on calculative analysis on the time complexity involved in training the neural network, refer fig. 4.9.

4.2.4 Clustering

Chinese whispers clustering algorithm is used to gather highly correlated face images. Refer the table 4.2 for the comparative study of clustering algorithm used. Clustering algorithms executed over the training data containing 80k images and the highest-performing algorithm is executed over the testing data. Silhouette coefficient as our primary evaluation metric, where a higher silhouette coefficient is more preferred. This metric is defined for each sample x as

$$s = (b - a)/max(a, b) \tag{4.1}$$

where a is the mean distance between x and all other points assigned to the same centroid, and b is the mean distance between x and all other points of the next nearest cluster. Refer figure 4.12 for the visualization of results after clustering.

Thus the clustering algorithm is a graph based approach where edges are created between highly correlated embeddings obtained from the neural network, but the problem is that, the centroid of the network will not be returned as result of algorithm. In case of using mean of all the embeddings as the centroid of a particular cluster results in less accuracy if data stream into server for a time interval T is less and also because of outliers which will affect the model on the go. To avoid that medoid for every cluster is calculated and stored in database.

Validation

For validating input cropped faces, images are passed into deep CNN architecture and embeddings are obtained. For every instance of existing medoid in database, validation set embeddings are correlated, only correlation with more than threshold t (optimal threshold found by experimentation is 0.8) will be assigned to that corresponding cluster in database, else exported embeddings are stored in separate directory until time T to get involved in clustering algorithm again.

T-SNE

A popular method for exploring high-dimensional data is something called t-SNE, introduced by van der Maaten and Hinton [Maaten and Hinton (2008)] . The technique has become widespread in the field of machine learning, since it has an almost magical ability to create compelling two-dimensonal "maps" from data with hundreds or even thousands of dimensions. Tsne is used to visualize 128 dimensional vector obtained with labels as results of clustering.

Accuracy	Training dataset	Architecture
0.9905	CASIA-WebFace	Inception ResNet v2
0.9965	VGGFace2	Inception ResNet v2

Table 4.1: Accuracy on Public available datasets

Accuracy	Test Accuracy	Algorithm	Tested On	#Samples
0.98	0.96	Chinese Whispers Clustering	Real-Time Dataset(IDrive Inc.)	80k
0.84	0.65	Scalable K means ++ Clustering	Real-Time Dataset	50K
0.98	0.94	Scalable K means ++ Clustering	CASIA and VGG Face2	80k
0.89	NIL	GMM Clustering	VGG Face2	3.3M
0.61	NIL	DBSCAN	VGG Face2	3.3M

