Face Recogniton from Real Time videos using Autoencoders

Erupothu Harish

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

Department of Computer Science Engineering

June 2015

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

E. Harish

(Signature)

Erupothu Harish

(Erupothu Harish)

CS13M1009

(Roll No.)

Approval Sheet

This Thesis entitled Face Recogniton from Real Time videos using Autoencoders by Erupothu Harish is approved for the degree of Master of Technology from IIT Hyderabad

[illegible]

(———-) Examiner Vineeth N B

Dept. of ~~Chem Eng~~ CSE

~~IITM~~ IITH

[illegible] 25.6.15

(———) Examiner Sumohana Channappayya

Dept. ~~Math~~ EE

IITH

[illegible]

(————) Adviser

Dept. of ~~Chem Eng~~ CSE

IITH

Dr. C.Krishna Mohan

~~(———) Co-Adviser~~

~~Dept. of Chem Eng~~

~~IITH~~

[illegible]

(———) Chairman

~~Dept. of Mech Eng~~

CSE IITH

Dr. Naveen Sivadasan

Acknowledgements

On the submission of my thesis work on 'Face recognition from Real Time video using Stacked Autoencoders', I would like like to take this opportunity to express my deep sense of of graduate and sincere thanks to all who helped me directly or indirectly during my thesis work. I would like to thank to my supervisor Dr.C Krishna Mohan, for being a great mentor and the best adviser. His advise and encouragement is the success behind my successful completion of my thesis work. I thank to Debaditya Roy a PHD Researcher who help me at the time i need most. I thank to IITH CCTV Security Officer for providing the CCTV videos for my research work. I thank to my classmates and co researches Kanishka Chowhan, Rahul Patil, Rohit katiyar, Kalla Srinivas, mettu srinivas, Dinesh and Ramaraju helping me for preparing database and making my work success. I thank my family members for their blessings and prayers.

Abstract

This thesis presents the Overview of the problem of Face recognition from Real time surveillance video. A survey on face detection algorithms as well as implementation and testing of facial feature extraction and dimension reduction methods and to prepare own data for good features extraction.

Human face detection and recognition play important roles in many applications such as video surveillance, Biometrics, Law enforcement and System security. In our project, we I have studied and worked on face detection, Tracking and Recognition techniques and developed algorithms for them. I used Deep Learning Technique Stacked Autoencoders to recognize face in continuously coming frames. These techniques works well under robust conditions like different lighting conditions, different face poses, at low resolution and even the video is obstruction to human face.

My aim, which I believe I have reached, was to develop a Deep Learning method Stacked Autoencoders for face recognition in unsupervised, robust and accurate. I applied Viola Jones Algorithms for face detection which is detecting fine. The examples provided in this thesis are real-time and taken from my surroundings. My database consist of Youtube databse, Webcam Real time data, IITH CCTV data and Chockepoint database.

Contents

Chapter 1

Introduction

1.1 Introduction to Face Recognition

The Recognition plays an important role in finding the identity and emotions of human face. We Human can recognize a many number of faces which are learned throughout our life and we can identify the faces even after many years. This skill is quite robust even though there are many variations like expression, age invariation, lighting condition, pose and Obstruction such as beard, changes in hairstyle and wearing glasses.

1.2 Applications and Challanges of Face Recognition

Face Recognition is applied many fields like law enforcement: for criminal identification, voter-Id and passport verification, security systems: for operating system, protecting database, human-computer interaction and authentication, Biometrics and Surveillance videos: for finding unknown person in boarder security, fraud detection in ATMs and in some security area, Face recognition is also used in many places now a days like social network like Facebook for face tagging, Google search for face searching and face morphing in film processing. face detection and recognition is difficult task because faces are complex and multidimensional.

Face Recognition from video is an important area in biometrics Research today. Most of the existing work focuses on recognition from video where the images are of high-resolution, containing faces in a frontal pose and where the lighting conditions are optimal. However, face recognition from video surveillance has become an increasingly important goal.

1.3 Description of video Quality

In Real time videos, the system must be robust to changes like pose, illumination, pose and expression. The system should also be able to perform detection and recognition in real time surveillance videos.

surveillance-quality video based on four different characteristics are variable illumination, pose, the low resolution, obstructions, Noise and blurriness . The challenges to face recognition in surveillance cameras are mainly due to the following reasons

1.3.1 Illumination

As the lighting is usually not uniform in the area of surveillance cameras, the illumination on the Human face could vary significantly. This can change the intensity of the image, even causing different parts of the image to be illuminated differently, which can cause problems for the recognition system. feace recognition at different lighting conditions is shown in fig: 1.1

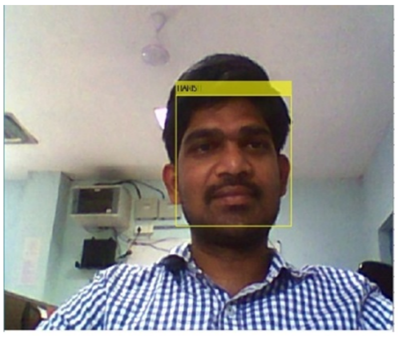


Figure 1.1: face Recognition at different lighting conditions

1.3.2 Pose Invariation

In Surveillance videos the subject will be moving freely in the field of view of camera and the captured face will be in different pose in different cameras making it difficult to face transformation. face recognition in different pose is shown in fig: 1.2

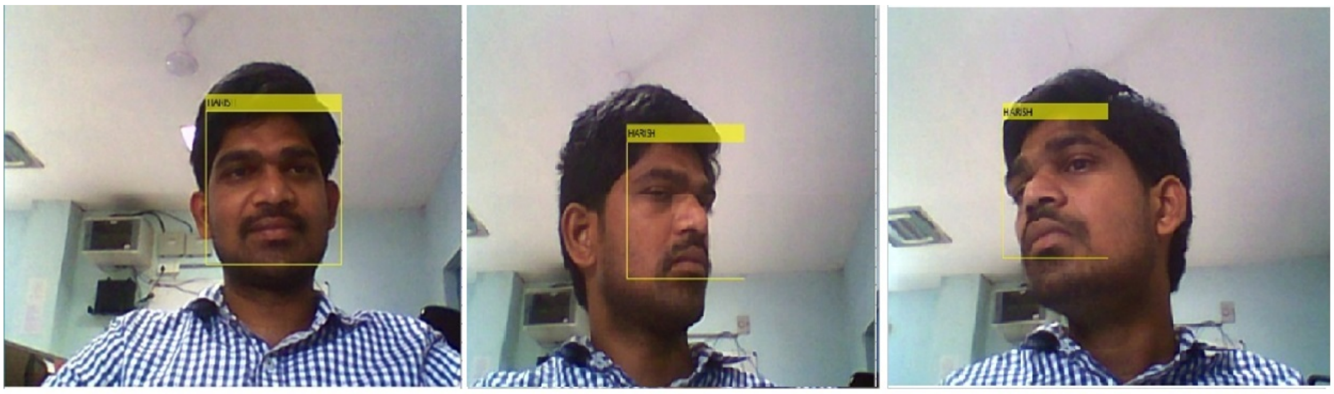


Figure 1.2: Face Recognition in different poses

1.3.3 Low resolution

Surveillance cameras normally capture images at low resolution the pixels in face images are very limited. If camera located far from the subject then subjects face will be small, causing the number of pixels on the subjects face to be low, making it difficult for robust face recognition. face recognition in low resolution frame is shown in fig: 1.3

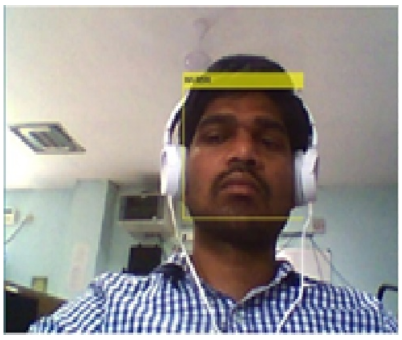


Figure 1.3: Face Recognition at low resolution

1.3.4 Obstruction

if the subject cover their face to prevent the camera from capturing their face by wearing Hats, glasses and makeup can also be used to change the appearance of the face to cause problems for recognition systems. face recognition in Obstruction is shown in fig:

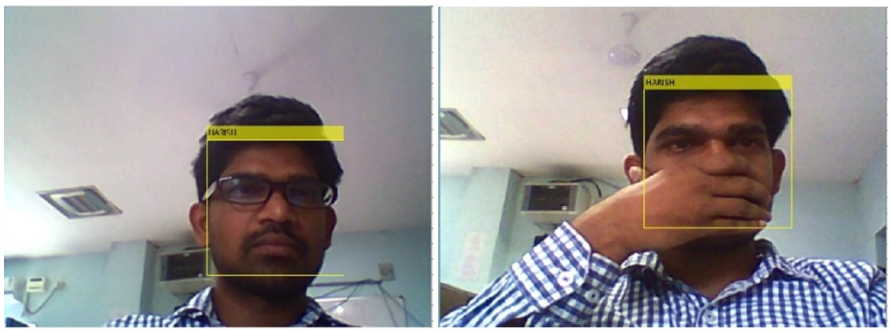


Figure 1.4: Face Recognition in occlution

1.3.5 Noise and Blurness

The captured images are often corrupted by noise and the motion of the subjects usually introduces blurriness to cause problems for recognition systems

1.4 Methods and Database for Face Recognition

I prepared database for face Recognition from Real world examples, IITH CCTV surveillance video database, Interview with politicians database, Hp Webcam Real time video databse and chockepoint standard database. The methods used are Viola Jones Face detection Algorithms, KLT tracker and Stacked Autoencoder for face Recognition.

