TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

MAJOR PROJECT REPORT

ON

"LOW RESOLUTION FACE RECOGNITION USING DEEP LEARNING"

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

SUBMITTED BY:

AARCHAN BASNET (PUL075BCT003)

BISHAL LAMICHHANE(PUL075BCT029)

BISHWASH GURUNG(PUL075BCT031)

May, 2023

Page of Approval

TRIBHUVAN UNIVERSIY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certifies that they have read and recommended to the Institute of Engineering for acceptance of a project report entitled **"Low Resolution Face Recognition"** submitted by **Aarchan Basnet**, **Bishal Lamichhane** and **Bishwash Gurung** in partial fulfillment of the requirements for the Bachelor's degree in Electronics & Computer Engineering.

.....
Supervisor
Dr. Dibakar Raj Pant
Associate Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

.....
Internal examiner
Person B
Assistant Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

.....
External examiner
Person C
Assistant Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

Date of approval:

Copyright

Acknowledgement

We would like to take this oppurtunity to express our deepest gratitude and sincerest appreciation to all those who gave us the possiblity to complete our project. We would like to give our special thanks to our project supervisor **Prof. Dr. Dibakar Raj Pant**, Associate Professor, Department of Electronics and Computer engineering, Institute of Engineering, Central Campus, Pulchowk whose help, stimulating suggestions, invaluable guidance, motivating feedbacks and ever encouragement throughout our Bachelor's Programme made the completion of the project a reality.

We are also grateful to **Prof. Dr. Ram Krishna Maharjan**, Head of Department of Electronics and Computer Engineering, **Mr. Loknath Regmi**, Deputy Head of Department of Electronics and Computer Engineering and **Mr. Nischal Acharya**, Deputy Head of Department of Electronics and Computer Engineering for their regular support and co-operation. We would also like to acknowledge the help and co-operation of all the teaching and non-teaching staffs of the campus for the fruition of the project.

Last but not least, we would like to express our deep appreciation and gratitude to our family, friends and well-wishers who have always helped us to keep up the morale and excitement in our work; without their encouragement and everlasting love we would not have achieved our goals.

Aarchan Basnet (PUL075BCT003)
Bishal Lamichhane (PUL075BCT029)
Bishwash Gurung (PUL075BCT031)

Abstract

In recent years, face recognition systems have achieved impressive performance and results using variety of algorithms and methods but such methods often fail to recognize a face image of low resolutiom. Face Recognition(FR) degrades when faces are of very low resolution since many details about the difference between one person and another can only be captured in images of sufficient resolution. In order to have better face recognition in low resolution environment, our project uses a Convolutional Neural Network(CNN) model of a Residual Network Architecture to reconstruct the low resolution image into a higher resolution image and another CNN model to extract features from the newly reconstructed image to compare and recognize the face using a classifier. In case of the project, the low resolution image is taken of resolution 32X32 and the higher resolution image is of 128X128. The PSNR value for the super resolution model is 29.3256 dB and SSIM value is 0.7686. The accuracy of the Face Recognition model is 86.32% The performance of proposed method is evaluated on a custom face dataset using confusion matrix and it shows a decent precision and recall values.

Keywords : Low Resolution(LR), High resolution(HR), Super Resolution, Convolutional Neural Network

Contents

List of Figures

List of Tables

List of Abbreviations

SR Super Resolution
FR Face Recognition
LR Low Resolution
HR High Resolution
CNN Convolutional Neural Network
DCNN Deep Convolutional Neural Network
MISR Multiple Image Super Resolution
SISR Single Image Super Resolution
PSNR Peak Signal to Noise Ratio
SSIM Structural Similarity Index
MSE Mean Squared Error

Chapter 1

Introduction

1.1 Background

In many surveillance scenarios in real life, people may be far from the camera and their faces may be small in the field of view resulting in low resolution images. Such low resolution images can seriously degrade the performance of conventional face recognition systems which have been mainly developed for recognizing high quality images in controlled conditions.The project discusses a pipeline system of a deep learning super resolution model and a face recognition model in order to address the problem of recognizing low resolution probe face images when a gallery of high quality images is available. Traditionally, there are three standard approaches to this problem: 1) down sampling the gallery images to the resolution of the probe images and then performing the recognition, 2)obtain high resolution probe images from low resolution images and perform recognition 3) simultaneously transfer both LR probe image and HR gallery images into common space where corresponding LR image and HR images are closest in distance. In this project, the second approach that employs a SRResnet model[2] to reconstruct the low resolution image into a high resolution face image has been used and then VGG16[4] model was used to extract features from the image.