Table 4.2: Clustering Accuracy

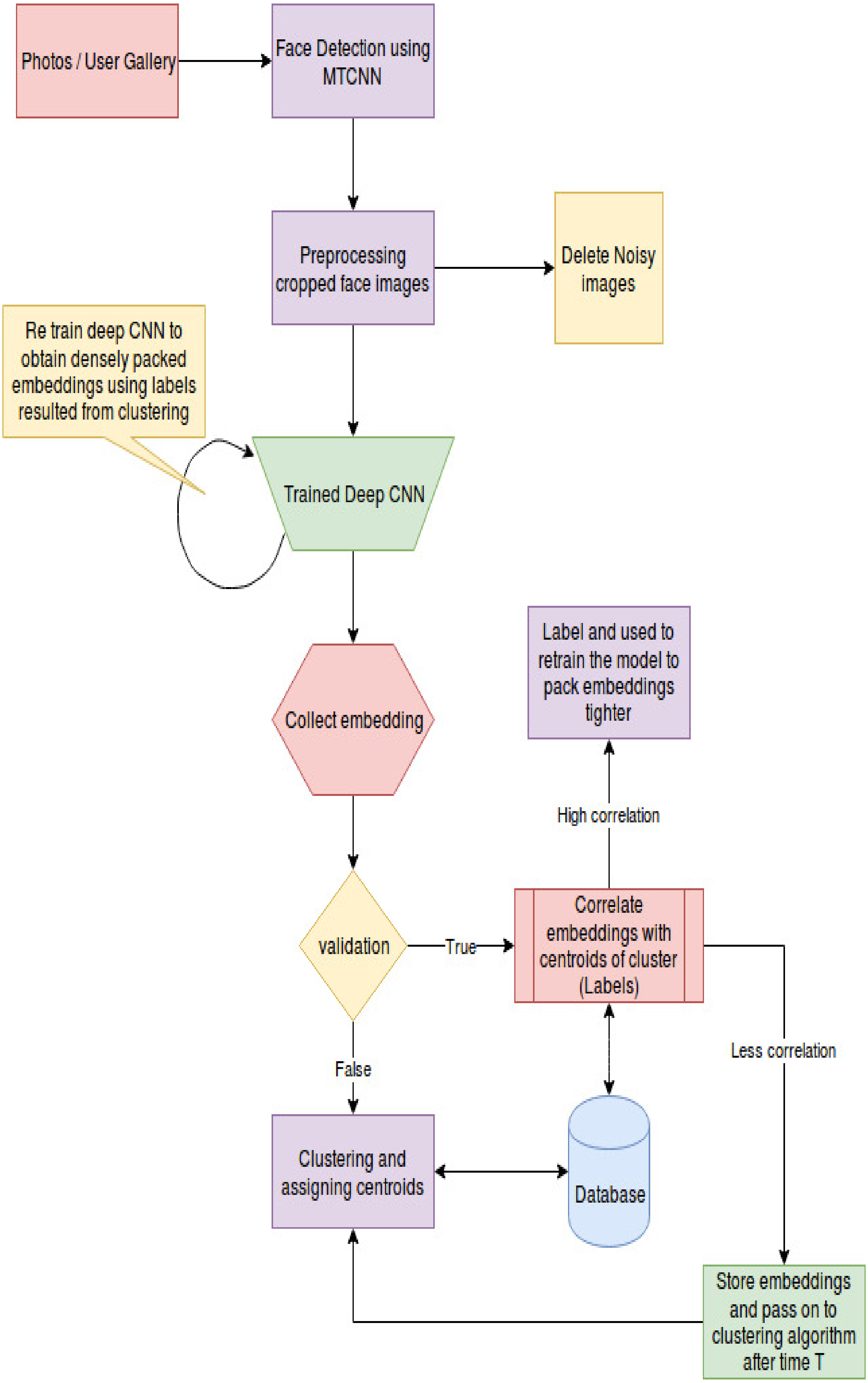
Photos / User Gallery
Face Detection using MTCNN
Preprocessing cropped face images
Delete Noisy images
Re train deep CNN to obtain densely packed embeddings using labels resulted from clustering
Trained Deep CNN
Collect embedding
Label and used to retrain the model to pack embeddings tighter
High correlation
validation
True
Correlate embeddings with centroids of cluster (Labels)
Less correlation
False
Clustering and assigning centroids
Database
Store embeddings and pass on to clustering algorithm after time T