1.5 Face Recogiton using Deep Learning

Deep Learning is a powerful tool that is used by by many giant companies like Google, Facebook, Baidu and Twitter because of its state of art results in many area like computer vision, speech recognition and Natural Language Processing. In my thesis I'm using Stacked Autoencoders a unsupervised Deep Learning Algorithm for face recognition.

Chapter 2

Face Detection

2.1 Viola Jones Face Detecton

Viola Jones Face detection Algorithms is a powerful tool that has the capacity to Processing the images extremely at high detection rate. This thesis gives the basic knowledge of face detection and how the humans face detected in mobile pones, digital cameras and Computers. we apply different methods like Haar features for extracting the edge features by subtracting the sum of pixels in black rectangle from white rectangle, and Integral images is used for computing these calculations very quickly. Adaboost is a learning algorithm that is used to select the features by removing redundant features which yield extremely efficient classifier finally combine the classifiers in a cascade to quickly discard the background regions of the image.

viola jones Face detection algorithm works with four basic methods they are Haar Features, Integral images, Adaboost and Cascading.

2.1.1 Haar Features

In object detection procedure, classificaton of images is done based on the value of simple features bu not the pixels directly. The reasons for using simple features are difficult to learn training data and the feature-based system identifies the face much more faster than a pixel-based system.

The simple features used resembles the Haar basis functions. In the face detection algorithm we use five types of features, first type is, the value computed between two horizontal rectangles is the difference between the sum of the pixels in white rectangle and black rectangle region. the value computed between two vertical rectangles is the difference between the sum of the pixels in white rectangle and black rectangle. These regions should be same size and shape and they are adjecent horizontally or verticallyt. A three-rectangle feature is computed as the difference between the sum within two outside white rectangles and center black region. Finally a four rectangle feature is the difference between sum of diagonal pairs of white rectangles and black rectangles. the size of the detection window is 24x24 all five types rectangle feature will extract 163000+ as shown in Fig. 2.1

Rectangle features provide rich image representation and they are somewhat relative when compared with alternatives such as steerable filters. Rectangle features are sensitive to the presence of simple image structure, bars and edges. Where as steerable filters are good at finding the bound-

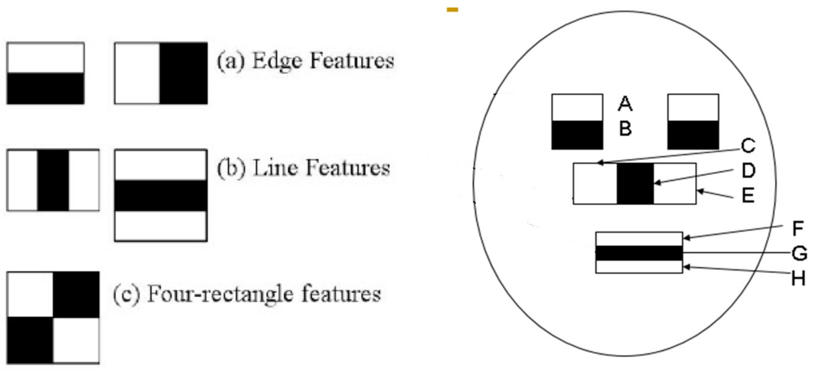


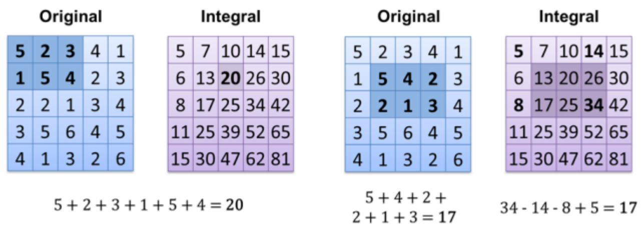
Figure 2.1: Types of Haar Features

aries, texture analysis and image compression. The orientation for Haar features are vertical and horizontal rectangle regions. These rectangle features provide a rich image representation which gives effective learning.

from the above Fig. 2.1. In two regions the difference between the sum of pixel within RegionA and regionB gives the Haar Feature that looks similar to the eye region. In three regions the difference between the sum of the outside regions regionC, regionE and the centre regionD gives the Haar feature that looks like Nose. similarly the for mouth region these Haar features detect the face in different aspects.

2.1.2 Integral image

integral image method is used to compute the rectangle feature very fast at any location of the pixels, viola jones algorithm does not work directly with image pixel values it computes the rectangle region features which are reminiscent of Haar Basis function. In order to compute these rectangle features, integral image representation is used. The integral image at particular location can be computed as the sum of the pixels above and to the left of the pixel. The integral image method with computed with very a few operations per rectangle. the intgral images is as shown in Fig. 2.2



Original

5	**2**	**3**	4	1
1	**5**	**4**	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

Integral

5	7	10	14	15
6	13	**20**	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

5 + 2 + 3 + 1 + 5 + 4 = 20

Original

5	2	3	4	1
1	**5**	**4**	**2**	3
2	**2**	**1**	**3**	4
3	5	6	4	5
4	1	3	2	6

5 + 4 + 2 +
2 + 1 + 3 = 17

Integral

5	7	10	**14**	15
6	13	20	26	30
8	17	25	**34**	42
11	25	39	52	65
15	30	47	62	81

34 - 14 - 8 + 5 = 17

Figure 2.2: Integral image representation

from the above figure we are clear how to compute the sum of the pixel values at particular location as the sum of the pixels above and to the left of the pixel with few operations.

2.1.3 Adaboost

Adaboost is one of the viola jones method which is used to construct a classifier by selecting important features. In a image Adaboost selects sub-window of size 24x24 and the total Haar features extracted with five types of rectangle will be 63000+ features which will take so much time for classification so in-order to make adaboost a fast classifier the learning process must remove a large majority of the redundant features and it should concentrate on a small set of important features. Feature selection is done by simple modification of the AdaBoost, in each stage of the boosting process Adaboost selects a new weak classifier. The weak learning classifier returned can depend on only a single feature. Adaboost is an effective learning algorithm that provieds the strong bounds in performance. Adaboost classifier is shown Fig. 2.3

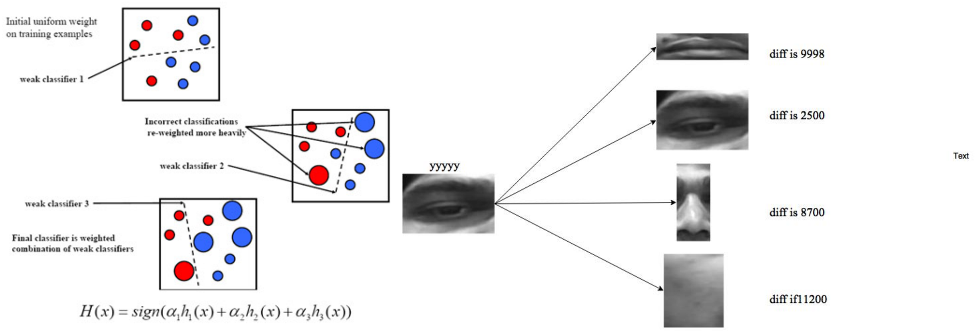

Initial uniform weight on training examples
weak classifier 1
Incorrect classifications re-weighted more heavily
weak classifier 2
weak classifier 3
Final classifier is weighted combination of weak classifiers
$H(x) = sign(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$
yyyyy
diff is 9998
diff is 2500
diff is 8700
diff if11200
Text

Figure 2.3: Adaboost Classifier

In face detection Algorithm, Adaboost selects the meaningful rectangle features. The first feature selected selected by Adaboost Classifier will focus on eye region which is darker than the nose and cheeks region. The second feature will focus on nose region in which the bridge of nose will be brighter than the sided region. Third feature will be the mouth in which the lips part is darker than the upper lip and lower lip.