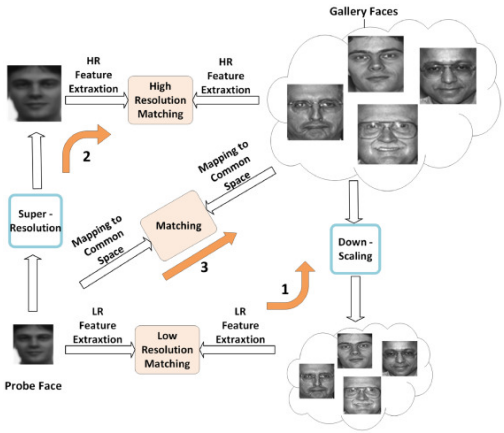

Figure 1.1: Three General Approaches for Low Resolution Face Recognition

1.2 Motivation

High Resolution images are not easily available in the real world as we tend to believe. Mostly used survelliance cameras, CCtvs, spy cameras and so on has lower specifications and need to work in wild environments resulting in low resolution images due to distance, lighting etc. Present face recognition system mainly work as biometric and use high resolution images to compare and recognize, so in order to have real world system that can actually take these low resolution images as input for systems like face recognition and image classification, we were motivated to develop the proposed system that can recognize a face when input is a low resolution face image and a gallery of high resolution images is available.

1.3 Problem Statement

The present Face Recognition systems fails to recognize the face images of low resolution which in the real-life are common occurrence in surveilllance, distant photography and so on. Due to various circumstances, the use of high definition cameras may not be possible or economical, so often cameras of very limited definitions are used, which produces low quality images that are hard for Face Recognition by the traditional systems.The main aim of the project is to develop a system that can recognize faces at low resolution when a gallery of corresponding high resolution images are available using deep learning. There are many researches and projects done in the field but the project aims to provide a web platform for clients to upload a low resolution image and have it reconstructed to a higher resolution and recognize it by comparing with dataset availabale.

1.4 Objectives

The objectives of this project are:

(a) To reconstuct a higer resolution image from a lower resolution image using CNN model.

(b) To use a face recognition model to classify the input face image.

(c) To validate the recognized faces.

Chapter 2

Literature Review

2.1 Related Works

Low Resolution Face Recognition is a challenging task due to the limited number of pixels available for recognition. There have been many studies on the field of Low Resolution Face Recognition(LRFR), that has experimented different methods from traditional algorithms to state of art deep learning models. The history of low-resolution face recognition dates back to the early 2000s with the development of algorithms and techniques for improving the accuracy of recognition from low-resolution images.

One of the eraliest techniques for low-resolution face recognition was based on the principle of subspace analysis, which involved modeling the image space as a low-dimensional subsapce. Xiaoyang Tan and Bill Triggs[6], in their paper presents the same idea of modeling the image as low dimensional subspace and using principal component analysis(PCA) to reduce the effects of noise and distortions. In the mid 2000s, a number of algorithms were deveoped for face recognition using Gabor wavelets, which are spatial filters that can capture texture and structue of an image. One of those researches was done by the same authors who used subspace analysis. Xiaoyang Tan and Bill Triggs[5] has proposed another technique for Low Resolution Face Recognition(LRFR), which invloves extracting the features of face using Gabor wavelets and matching them with the databases. Similarly, around that same time, Jianchao Yang et.al.[10] proposed a system for low resolution face recognition using sparse representation. This paper presented a technique that involved representing the face images as sparse linear combinations of basis images and using the sparse representation to improve the accuracy of recognition. These techniques and algorithms had dominated the scene of low resolution face recognition for many years. Researchers had even started to combine two or more techniques in hope for better accuracy. S.S. Mahapatra et.al(2014) suggested a hybrid approach for low resolution face recognition which combines the strengths of subsapce analysis and Gabor wavelets.Many other techniques and algorithms like local

binary patterns, dictionary learning, linear dicriminant analysis(LDA) for feature extraction from low resolution image and classification algorithms like support vector machines(SVM). With development in technology of super resolution algorithms, it became imperative to use super resolution techniques and algortihms to increase the accuracy.