Figure 4.1: Overview of the Unsupervised face recognition

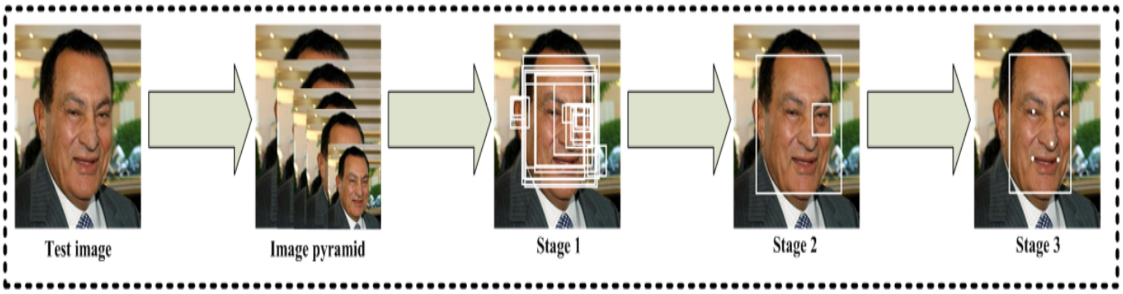

Figure 4.2: Overview of how MTCNN works

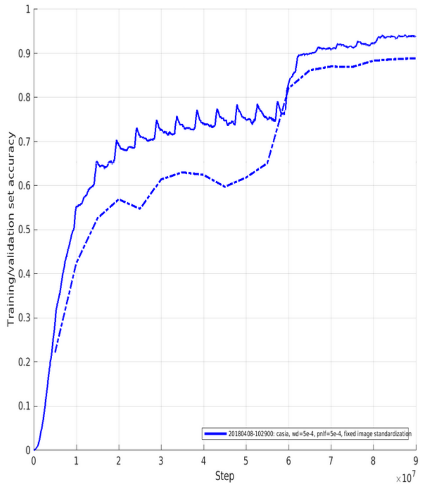

Figure 4.3: Training and Validation Accuracy on CASIA-webface Dataset

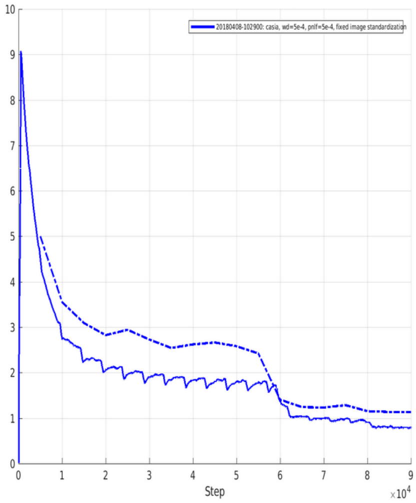

Figure 4.4: Training and Validation Loss on CASIA-webface Dataset

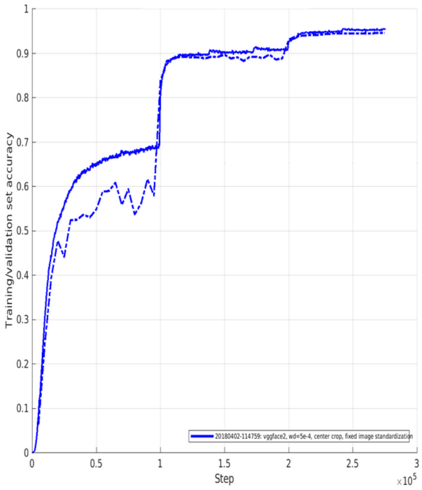

Figure 4.5: Training and Validation Accuracy on VGG-face2 Dataset

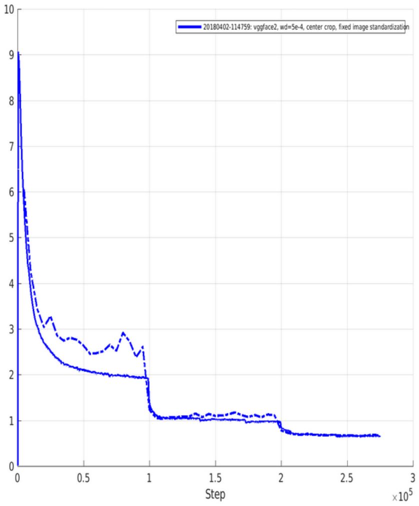

Figure 4.6: Training and Validation Loss on VGG-face2 Dataset

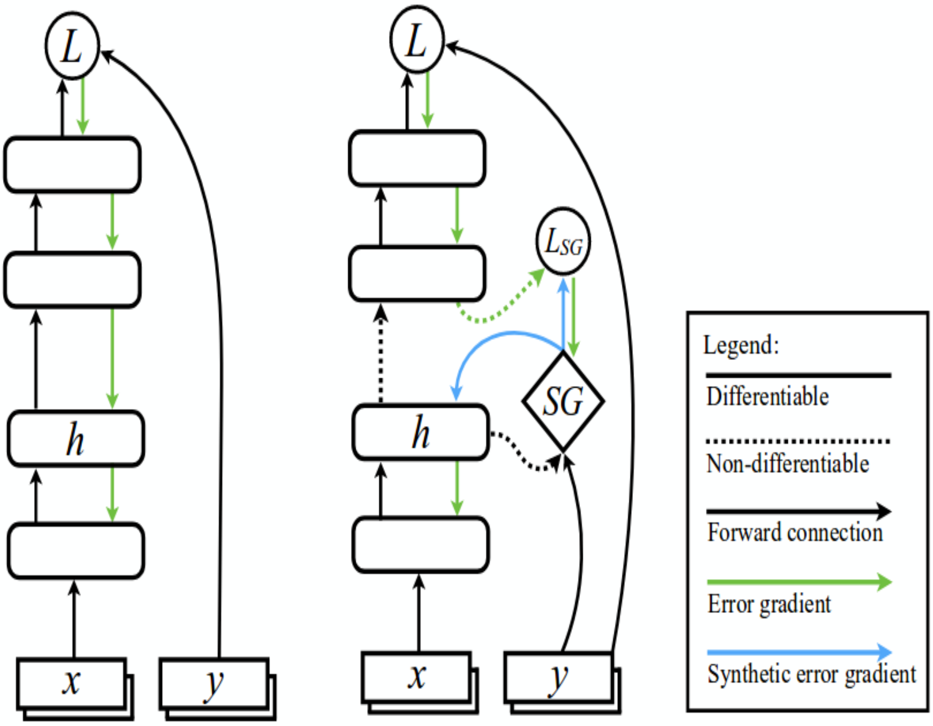

Figure 4.7: Back propagation without synthetic gradients and with synthetic gradients

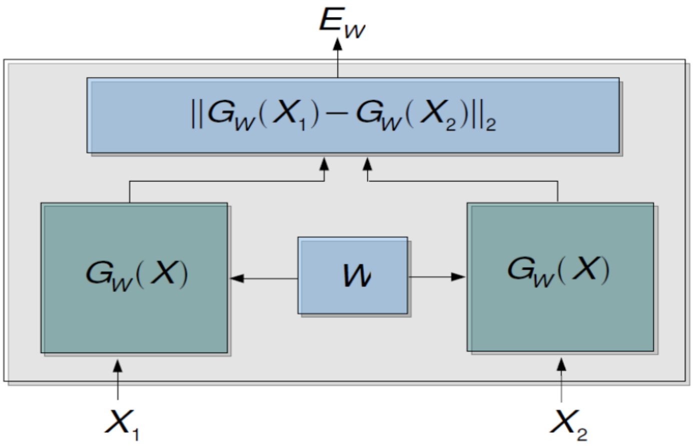

Figure 4.8: Siamese Architecture

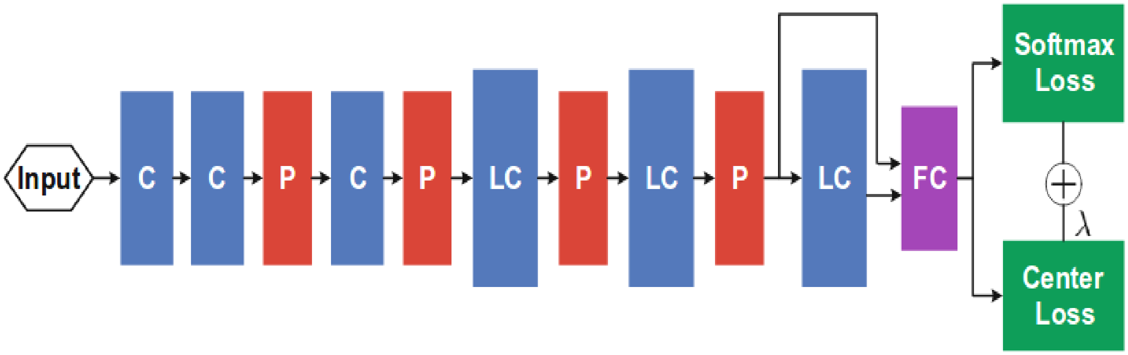

Figure 4.9: Training model using center loss where input is embeddings obtained from Inception-Resnet-v2, The CNN architecture using for face recognition experiments. Joint supervision is adopted. The filter sizes in both convolution and local convolution layers are 3x3 with stride 1, followed by PReLU [He et al. (2015)] nonlinear units. Weights in three local convolution layers are locally shared in the regions of 4x4, 2x2 and 1x1 respectively. The number of the feature maps are 128 for the convolution layers and 256 for the local convolution layers. The max-pooling grid is 2x2 and the stride is 2. The output of the 4th pooling layer and the 3th local convolution layer are concatenated as the input of the 1st fully connected layer. The output dimension of the fully connected layer is 512.