2.1.4 Cascading

Cascading is the fourth method in viola jones algorithm which is used to fast the Algorithm by selecting series of classifiers and dividing the feature into each classifier and discarding the negative examples. Cascade of classifiers is constructed to increased detection performance and reduces computation time which is more efficient and boosted classifiers is constructed to reject many of the negative sub-windows. cascading is shown in Fig. 2.4

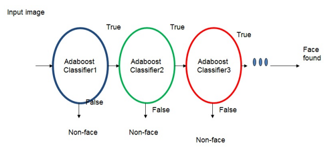


Figure 2.4: Cascading

From the above figure A series of classifiers are applied to every sub window. Each classifier is used to eliminate a large number of negative examples with very little processing and then the second classifier eliminates few more negative examples. ones the negative example found it will immediately discards the negative example. After several processing steps the number of sub windows will get reduced.

2.2 Summary

In this thesis viola jones Algorithm is used for face Detecton, below results show that all the 24 faces in IITH teacher Day photo and 19 face in Vigil Group photo are detected correctly. These face are detecting even half of the face is hidden in vigil group photo in 720x480 pixel image. still there is some feature work to do, for high resolution image 1320x2400 pixel image it is detecting all faces including some non faces and for very low resolution image few faces are not detecting.

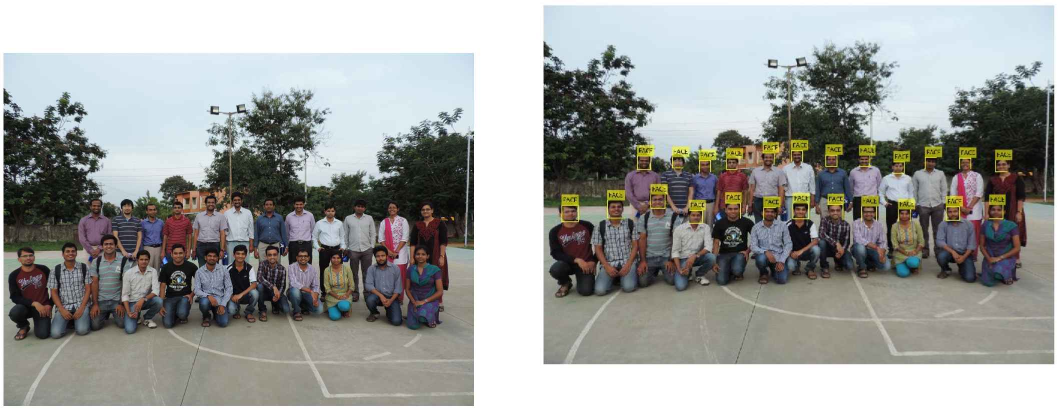


Figure 2.5: Face Detection in IITH Teachers Day photo

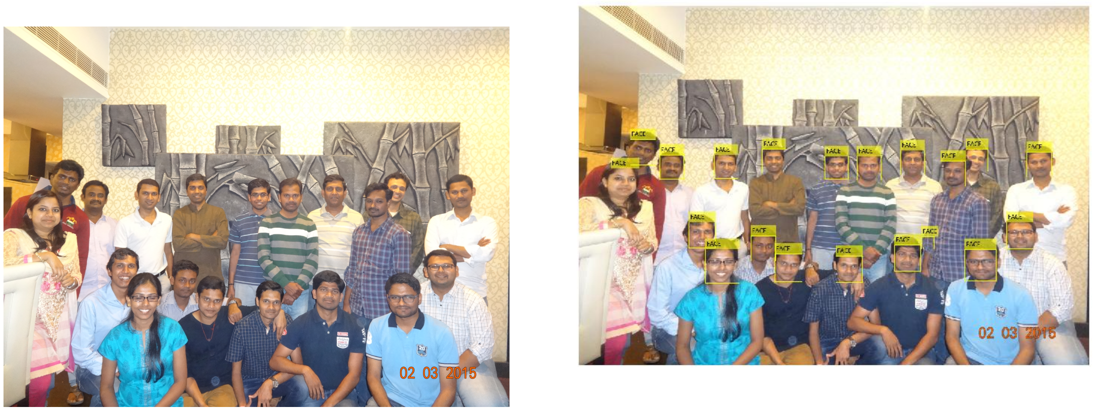


Figure 2.6: Face Detection Vigil group photo

Chapter 3

Face Tracking

3.1 Introduction to Face Tracking

For detecting a face in Real Time video Face Tracking is also plays an important role. In this thesis i'm using KLT tracker to track the detect face by extracting the eigen feature from the face bounding box. here I used standard matlab tool box for face tracking and describing the basic introduction to face tracking.

3.1.1 Feature Selection

Usually the KLT tracker uses the algorithm of Shi and Tomasi for detecting good features to track automatically but any points can be selected according to any criteria. The ability to track may not be as good. Good features to track are corners, pixels in regions with irregular intensities and points on non-straight edges. These features can be found by automatic analysis of the texture in the region. The regions with a high eigenvalue are then set as features. The differentiation between good features to track and non-good features to track is done by comparing the computed two eigenvalues to a threshold. The threshold is obtained from the histogram of the eigenvalues.

The feature selection process, the algorithm sorts the minor eigenvalues in decreasing order in order to picks feature coordinates from the top of the sorted list. It is assigned a new feature number every time a coordinate pair is selected. these feature changes when the frame is changed and if the same frame occur then the algorithm checks the matching features by selecting the eigenvalues and tracks.

3.1.2 Feature Extraction

When the camera moves the pixel intensities in the image changes in complex way. If the face did't move then there will be very little difference in change in the Object. but the face features will have some relative features and if the blur face if found the algorithm faces difficult to track the face. For face tracking process, the tracker picks the bounding box of the face image and computes the eigenvalues to extract the features.

An important problem in feature extracting is to find the displacement of a point from one frame to the next frame. It is very difficult to track single pixel unless it has a very distinctive brightness with respect to all of its neighbors. It is impossible to find where the pixel went in the frame based on local information or single pixel in the window. Noise pixel value will be changed and will be confused with adjacent pixels. Because of these problems we do not track single pixels but we take windows of pixels and look for good features.

3.1.3 Tracking

Tracking the face from the selected good features will makes tracking fast, we use only regions with a rich features to track corners, or windows with a high spatial frequency content. tracking is done by selecting minor eigenvalues from one frame to another and checking for matching feature and traks.

3.2 Summary

In this thesis a face window is taken from each frame and and set of good features are extracting and then the points are tracking with its tracker id, value. The feature points in the boundary box of face is show in Fig. 3.1