Super resolution involves upscaling the low-resolution image to a higher resolution image by predicting missing high-frequency details using a high-resolution training set. Zhang et al.[12] proposed a method for face recognition from low resolution images using super resolution algorithm based on Markov random field. The proposed method involves using MRF to model the relationship between high-resolution and low-resolution face images and then using this model to reconstruct the high-resolution image from the low-resolution input. The reconstructed high-resolution image is then used for recognition. Similarly, Hennings-Yeomans et al.[1] had also proposed a method in which face features extracted for a face recognition are included in super resolution method as prior information in order to provide measure fit of super resolution result from both reconstruction and recognition perspectives.

In mid 2010s, the machine learning models then slowly deep learning models started to dominate the field. Li et al.[3] proposed a low-resolution face recognition method based on deep convolutional neural networks(CNNs) and super resolution. The method involved training a CNN to predict the high resolution image from low resolution input and then using the predicted high resolution image for recognition. Zangeneh et al.[11] had researched on possibility of using two branched CNN architecture where a model using SRCNN and VGGnet was build to recognize the face in low resolution dataset. Along with CNNs, Generative Adversial Networks(GANs) were also popular for the task. Wang et al.[8] proposed a GAN-based approach for low resolution face recognition that involves trainign a generator network to produce high-resolution face images from low-resolution images and the ground truth high-resolution images. The generated high resolution images are then used to train a face recognition model.

Chapter 3

Methodology

3.1 System Block Diagram

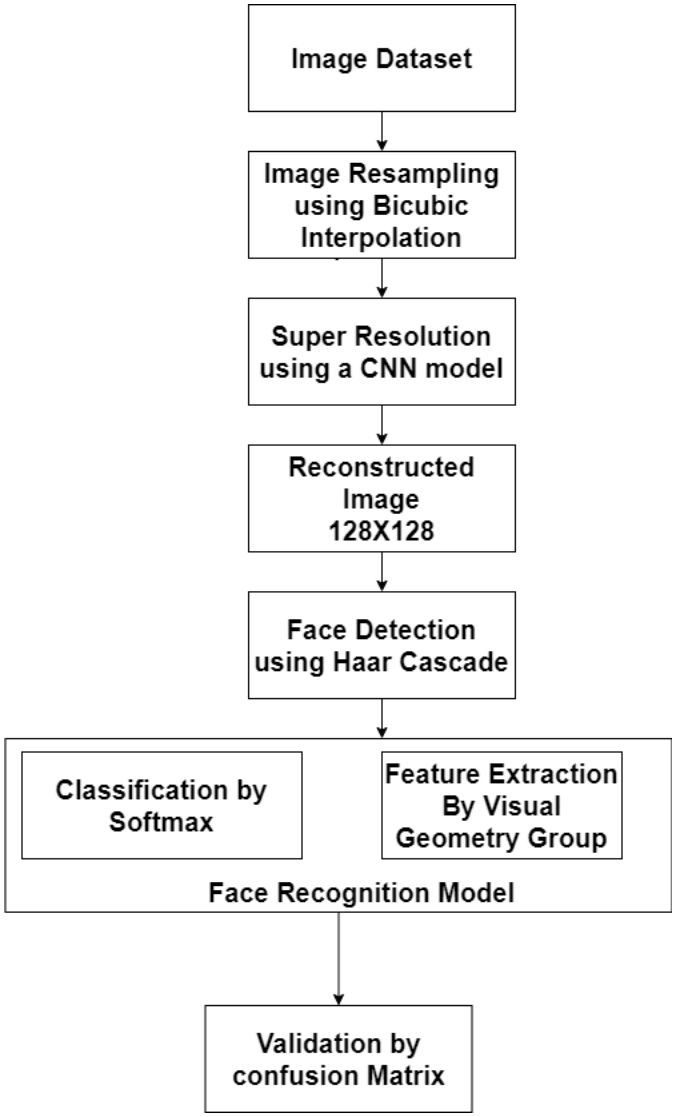

Figure 3.1: System Block Diagram

3.1.1 Data Collection

The images needed for the project were collected using webscraping tools to create a custom celebrity face dataset. For this we simply Created a python script which downloaded the images from google

To open google chrome from vs code, a selenium driver was used in the script. For web scraping an image of particular person, a custom search query with the class name was used. For example if an image of 'Aishwarya Rai' was needed, then a query as link was mentined to the google images. For each new class the query was simply replaced per their name. If a 'Salman Khan' image was to be collected, then aishwarya+rai was simply replaced with salman+khan in the query. This takes us to the image section of google where each img tag with same class name in html tag can be found. So, for data collection, all those images with the img tags were downloaded in the same class. The collected images are then corped out so that only faces are visible and focused.