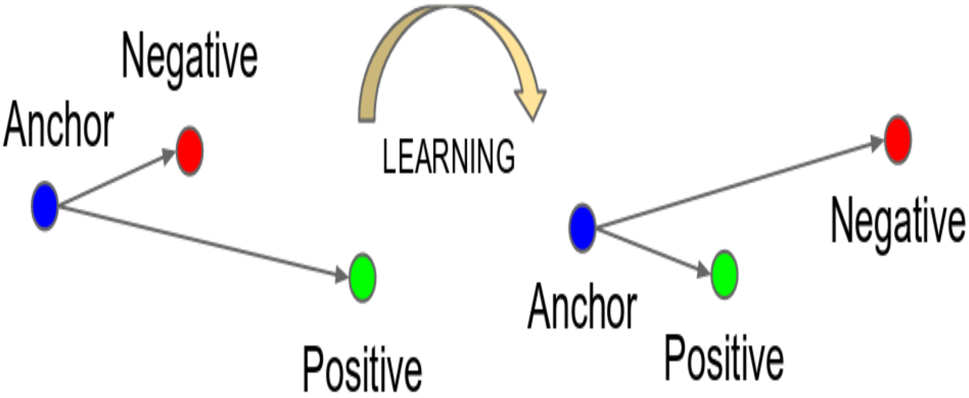

Figure 4.10: The Triplet Loss minimizes the distance between an anchor and a positive , both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity.

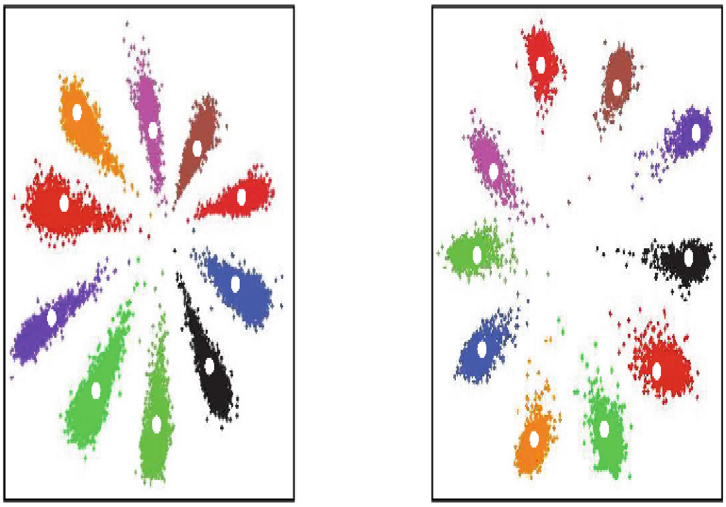

Figure 4.11: Embeddings visualization between triplet loss and center loss using tsne

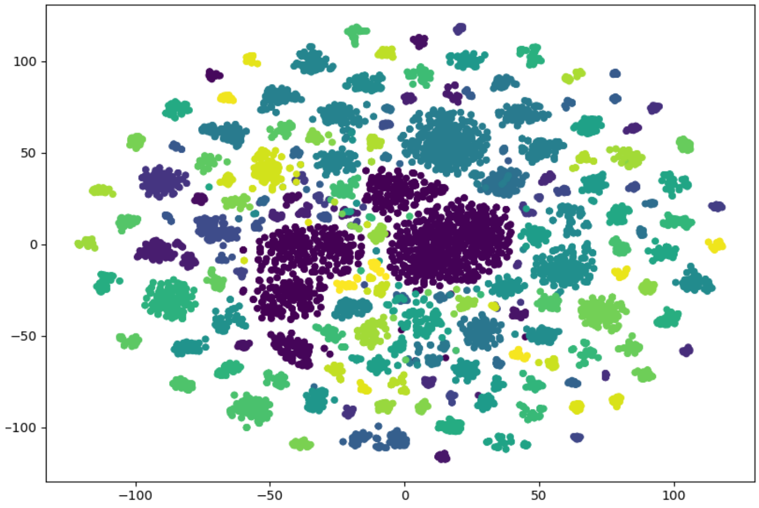

Figure 4.12: T-SNE visualization for 15K images as result of chinese whispers clustering

CHAPTER 5

CONCLUSION

5.1 CONCLUSION AND FURTHER RESEARCH DIRECTIONS

The model is deployed for handling streaming data where 100k images are uploaded every hour, while deploying the model big data engines are supposed to be used for scalability. Thus by retraining the model at different instance at nodes in cloud for every *T* time period with labels obtained from results of clustering, more optimal and precise face recognition results can be produced. Usage of high quality digital sources to capture images reduces the error rate in training the neural network. This research can be extended by usage of variational auto encoders or generative model to simulate and understand peoples feeling through facial expressions which adds up a factor in understanding the patterns of people behavior by Government and Business Agencies. Deep CNN model can be fine tuned to learn the behavior of primates in forests and which in turn results in understanding the ecosystem.