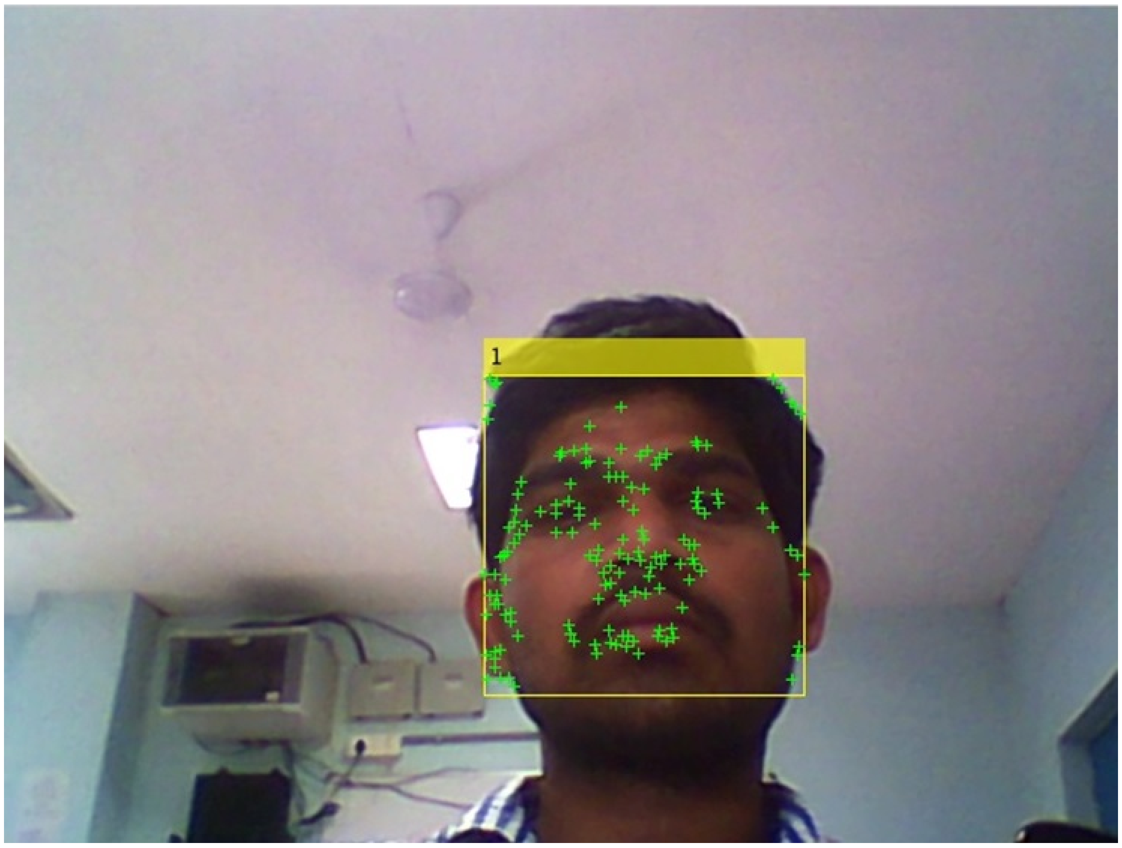


Figure 3.1: Tracking Face using KLT Tracker

Chapter 4

Deep Learning

4.1 Intoduction to Deep Learning

Deep learning has become more popular due to their success in challenging machine learning tasks the popularity is mainly due the way of learning. Research on artificial neural networks began in 1958 with perceptron learning algorithm. After that various kinds of artificial neural networks have been proposed like autoencoders, Deep belief networks,multi-layer perceptron,Boltzman machine, Principle component analysis, support vector machine, Hopfield networks, self-organizing maps and radialbasis function networks.

Deep networks were introduced in the 1980s by Fukushima which presented a hierarchical multi-layered neural network used for pattern recognition, such as the recognition and classification of handwritten characters. In 1991 Sepp Hochreiter identified the issue as the vanishing gradient problem which is the reason for the failure of deep neural network. In 1992, Hochreiters mentor, Jrgen Schmidhuber, tried solve the problem associated with deep networks with multi-level deep hierarchy which can pre-trained effectively one level at a time by initializing the weights randomly and unsupervised learning followed by a supervised backpropagation pass for fine-tuning. This method allows each level of the hierarchy to learn the input which is fed into the next level as the successive input the vanishing gradient problem was solved.

By the mid-1990s to early 2000s, The computational power has raised at satisfactory results was still out of reach the shallow architectures like Support Vector Machines (SVMs), single-layer neural network and kernel machines continued to be the predominant structure for machine learning algorithms. complex high-dimensional problems with sufficient data to capture the complexity are solved more efficiently and accurately with deep architecture for learning.

4.1.1 Deep Architecture

When your given a problem to solve, humans decompose the problem into smaller sub-problems and they solve in different levels. For example humans look at much smaller features of images, such as shape, lines, curves, and edges to determine the higher-level features. These highly-varying, non-linear features can build into layers that forms a deep network.

Deep learning generally refers to learning models which use feature hierarchies with many layers. composed of input and output and hidden layers. hidden layers can represent low-level features, such as pixels, edges, lines and shapes to high-level features such as mouth, eyes, ears, nose then to much higher features like face, hands, legs and body of image. Deep Learning aims to automatically discover these abstractions from the lowest to highest levels representation and learning algorithms is unsupervised. Deep Learning allows the network to discover features on its own rather than requiring a pre-defined set of all possible abstractions. Deep Learning has the power to automatically learn important underlying features which reason for the popularity of deep architectures as the wide applications of deep machine learning.

Feature re-use is of the advantage of deep learning, which explains the power of distributed representations, after training the number of examples and learning the features deep learning allows to add new examples to the existing data and train for feature extraction. as the weights are already learned the structure of example adding new example will make the feature to learn better.

Geoff Hinton introduced Deep learning in 2006 which is a breakthrough in feature learning. The main idea is greedy layerwise pre-training which is to learn a hierarchy of features one level at a time and the features extracted at each level is used as the input to next hidden layer to learn a new transformation at each level. Finally, we add the Classification layer to train the network in supervised Learning fashion and we combine all the layer and do forward propogation for better feature learning.

4.1.2 Importance of Deep learning

Deep learning is a hot topic in artificial intelligence. A branch of machine learning based on a set of algorithms that represents high-level abstractions in data and composed of multiple non-linear transformation, deep learning deals with neural networks to improve things like computer vision, natural language processing and speech recognition. Many Researcher shifted their research to Deep Learning due to its importance.

Giant companies like Google, Facebook, Microsoft, Baidu show more interest on Deep learning due to its state of art results which make it more popular. In 2006 Stanford University professor Andrew Ng became interest in Deep learning and started Google's Deep Learning project in 2011 as one of the Google X projects which is known as Google Brain. recently Andrew Ng shifted to Chinese search company Baidu. Google introduced deep-learning software word2vec tool designed to understand the relationships between words with no human guidance. Googles models were able to identify cats and human faces without any training. Google wants to make vast amount of data like search photos, web pages and YouTube videos to be searchable and understandable. Google working in speech recognition in mobile device business. Googles technology is currently used in the Android Operating System for speech recognition and photo search for Google+.

Facebook working on Deep learning its AI lab will is used for developing the deep learning techniques. Currently Facebook, automatically identifying the face in the photo and tagging the uploaded pictures with their names and sharing with friends, family and to the public which making an easy task to the users. Using similar techniques facebook wants to analyze our daily activity on the site so that it could automatically show you more stuff you wanna see.

4.2 single layer Neural Network

Neural Networks is a complex non-linear form of representation that fit to our data. A single layer neural network will have one input layer with number of input neurons and one output layer to learn desired output by using sigmoid activation function. To describe the single layer Neural Network we will start with simple possible network with single neuron displayed in Fig. 4.1

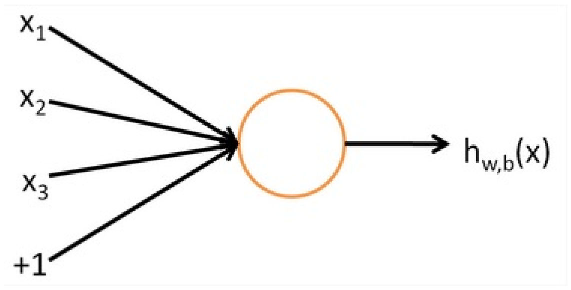


Figure 4.1: Single Neuron

From the above figure the neuron at the ouptut layer is sigmoid activation fuction or computational unit that takes input x_1, x_2, x_3 and bias or intercept term as +1, and computed output is $h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$, where f : $\Re \to \Re$ is called the activation function f(.) is the sigmoid function.

$$\text{f(z)} = \frac{1}{1+exp_{-z}}$$

this neuron show the input output that looks like logistic regression and f(z) is sigmoid activation function and its output range is [0 1]. if f(x) = $\frac{1}{1+exp_{-x}}$ is a sigmoid activation function then its derivative is f'(x) = f(x)(1-f(x)). This linear function has gradient 0 when z≤0 and 1 otherwise. The gradient is undefined at z=0, which will not cause problems in practice because we average the gradient over many training examples during optimization.