3.1.2 Image Dataset

In this project, a custom face dataset, Set5 and SuperResDT dataset has been used. SuperResDT dataset has been used for training the super resolution model to reconstruct the low resolution image into high resolution image. Set5 dataset has been used to validate the super resolution model. The custom face dataset has been used for the face recognition purpose with 5 different classes identifying five different people.

SuperResDT dataset is a freely available dataset that is a combination of DIV2K and Flickr2K like famous image dataset with images of different resolutions like 32X32, 128X128, 512X512 etc. It can be easily found in kaggle under the name Single Image Super-Resolution 2022 dataset. It has around 17,455 images across various resolutions. The dataset includes a variety of image types including natuarl scenes,objects and textures. This dataset was used to train, test and validate the super resolution model.

Figure 3.2: SuperResDT dataset for training Super Resolution Model

Set5 dataset is a popular benchmark dataset used for evaluating image super-resolution algorithms. It was first introduced in the paper "Image Super-Resolution Using Sparse Representation and Local Similarity" by Yang and et al[9]. The Set5 dataset consists of five low-resolution images and their corresponding high-resolution versions. The images in the dataset are: a baby, a head, a woman, a bird and a butterfly. This dataset is used to validate the SR model as it is a standard dataset.

Figure 3.3: Set5 dataset for evaluating Super Resolution Model

Custom Face Dataset is a custom dataset consisting around 100 classes each representing a celebrity with total of 10,000 unique images. The dataset was created using web scrapping tools along with face detection and cropping methods. This dataset was used to train and test the face recognition model. It was also used to train and test the SR model.

Figure 3.4: Custom face dataset for training Face Recognition Model

3.1.3 Image Resampling

The project uses images of resoultion 32X32 as low resolution images and images of resolution 128X128 as high resolution. The CNN model uses low resolution(LR) images as input, high resolution(HR) images as target images during training and produce a reconstructed HR image of input LR image. The same reconstructed image is then used as input for the face recognition model. So, the images in dataset needs to be resampled into LR images of 32X32 resolution and HR images of 128X128 resolution. For image resampling, bicubic interpolation is used.

Bicubic interpolation is a very popular technique for resampling digital images that involves using a mathematical function to estimate the pixel values of a higher-resolution image from the pixel values of a lower-resolution image. It is performed by using a kernel interpolation equation:

$$h(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \leq |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases} \tag{3.1}$$

Here, the value of coefficient 'a' determines the performance of the kernel and is taken between -0.5 to -0.75.

3.1.4 Super Resolution by CNN

CNN have been used for many applications and super resolution is one of them. The CNN achieves super resolution by trying to learn the inverse function of the downsampling filter to restore the lost details of the HR images.Residual Network (ResNet) is a type of CNN

architecture that has been successfully used for various computer vision tasks, including image super resolution. In a ResNet model, residual connections are added between the layers, allowing the network to learn residual mappings between the input and output images.The residual connection allows the model to learn the difference between the low-resolution input image and the high-resolution output image, which is the key to successful super resolution. The residual connection also helps to mitigate the vanishing gradient problem, which is common in deep neural networks.

A ResNet-based super resolution model consists of an encoder network, a series of residual blocks, and a decoder network. The encoder network takes the low-resolution input image and generates a set of feature maps. These feature maps are then passed through a series of residual blocks, each consisting of several convolutional layers with residual connections. The decoder network then takes the output from the residual blocks and upscales the feature maps to generate the high-resolution output image. The upscaling is typically performed using transposed convolutional layers or some other form of interpolation.