REFERENCES

Agarwal, A. (2015). How to enable facial recognition in your google photos. https://www.labnol.org/internet/facial-recognition-in-google-photos/28892/.

Amos, B., Ludwiczuk, B., and Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. (2012). Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633.

Biemann, C. (2006). Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the first workshop on graph based methods for natural language processing*, pages 73–80. Association for Computational Linguistics.

Bishop, C. M. (2006). Softmax activation. https://en.wikipedia.org/wiki/Softmax_function.

Bromley, D. B. (1993). *Reputation, image and impression management.* John Wiley & Sons.

Chen, W., Chen, X., Zhang, J., and Huang, K. (2017). Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proc. CVPR*, volume 2.

Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357.

Cogswell, M., Ahmed, F., Girshick, R. B., Zitnick, L., and Batra, D. (2015). Reducing overfitting in deep networks by decorrelating representations. *CoRR*, abs/1511.06068.

Constine, J. (2017). Facebook's facial recognition now finds photos you're untagged in. https://techcrunch.com/2017/12/19/facebook-facial-recognition-photos/.

Cui, J., Wen, F., Xiao, R., Tian, Y., and Tang, X. (2007). Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 367–376. ACM.

Denyer, S. (2018). Beijing bets on facial recognition in a big drive for total surveillance. https://www.washingtonpost.com/news/world/wp/2018/01/07/feature/in-china-facial-recognition-is-sharp-end-of-a-drive-for-total-surveillance/?noredirect=on&utm_term=.5eff5fab0775.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ho, J., Yang, M.-H., Lim, J., Lee, K.-C., and Kriegman, D. (2003). Clustering appearances of objects under varying illumination conditions. In *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, volume 1, pages I–I. IEEE.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., and Kavukcuoglu, K. (2016). Decoupled neural interfaces using synthetic gradients. *CoRR*, abs/1608.05343.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition.

Koch, G. G. (1982). Intraclass correlation coefficient. *Encyclopedia of statistical sciences*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Li, C.-G., Vidal, R., et al. (2015). Structured sparse subspace clustering: A unified optimization framework. In *CVPR*, pages 277–286.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In *BMVC*, volume 1, page 6.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567.

Tian, Y., Liu, W., Xiao, R., Wen, F., and Tang, X. (2007). A face annotation framework with partial clustering and interactive labeling. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., and Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. *CoRR*, abs/1801.09414.

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer.

Xiang, J. and Zhu, G. (2017). Joint face detection and facial expression recognition with mtcnn. In *Information Science and Control Engineering (ICISCE), 2017 4th International Conference on*, pages 424–427. IEEE.

Zadeh, A., Lim, Y. C., Baltrušaitis, T., and Morency, L.-P. (2017). Convolutional experts constrained local model for 3d facial landmark detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, volume 7.

Zhang, G., He, R., and Davis, L. S. (2014). Jointly learning dictionaries and subspace structure for video-based face recognition. In *Asian Conference on Computer Vision*, pages 97–111. Springer.

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

Zhao, M., Teo, Y. W., Liu, S., Chua, T.-S., and Jain, R. (2006). Automatic person annotation of family photo album. In *International Conference on Image and Video Retrieval*, pages 163–172. Springer.

Zhu, C., Wen, F., and Sun, J. (2011). A rank-order distance based clustering algorithm for face tagging. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 481–488. IEEE.

BIODATA

Name	RaviRaaja L
Address	210/1 Shanthi Illam, Vandipet street, Swaminathapuram, Salem , TamilNadu PIN code - 636009
E-mail	mailstoraviraja@gmail.com
Mobile	9626161704
Qualification	B.Tech. in Information Technology (Anna University, Chennai) M.Tech. in Computer Science and Engineering - Information Security (NITK Surathkal)