4.3 Multilayer Neural Network

In multilayer neural network the, network is put together with many simple hidden layers between the input and output layer so that the output of a each layer can be the input of another which is shown in Fig. 4.2

In the above figure the left most layer is called input layerx_1, x_2, x_3 and right most layer is called output layer $a_i^{(4)}$ and the middle layer and hidden layers $a_i^{(2)} and a_i^{(3)}$, and a fixed set of parameters Weights W and bias b. the computation is done as follows to compute at layer 2.

$$\begin{aligned} a_1^{(2)} &= f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)}) \\ a_2^{(2)} &= f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)}) \\ a_3^{(2)} &= f(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)}) \\ h_{W,b}(x) = a_1^{(3)} &= f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(2)}) \end{aligned}$$

we compute output at first hidden layer as $a^{(2)}$ similarly we compute the activation second hidden layer at layer 3 $a^{(3)}$ and finally compute the output at layer L $a^{(l+1)}$ as follows.

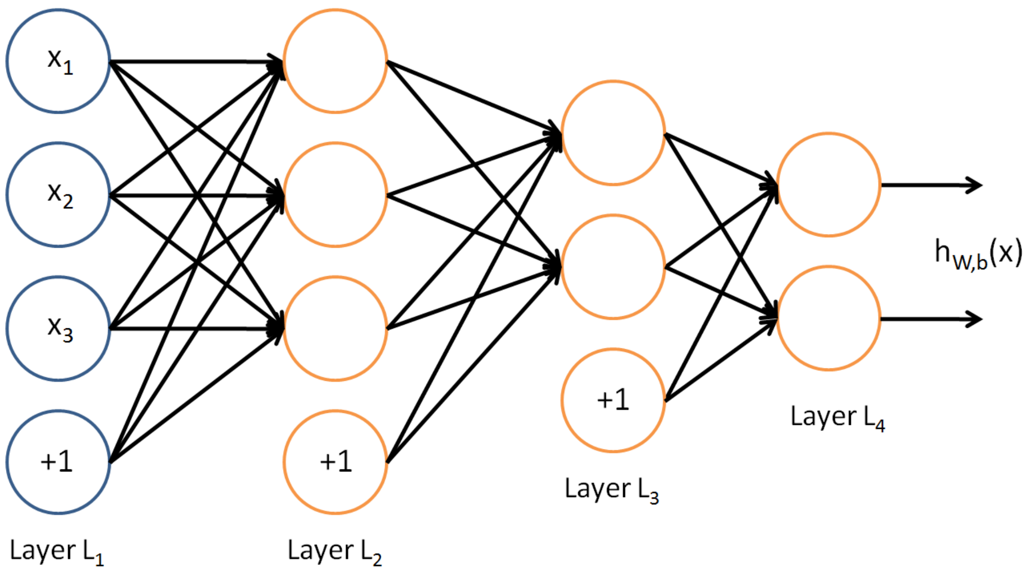


Figure 4.2: Multilayer Neural Network

$$z_i^{(2)} = \sum_{j=1}^{n} W_{i,j}^{(1)} x_j + b_i^{(1)}$$
$$a^{(2)} = f(z^{(2)})$$
$$z_i^{(3)} = \sum_{j=1}^{n} W_{i,j}^{(2)} x_j + b_i^{(2)}$$
$$h_{W,b}(x) = a^{(3)} = f(z^{(3)})$$
$$\text{at output layer : } z_i^{(l+1)} = \sum_{j=1}^{n} W_{i,j}^{(l)} x_j + b_i^{(l)}$$
$$a^{(l+1)} = f(z^{(l+1)})$$

The above equation shows the feed forward Neural Network that is computed the output at each layer. To train this neural network first take the input examples as $(x^{(i)}, y^{(i)})$ where where $x^{(i)}$ is the input vector and $y^{(i)}$ is Output vector. there will be number of hidden layer to computes the compressed feature in which each feature extracted in each hidden layer will become the input to the next hidden layer in that way this is done till the final layer. computing such big network will be more complex for computing so we go for stacked Autoencoders which is discussed in chapter 6.

Back Propagation

In the process of backpropagation the weights are readjusted depending on the existence of some expected output. Suppose we have training examples as $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)})$...., $(x^{(k)}, y^{(k)})$ of total k training examples. first we do forward propagaion as discussed in the above section then to adjust the weights we go for back propagation to learn good features in the network we make the initial weights close to zero then we train the neural network using batch gradient descent. We write the cost function as shown in below equation.

$$\text{J(W,b;x,y)} = \frac{1}{2} \parallel h_{W,b}(x) - y \parallel^2$$

now we define the overall cost function to be the avarage sum of square error term and regularization or weight decay term that decreases the magnitude of weights and helps to prevent overfitting. the equation is as follows.

$$\text{overall cost function J(W,b)} = [\frac{1}{m} \sum_{i=1}^{m} J(W,b;x,y)] + \frac{\lambda}{2} \sum_{l=1}^{n-1} \sum_{i=1}^{s1} \sum_{j=1}^{sl+1} (W_{j,i}^{(l)})^2$$

Gradient Descent

In this thesis Our goal is to minimize J(W,b). to minimize cost will initialize weights and bias term to a small random value near zero and train network with limited memory batch gradient descent optimization method toolbox or stochastic gradient descent optimization method which a simple alternative technique optimization method. Since cost function is a non-convex function, gradient descent is close to local optima and gradient descent works fairly well. the weight and bias update can be shown in below equations.

$$W_{i,j}^{(l)} = W_{i,j}^{(l)} - \alpha \frac{\partial}{\partial W_{i,j}^{(l)}} J(W,b)$$
$$b_{i}^{(l)} = b_{i}^{(l)} - \alpha \frac{\partial}{\partial b_{i}^{(l)}} J(W,b)$$

4.4 Summary

In this thesis we learn how the neural network learning a hierarchy of features by feed forward Neural network, Back propagation and Stochastic gradient descent optimization method. this method will be used for stacked Autoencoders for learning hierarchy of features in greedy layer wise manner.

Chapter 5

Prepare Database for Deep Learning

5.1 Extract Face images

preparing database in an important step in many deep learning algorithms. Many algotithms work best after the data has been normalized and whitened. However the exact parameters for preparing database are not easy. In my thesis I have prepared the database for testing in videos and my database resembles the real world examples. I tested my Stacked Autoencoders Deep learning algorithm with YouTube videos, Real time Webcam, Chockepoint and IITH CCTV database.

5.2 Normalize

There are few possible approaches for data normalization depending upon the data. The common methods for feature normalization are:

Re-scaling

Re-scale the data along each data dimension so that the final data vectors lies in the range [0,1], [1,1] or [0 255]. depending on Algorithm requirement.This is useful for later processing as many default parameters for data to scale to a reasonable range. For example Natural images obtain pixels values in the range [0 255] to rescale these values to [0 1] we divide the data by 255.

Mean subtractions

If your data is stationary then you have to subtract the mean-value for each example. For example In images this normalization has the property of removing the average brightness of the data point. This method is generally used for color images.

Feature Standardization

For Feature standardization each dimension of the data should have zero-mean and unit-variance. This is the most common method for normalization and is widely used. In practice we compute the mean of each dimension and subtracts this from each dimension then this dimension is divided by its standard deviation. For example When working with audio data, it is common to use MFCCs as the data representation.

5.3 PCA and ZCA Whitening

After doing normalization next step in reprocessing is whitening which helps to make our algorithms work better. Many deep learning algorithms rely on whitening to learn good features.

PCA Whitening

PCA play important role in whitening the raw images by extracting the eigenvalues and speed up mos of the Deep learning Algorithms. Principal Components Analysis (PCA) is a dimensional reduction algorithm. The understanding PCA features will enable us to implement whitening, which is an important prepossessing step for many algorithms.

Most of input raw images will be somewhat redundant because the values of adjacent pixels in an image are highly correlated to make these adjacent pixels uncorrelated we do whitening in this process we do PCA whitening, PCA will allow us to make the input with a much lower dimensional and reduce the error.