In Single Image Super Resolution(SISR) like this model, the aim is to estimate a high-resolution, super-resolved image $\mathrm{I^{SR}}$ from a low-resolution input image $\mathrm{I^{LR}}$. The high resolution imagers are only available during training. The $\mathrm{I^{LR}}$ are obtained from $\mathrm{I^{HR}}$ using bicubic interpolation with downsampling factor of r. The goal is to train a generating function G that estimates for a given LR input image its corresponding HR counterpart. To achieve this,we train the Residual network as feed-forward CNN G_θ parametrized by θ_G where θ_G denotes the weights ad bises of L-layer deep network which is obtained by optimizing super-resolution specific loss function $\mathrm{l^{SR}}$. For training images $\mathrm{I^{HR}{}_n}$ with corresponding $\mathrm{I^{LR}{}_n}$, we solve:

$$\theta_\mathrm{G} = argmin_{\theta_\mathrm{G}} \frac{1}{N} \sum l^{\mathrm{SR}}(G_{\theta_\mathrm{G}}(I^{\mathrm{LR}}{}_{,\mathrm{n}}), I^{\mathrm{HR}}{}_{,\mathrm{n}}) \tag{3.2}$$

In this work, a content loss $\mathrm{I^{SR}}$ has been used which denotes pixel wise MSE loss.

Content Loss

The super resolution residual network models use Mean Squared Error(MSE) loss as primary objective function to train the model. The MSE loss measures the average squared difference between the predicted high-resolution image and the ground truth high-resolution image. Durin training, the network learns to minimize the MSE loss by adjusting its weights and biases to produce better high-resolution image predictions. The pixel wise MSE loss is calculate as:

$$l^{\mathrm{SR}} = \frac{1}{r^2WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I^{\mathrm{HR}}{}_{\mathrm{x,y}} - G_{\theta_\mathrm{g}}(I^{\mathrm{LR}}{}_{x,y}))^2 \tag{3.3}$$

Pixel Shuffle

Pixle shuffle layer is one of the most importnat layer in super-resolution process by CNN as it alloows the network to upscale the low resolution image while preserving the high frequency details in the image. Pixel shuffle transformantion re-organizaes the low-resolution image channels to obtain a bigger image with few channels. Pixel shuffling involves rearranging the elements of a feature map, so that groups of neighbouring pixels are stacked into new feature maps with higher spatial resolution. This operation allows for the synthesis of a higher-resolution image from a lower-resolution image, by using the information from the low-resolution image to generate new high-resolution pixels. In SRResNet, pixel shuffling is used in the last layer of the generator network to produce the final high-resolution output image. Without pixel shuffling, the network would only be able to produce a blurred version of the low-resolution input image. Thus, pixel shuffling is a critical component in the SRResNet architecture, which enables it to achieve good performance in image super-resolution.

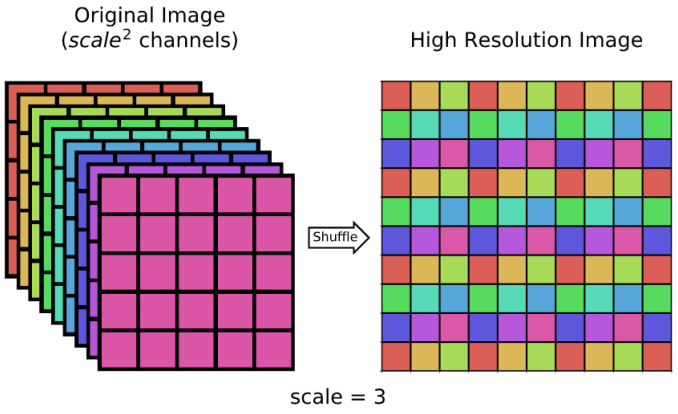

Figure 3.5: Pixel Shuffling for Super Resolution

On the left the input image with scale2 (= 9) channels. On the right the result of Pixel Shuffle transformation.

Peak Signal to Noise Ratio(PSNR)

PSNR is a quantitative measure of the fidelity of an image, which compares the original (ground truth) image to a processed image by measuring the average squared difference between the pixel values of the two images. The higher the PSNR value, the higher the fidelity of the processed image to the original image.The PSNR value is expressed in decibels (dB) and is defined as:

$$PSNR = 20 * log_{10}(MAXp) - 10 * log_{10}(MSE) \tag{3.4}$$

Where, MAXp is the maximum possible pixle value and MSE is the average squared difference between the pixel values of the original and processed images.

Structural Similarity Index(SSIM)

Unlike PSNR, which only considers pixel-wise differences between images, SSIM takes into account the structural information and perceptual features of images.SSIM ranges from -1 to 1, with 1 indicating perfect similarity and 0 indicating no similarity. Higher SSIM values indicate higher similarity between the two images being compared. SSIM is defined as:

$$SSIM(x, y) = \frac{(2 * \mu_x * \mu_y + c1)(2 * \sigma_{xy} + c2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \tag{3.5}$$

where x and y are the two images being compared, μx and μy are the mean values of x and y, respectively, σ_{x} and σ_{y} are the standard deviations of x and y, respectively, σ_{xy} is the covariance between x and y, and c1 and c2 are small constants added to prevent division by zero.