For whitening process first we compute the standard deviation of the data, if data x has zero mean then Σ is exactly the co-variance matrix of x. the equation for standard deviation is

$$\Sigma = \frac{1}{m}\Sigma_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

next PCA computes the eigen vectors of Σ by using single value decomposition, in matlab we use the standard command [U,S,V] = svd(sigma). Here U is the principal direction of variation of the data, is the top eigenvector Σ, and V is the second eigenvector and S will contain corresponding eigenvalues.

$$x_{rot} = U^T x$$

Now we compute the rotated version of the data x_{rot} the subscript rot is the reflection of original data. Here U is that it is an orthogonal matrix that can satisfy the property $U^T U = UU^T = I$. So you can go back to the orthogonal data x from rotated vectorx_{rot} , this can be shown in below equation.

$$x = Ux_{rot}$$

because $Ux_{rot} = UU^T x = x$ finally we compute PCA whitening, To make each of our input features have unit variance, we can simply rescale each feature $x_{rot,i}$ by $\frac{1}{\sqrt{\lambda_i}}$ we define our whitened data $x_{PCAwhite}$ as follows:

$$x_{PCAwhite,i} = \frac{x_{rot,i}}{\sqrt{\lambda_i}}$$

Finally the data have covariance identity I is not unique. if U is any orthogonal matrix then $Ux_{PCAwhite}$ will also have identity covariance

$$x_{ZCAwhtie} = Ux_{PCAwhite}$$

Chapter 6

Facial Feature Extraction

6.1 Autoencoders

autoencoders were first introduced by Rumelhart in 1986 for learning more about the structure of data without using any labels to enable unsupervised learning and to produce the output similar to input and one common constrain is the number of hidden units are less compared to the input and output which enable the network to learn compressed representation of data thereby getting dimensionality reduction. though there are several advantages of deep network they are extremely hard and costly. in 2006 HInton showed that Restricted Boltzmann Machines] could be trained in a greedy unsupervised manner to find a good set of weights for a deep belief network. using these weights and fine-tuning provided state of art results at that time.in 2007 Bengio showed that deterministic shallow autoencoders could also be used to find good initial weights for supervised tasks.

$$Encodingstep : a^{l+1} = f(\Sigma_{j=1}^{n} W_{i,j}^{(l)} a_j^l + b_i^{(l)})$$
$$Decodingstep : a^{n+l+1} = f(\Sigma_{j=1}^{n} W_{i,j}^{(n-l)} a_j^{(n+l)} + b_i^{(n-l)})$$
$$where f(x) = \frac{1}{1+e^x}$$

An autoencoder is an unsupervised feature learning algorithm that uses backpropagation by setting the output values to be equal to the inputs. Autoencoder tries to learn the approximation of identity funciton i.e output $\widehat{x}$ is similar to x. Input data is x and output data is y where x=y. after computing we get the output as $\widehat{x}$. the Autoencoders is shown in Fig. 6.1. Autoencoders offer a method of automatically learning features from unlabelled data, allowing for unsupervised learning and tries to discover generic features of the data which learns approximation of identity function by learning important sub-features. Autoencoder does Compression which is important step for dimension reduction.

Previous deep learning approaches used random initialization of parameters before training a network. This was a major downfall of previous deep neural networks as a vast amount of time was required for the networks to learn, making the technique infeasible for practical learning. so preprocessing step sovles this problem by greedy layer-wise pre-training of each layer by initializing the parameters near a local.

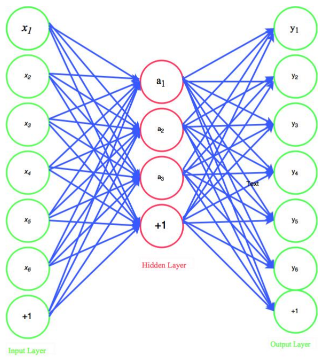


Figure 6.1: Autoencoders

6.2 Preprocessing

pretraining each layer is an important step for feature learning, less time is required to learn the optimal parameters and the model becomes significantly more computationally powerful. In stacked autoencoders the pre-training step will be performed on each layer independently. Initialize the parameters, optimizing method, cost function, Average Sum-of-Squares Error Term, Weight Decay Term Backpropagation and gradient descent are same as Sparse Autoencoders. pretraing takes care of following steps.

initializing weights : The goal of the pre-training is to minimize the cost, J(W,b), to minimize the cost we initialize the parameters weight and bias(W, b) to small near-zero values as a function of W and b in order to set the parameters in a good neighbourhood for further training. now Optimize the parameters by minimizing the cost function, J(W, b).

Optimization : optimization is a technique that makes the cuputation very fast.The cost function is minimized using the Limited memory BFGS optimization algorithm. Stochastic gradient descent(SGD) is a simple alternative optimization technique commonly used in training.

Cost function : Cost function is an important step for learning good features. cost function in Autoencoders is the sum of three components: a) the average sum-of squares error term, b) the weight decay term, and c) the sparsity penalty.

average sum-of squares error term : This term is used to measures the discrepancy between the observed or output data and the expected or estimated data, this is done by adjusting the wights by doing backpropagation. the equation for this is as follows.

$$\text{J(W,b;x,y)} = [\tfrac{1}{m}\Sigma_{i=1}^{m}\tfrac{1}{2} \parallel h_{W,b}(x) - y \parallel^2]$$

weight decay term : weight decay term is used to add a penalty to the error function. This term is useful for decreasing the magnitudes of the weight, thus minimizing the risk of overfitting. Without this penalty, large weights can cause excessive variance to the output. Therefore the weight decay term is used to regularize the weights by decreasing their magnitude. the equation is as follows.

$$\text{J(W,b;x,y)} = [\tfrac{1}{m}\Sigma_{i=1}^{m}\tfrac{1}{2} \parallel h_{W,b}(x) - y \parallel^2] + \tfrac{\lambda}{2}\sum_{l=1}^{n-1}\sum_{i=1}^{s1}\sum_{j=1}^{sl+1}(W_{j,i}^{(l)})^2$$

sparsity penalty : A sparsity constraint is placed on the network, limiting the activation of the hidden units with the goal of discovering the underlying structure of the data regardless of the number of hidden units used. the overall equaltion is as follows.

$$J(W,b;x,y) = [\frac{1}{m}\Sigma_{i=1}^{m}\frac{1}{2} \parallel h_{W,b}(x) - y \parallel^2] + \frac{\lambda}{2}\sum_{l=1}^{n-1}\sum_{i=1}^{s1}\sum_{j=1}^{sl+1}(W_{j,i}^{(l)})^2 + \beta\Sigma_{j=1}^{m}KL(\overline{a}\|a)$$

6.3 Feature Extraction using Stacked Autoencoders

Stacked Autoencoder is an unsupervised learning Algorithm consisting of multiple layers of sparse autoencoders in which the outputs of each layer is connected to the inputs of the next layer. The encoding and decoding step for Autoencoders is as follows:

$$Encodingstep : a^{l+1} = f(\Sigma_{j=1}^{n}W_{i,j}^{(l)}a_j^l + b_i^{(l)})$$
$$Decodingstep : a^{n+l+1} = f(\Sigma_{j=1}^{n}W_{i,j}^{(n-l)}a_j^{(n+l)} + b_i^{(n-l)})$$
$$wheref(x) = \frac{1}{1+e^x}$$

This is explain in detail as train first layer using your data without the labels using sparse Autoencoder then freeze the first layer parameters and start training the second layer using the output of the first layer as the unsupervised input to the second layer. Repeat this for as many layers as desired and use the outputs of the final layer as inputs to a supervised softmax regression finally Unfreeze all weights and fine tune the full network. the stacked autoencoders is shown in Fig. 6.2 fromt the

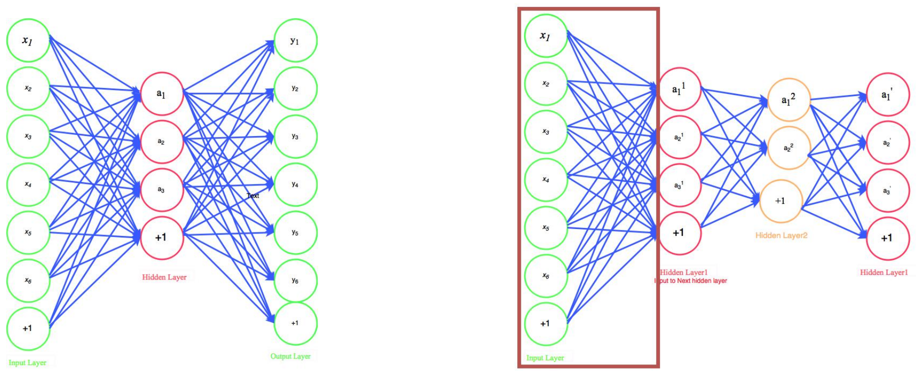


Figure 6.2: stacked Autoencoder

above figure train first layer using your data without the labels then freeze the first layer parameters and start training the second layer using the output of the first layer as the unsupervised input to the second layer repeat this for as many layers as desired this builds our set of robust features. now use the outputs of the final layer as inputs to a supervised layer/model and train the last supervised softmax regression layers then unfreeze all weights and fine tune the full network by training with a supervised approach for a given pre-processed weight settings.

A fine-tuning is commonly used in training stacked Autoencoders to improve the performance and layer-wise pre-training is used for extracting the good features in the network, fine-tuning will combine all the layers and do forward propagation so that it can slightly modify the features in network and adjust the boundaries between the softmax regression.

Chapter 7

Softmax Regression

7.1 Logistic Regression

Logistic regression is a simple classification algorithm for learning to make decisions whether a grid of pixel intensities represents class A or class B. In logistic regression we will try to predict the probability that a given example belongs to the class A or the probability that it belongs to class B. Specifically, we will try to learn a function of the form:

$$P(y = 1 \mid x) = h_\theta(x) = \frac{1}{1+exp^{(-\theta^T x)}} \text{ and}$$
$$P(y = 0 \mid x) = 1 - P(y = 1 \mid x) = 1 - h_\theta(x)$$

The funcition $h_\theta(x)$ is called the sigmoid or logistic function which keeps the value $\theta^T x$ in the range [0 1]. so that if the probability of $P(y = 1 \mid x)$ is large then x belongs to class 1 otherwise x belongs to class 0.

For a set of training examples with binary labels the following cost function measures how well a given $h_\theta(x)$ does this:

$$j(\theta) = -\Sigma_i (y^{(i)} log(h_\theta(x^{(i)}) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))$$

7.2 softmax Regression

Softmax regression is a extension of logistic regression, in logistic regression we use only two classes where as softmax regression handles multiple classes. In logistic regression we assumed that the labels were binary either 0 or 1 to distinguish between two classes.In the softmax regression we learn multi-class classification so the label y can take on K different classes, rather than only two.

Let k be the number of training labels and x be the input data then we estimate the probability of training label y to the given training data x as $P(y = k \mid x)$ for each value of k = 1.....K., we estimate the probability of getting class label y by taking each of the K different values. Thus the output is a K-dimensional vector which gives K estimated probabilities. Now the theta is $h_\theta(x)$ takes the form:

$$h_\theta(x) = \frac{1}{\sum_{j=1}^{K} exp^{(-\theta^{(j)T}x)}} \begin{bmatrix} exp^{(-\theta^{(1)T}x)} \\ exp^{(-\theta^{(2)T}x)} \\ . \\ . \\ exp^{(-\theta^{(3)T}x)} \end{bmatrix}$$

Cost fucntion now we describe the cost function that uses softmax regression. now we minimize the cost as $1\{Truestatement\} = 1$ and $1\{Falsestatement\}$ =0. out cost function is as follows.

$$j(\theta) = -[\Sigma_{i=1}^{m}(y^{(i)}log(h_\theta(x^{(i)})) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))]$$

$$= -[\Sigma_{i=1}^{m}\Sigma_{k=0}^{1}1\{y^{(i)} = k\}logp(y^{(i)} = k|x^{(i)};\theta)]$$

final layer features inputs to a supervised softmax regression layer and train the last supervised softmax regression layers then unfreeze all weights and fine tune the full network by training with a supervised approach for a given pre-processed weight settings. the softmax regression layer is shown in Fig. 7.1

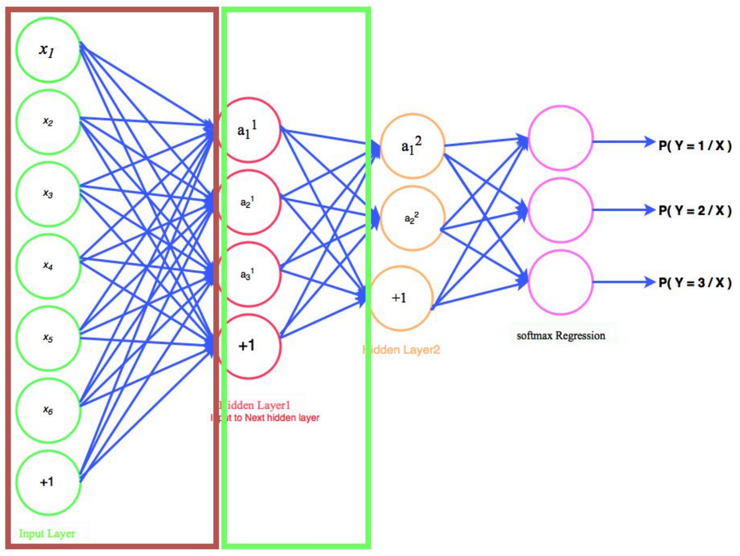


Figure 7.1: softmax regression Layer

Chapter 8

Experimental setup

8.1 System setup

in this thesis Face Recognition from Real Time video using stacked Autoencoders was implemented in Matlab R2013b which requires 5GB space for installation and the operating system used is Windows7 and windows8.1.

8.2 Databses

I prepared four databases base on real world examples they are webcam databse, politician database, IITH CCTV database and chokepoint database the details of these databases are as follows.

Webcam

Webcam data is extracted from hp laptap webcam to test with low resolution videos, the database consist of 32x32 images of 10 subjects of my classmates and phd students the labels are named as Harish, mettu srinivas, Debaditya Roy, Earnest Paul, Dinesh, Kalla srinivas, Kanishka Chohan, Ramaraju, Rahul and Rohit. I have taken 200 faces of each subject and trained with Stacked Autoencoders with two layer and three layers and labels are trained with softmax regression. I used Matlab for programming and implemented this in Matlab GUI to make clear visulization. the dataset is as shown in Fig. 8.2

IITH CCTV video

First i would like to thank to IITH CCTV officer for providing the CCTV videos for my research work. In IITH CCTV videos I have extracted 20 subjects each subject is trained with 100, 32x32 face images. since CCTV have very less resolution I would like to recognize the face even an low resolution so I have choose to prepare this database. The database consist of three professors Dr. Thamma Bheemarjuna Reddy sir, Dr. M V Pondu Ranga Rao sir, Dr. Ramakrishna Upadrastha sir and 16 IITH students and one unknown subject. The recognition one this database was successfully recognized. the database is shown in Fig. 8.2

politicians

Politician database is extracted from YouTube videos to test with high resolution videos. the videos that are taken was 720p it consist of 10 subjects with 32x32 face images these faces consist of inter-

Figure 8.1: Subjects of Webcam database

Figure 8.2: Subjects of IITH CCTV database

views of politicians like Barak obama, Nandamuri tharaka Ramarao, Atal Bihar vajpayee, Narendra modi, Sonia Ghandi, Venkaiah Naidu, Chandrababu Naidu, YS Rajashekar Reddy and Pawan kalyan. from each subject 200 face images are trained with Stacked Autoencoders and then classified. the subject in the database is shown in Fig. 8.3

Figure 8.3: Subjects of Politician database

Chokepoint dataset

Chokepoint dataset is a standard dataset that is taken from internet. This dataset consist of three cameras views with different lighting conditions and wanking one person at a time and multiple persons walking in front of cctv. from this dataset i have taken both single and multiple person waking data with 29 subjects 150 faces from each subject with different poses, their labels are not there so i created by myself. i have taken 32x32 face image to train with stacked Autoencoders.