3.1.5 Feature Extraction by CNN

CNNs are the feed forward neural networks made up of many hidden layers.The project uses a Visual Geometry Group(VGG) CNN model called VGG16 to extract the features from the images. CNNs consist of filters or kernels or neurons that have learnable weights or parameters and biases. Each filter takes some inputs and does convolution. The components of CNN consist of following layers:

- Convolution layer
- Rectified Linear Unit(ReLU) layer
- Pooling layer
- Adam Optimizer
- Fully Connected Layer

Convolutional Layer

This layer is the core building block of a convolutional network that performs most of the computational heavy lifting. Its primary purpose is to extract features from the input data which is an image. Convolution preserves the spatial relationship between pixels by learning features using small squares of input image. This produces a feature map or activation map in the output image and after then feature maps are fed as input data to the next convolutional layer. A convolution is done by multiplying a pixel's and its neighboring pixels color value by a matrix (kernel). The convolution formula is defined as below:

$$y[m, n] = x[m, n] * h[m, n] = \sum_j \sum_i x[i, j]h[m - i, n - j] \tag{3.6}$$

where, y is the convolved featuremap,x is the input image and h is a kernel.

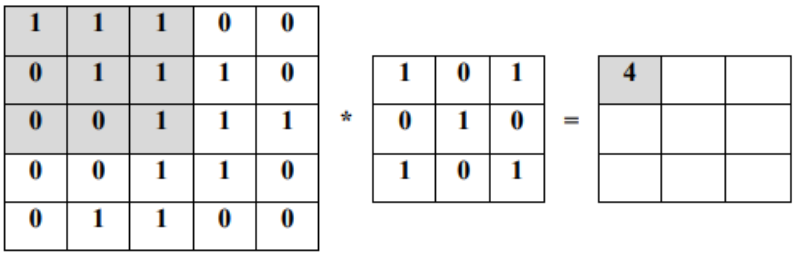

Figure 3.6: Illustration of Convolution operation in image processing

ReLU Layer

It is a non-linear operation similar to the rectification. It is an element wise operation that reconstitutes all negative values in the feature map by zero. The equation of ReLU operation is defined below:

$$f(x) = max(0, x) \tag{3.7}$$

where, x is the value in feature map

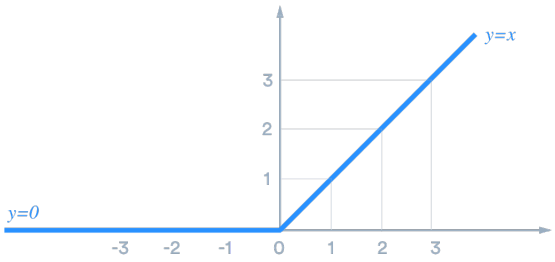

Figure 3.7: Graphical representation of the ReLu function

Pooling Layer

This layer reduces the dimensionality of each activation map and continues to have the most important information. The input images are divided into a set of non-overlapping rectangles. Each region is down-sampled by a non-linear operation like average or maxima. This layer gains better generalization, faster convergence, robust to translation and distortion and usually placed between convolutional layers.

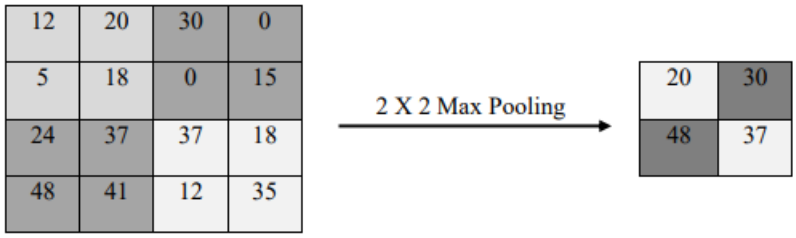

Figure 3.8: Illustration of Max Pooling in Image Processing

Fully Connected Layer

This indicates that every filter in the previous layer is connected to every filter in the next layer. The output from the convolutional, pooling and ReLU layers are embodiments of high-level features of the input image. Using fully connected layer employs these features for classifying the input image into various classes based on training set. Fully connected layer is the final pooling layer feeding the features to a classifier that uses Softmax activation function. For the purpose of features extraction from the input images, a variant of Convolutional Neural Network called VGG16 architecture has been used. VGG16 usually refers to a convolutional network for object recognition developed and trained by Oxford's renowned Visual Geometry Group (VGG), which achieved very good performance on the ImageNet dataset.