8.3 Summary

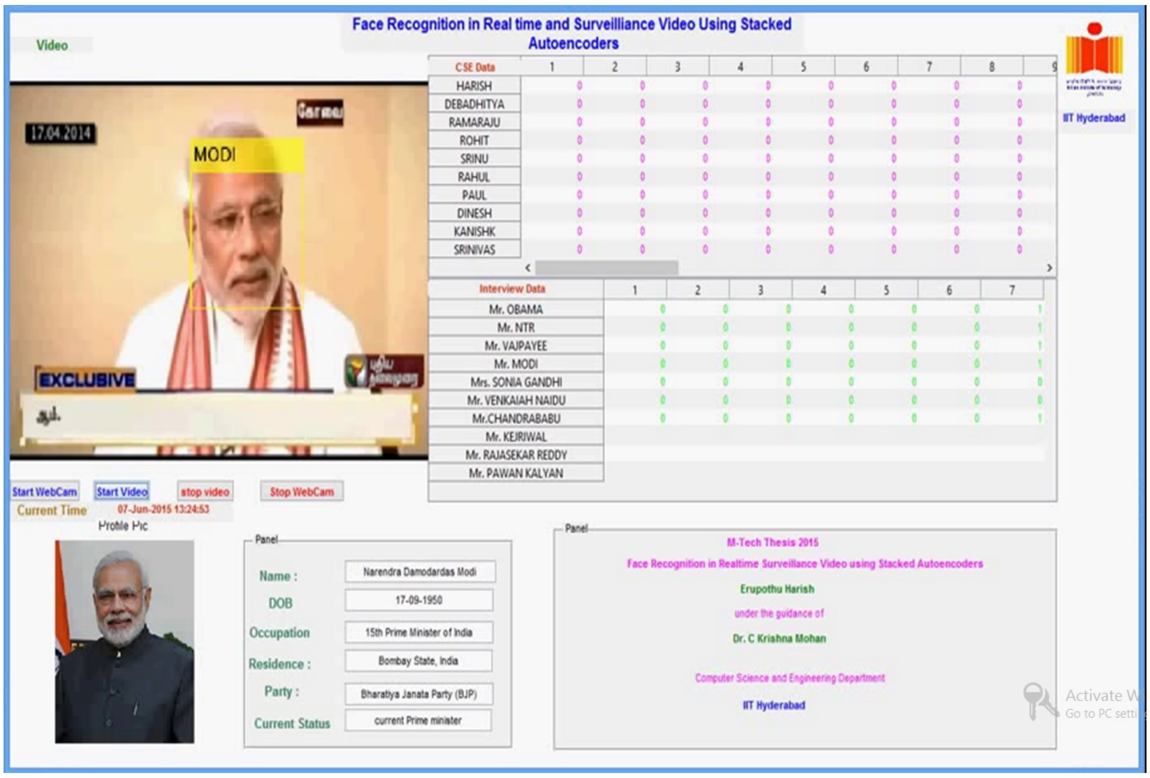
The above databases like Webcam database, IITH CCTV database and Politicians database was prepared by myself by studying the ufldl tutorial and chokepoint database is taken from internet but for training neural network I have modified it based on my requirement and the recognition of faces in all the database was successful in my operating system to the best of my knowledge.

Chapter 9

Results and Summary

9.1 Results

The Face recognition from Real time video using Stacked Autoencoders was successfully for all the databases that i have prepared and chokepoint database that has extracted from internet. the following figure below shows the one of the snapshot taken form my results. **Result1**



CSE Data	1	2	3	4	5	6	7	8	9
HARISH	0	0	0	0	0	0	0	0	
DEBADHITYA	0	0	0	0	0	0	0	0	
RAMARAJU	0	0	0	0	0	0	0	0	
ROHIT	0	0	0	0	0	0	0	0	
SRINU	0	0	0	0	0	0	0	0	
RAHUL	0	0	0	0	0	0	0	0	
PAUL	0	0	0	0	0	0	0	0	
DINESH	0	0	0	0	0	0	0	0	
KANISHK	0	0	0	0	0	0	0	0	
SRINIVAS	0	0	0	0	0	0	0	0	

Interview Data	1	2	3	4	5	6	7
Mr. OBAMA	0	0	0	0	0	0	1
Mr. NTR	0	0	0	0	0	0	1
Mr. VAJPAYEE	0	0	0	0	0	0	1
Mr. MODI	0	0	0	0	0	0	1
Mrs. SONIA GANDHI	0	0	0	0	0	0	0
Mr. VENKAIAH NAIDU	0	0	0	0	0	0	0
Mr.CHANDRABABU	0	0	0	0	0	0	1
Mr. KEJRIWAL							
Mr. RAJASEKAR REDDY							
Mr. PAWAN KALYAN							

Current Time 07-Jun-2015 13:24:53

Profile Pic

Panel

Name :	Narendra Damodardas Modi
DOB	17-09-1950
Occupation	15th Prime Minister of India
Residence :	Bombay State, India
Party :	Bharatiya Janata Party (BJP)
Current Status	current Prime minister

Panel

M-Tech Thesis 2015

Face Recognition in Realtime Surveillance Video using Stacked Autoencoders

Erupothu Harish

under the guidance of

Dr. C Krishna Mohan

Computer Science and Engineering Department

IIT Hyderabad

Figure 9.1: Result1 Recognizing Modi from Real Time video

The above figure recognizes the face of our Current prime minister of India Mr. Narendra modi and display his profile details like data of birth, National party, Residential address etc. The table is made for two databases one for Real time webcam face recognition and the second table show the politicians attendance based on the date he has given the interview.

9.2 Summary

The below Block diagram summarizes the complete implementation of Face Recognition system. the upper block is for training the databse and below block represents the Real Time face recognition system that take the feature from the top block and classify the person in the Ral Time video. **Block Diagram for Face Recognition using Stacked Autoencoders**

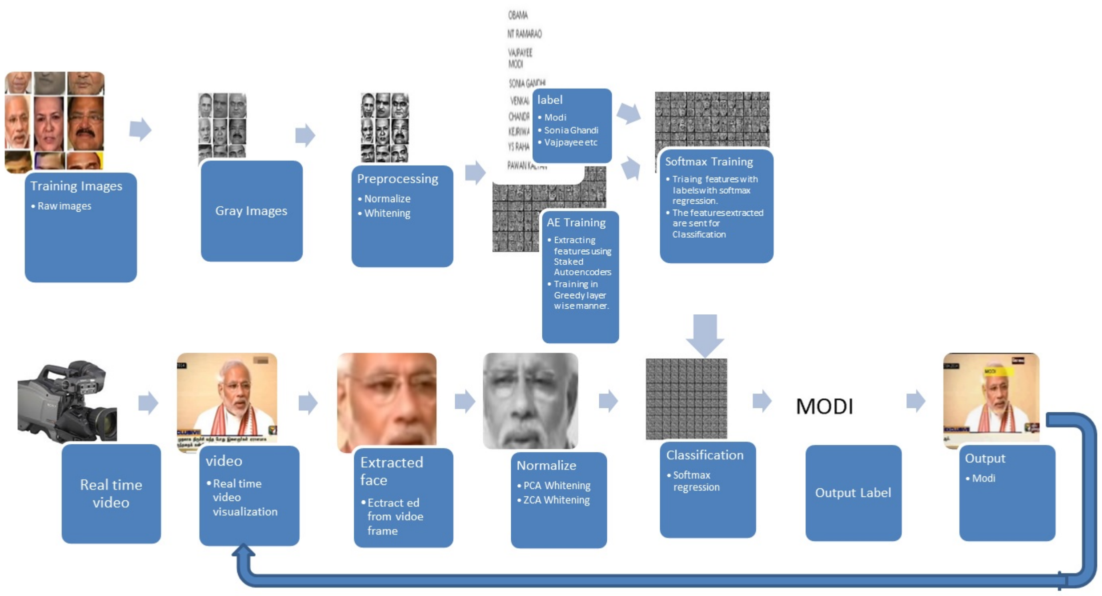


Figure 9.2: Block Diagram For Face Recognition

In the above figure show the block diagram of implementation of Face recognition system. each step in the bock diagram is explain in overall thesis from first chapter.

9.3 Future work

This thesis describes face recognition system that is able to recognize the face successful by using Stacked Autoencoders still some feature work to be done for better results in multiple face detection and recognition. My aim is build a face recognition system that can be used for recognizing the face in different lighting conditions, low resolution, at different poses and obstruction like wearing glasses, changing hair style etc and these problems are solved, the results are shown in chapter2. Deep Learning is a very good area that can make the machine learn by its own. most of the problem like low resolution, pose invariation and obstruction are solved in this thesis still research have to done for the better results.

References

[1] Y. Bengio. Learning deep architectures for AI. Foundations and Trends in Machine Learning. (2009) 1-127.

[2] Y. Bengio Deep learning of representations: Looking forward. 2013.

[3] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science. 2006.

[4] Vincent2011 A connection between score matching and denoising autoencoders. Neural Computation. 2011.

[5] P. Vincent and H. Larochelle and I. Lajoie and Y. Bengio and and P.-A. Manzagol Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. 2010.

[6] Yoshua Bengio and Aaron Courville and Pascal Vincent Representation Learning: A Review and New Perspectives. 2011.

[7] Viola, Paul A and Jones, Michael J Rapid Object Detection using a Boosted Cascade of Simple Features. 2001.

[8] Castrilln Marco and Dniz Oscar and Guerra Cayetano and Hernndez Mario Real-time detection of multiple faces at different resolutions in video streams. 2007.

[9] Carlo Tomasi and and Takeo Kanade. Detection and Tracking of Point Features. 1991.

[10] Jianbo Shi and and and Carlo Tomasi. Good Features to Track. 1994.

[11] G. Hinton and and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science. 2006.