For extracting features, the last fully connected layers have been excluded. In VGG16 architecture, there are 13 convolutional layers with ReLU activation function with stride of 1 and 5 MaxPooling layers with stride of 2. All the Conv layers have one padding and MaxPooling layers have zero padding. The feature map of last Convolutional Layer with feature map 512 is taken and fed to region proposal network. As the SoftMax output of head network is not taken it is not necessary to convert the images to size of 224x224.

Adam Optimizer

Adam (Adaptive Moment Estimation) is a popular optimization algorithm used in deep learning for minimizing the loss function during training. Adam is an extension of stochastic gradient descent (SGD) that uses a combination of adaptive learning rates and momentum updates to improve the efficiency and effectiveness of the optimization process. Adam maintains an exponentially decaying average of the past gradients and their squares, which are used to adaptively adjust the learning rates for each parameter in the model. The update rule for each parameter at time step t is given by:

$$
\begin{aligned}
v_t &= \beta 1 * v_{t-1} + (1 + \beta 1) * g_t \\
s_t &= \beta 2 * s_{t-1} + (1 + \beta 2) * g_t^2 \\
v_{t_{hat}} &= v_t / (1 - \beta 1^t) \\
s_{t_{hat}} &= s_t / (1 - \beta 2^t) \\
\theta_t &= \theta_{t-1} - \alpha * v_{t_{hat}} / (sqrt(s_{t_{hat}}) + \epsilon)
\end{aligned}
\tag{3.8}
$$

where g_t is the gradient of the loss function with respect to the parameters at time step t, α is the learning rate, $\beta 1$ and $\beta 2$ are exponential decay rates for the first and second moments of the gradients, and ϵ is a small constant added to prevent division by zero. The

first moment estimate v_t is an exponentially weighted average of the past gradients, and the second moment estimate s_t is an exponentially weighted average of the past squared gradients. These estimates are used to update the parameters theta, where the learning rate is adaptively scaled by the ratio of the root mean squared estimate of the second moment to the first moment estimate.

3.1.6 Softmax Classification

The Softmax function gives the outputs of each unit to be between 0 and 1. It also divides each output such that the total sum of the outputs is equal to 1. Mathematically, the Softmax function is shown below, where z is a vector of the inputs to the output layer (having 10 output units, then there are 10 elements in z). And again, j indexes the output units, so j = 1, 2, ..., K.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{N} e^{z_k}} \tag{3.9}$$

Similarly, for the classification process the classification loss is the cross-entropy loss which is calculated as:

$$-\sum_{c=1}^{M} y_{o,c} log(p_{o,c}) \tag{3.10}$$

where, M is the number of classes, y is a binary indicator(0 or 1) if class label c is the correct classification for observation o and p is the predicted probability observation o is of class c. The method takes in feature maps extracted by the CNN model and uses three fully connected layers of size 4096 neurons and ReLU activation with final layer of N + 1 unit where N is the total number of classes that extra one for background class.

3.1.7 Face Detection by Haar Cascade

Haar Cascade is a machine learning object detection algorithm used to identify objects or features in an image or video. It was developed by Viola and Jones in 2001 and is based on the concept of Haar-like features. These features are rectangular boxes with different values in each region of the box, which are used to identify the object of interest. The project uses Haar Cascade to detect face in the images which then can be recognized using the face recognition model.The Haar Cascade algorithm works by training a classifier using positive and negative images. The positive images contain the object of interest, while the negative images do not. The algorithm then searches for the object of interest in new images by sliding a window over the image and evaluating the Haar-like features at each position. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features"[7] published in 2001.

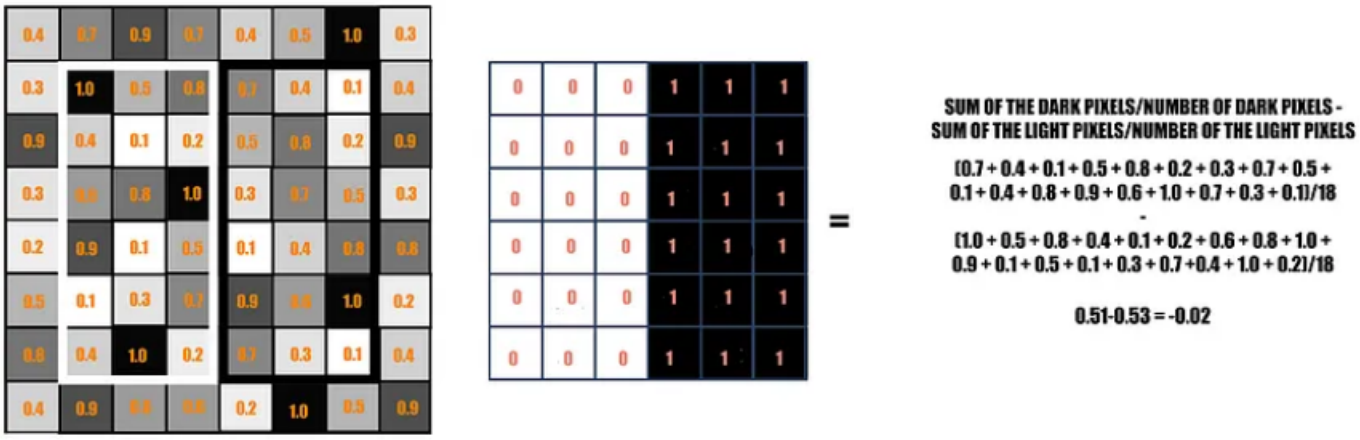

Figure 3.9: Working of Haar Cascade

The rectangle on the left is a sample representation of an image with pixel values 0.0 to 1.0. The rectangle at the center is a haar kernel which has all the light pixels on the left and all the dark pixels on the right. The haar calculation is done by finding out the difference of the average of the pixel values at the darker region and the average of the pixel values at the lighter region. If the difference is close to 1, then there is an edge detected by the haar feature.

3.1.8 Validation by Confusion Matrix

For the validation and performance evaluation of the model, Confusion Matrix is used. From the confusion matrix accuracy, precision and recall are calculated. Confusion Matrix gives a matrix as output and describes the complete performance of the model.

	True diagnosis		
	Positive	Negative	Total
Positive	TP	FN	$TP + FN$
Negative	FP	TN	$FP + TN$
Total	$TP + FP$	$FN + TN$	

Table 3.1: Table showing the confusion matrix

- True Positives(TP): The cases in which predicted value is True and the actual output is also True.
- True Negatives(TN): The case in which predicted value is True and the actual output is False.

- False Positives(FP): The cases in which predicted value is True and the actual output is False.
- False Negatives(FN): The cases in whihc predicted value is False and the actual output is True.

The Confusion Matrix Parameters for model evaluation are as fllows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.11}$$

$$Precision(Exactness) = \frac{TP}{TP + FP} \tag{3.12}$$

$$Recall(Completeness) = \frac{TP}{TP + FN} \tag{3.13}$$

$$TruePositiveRate(TPR) = Sensitivity = \frac{TP}{TP + FN} \tag{3.14}$$

$$FalsePositiveRate(FPR) = 1 - Specificity = \frac{FN}{TN + FN} \tag{3.15}$$

Chapter 4

Result and Discussion

In this section, we discuss the result of training and testing of the models on the datsets mentioned before. SuperResDT dataset with 17,400 images, a custom celeb face dataset of around 13,000 images, set5 dataset to evaluate the standard of super resolution model and a dataset of reconstructed images by SR model for custom celeb face dataset were used to draw various results.

4.1 Super Resolution

Super resolution is a major part of the project. The super resolution is responsible for upscaling the low resolution images into high resolution reconstructed image using which features for the recognition face are extracted. Two metrics PSNR and SSIM were used to evaluate the SR model. The PSNR value for the reconstructed image was found to be 28.0231 dB and SSIM was 0.6678 for 200 epochs of training of model on SuperResDT dataset.
From the table 5.1, we can clearly see that it has highest PSNR value when custom face datset of around 13,000 images is used with the SSIM value 0.7686. Similarly, when used to evaluate the set5 validation dataset it resulted in 28.9310 dB average PSNR and 0.7256 average SSIM.

Dataset	No. of Images	PSNR Value	SSIM Value
SuperResDT	17,400	28.0231	0.6678
Custom Face Dataset	13,000	29.3256	0.7686

Table 4.1: Table showing the PSNR and SSIM value for different dataset

The following shows the result of images in set5 dataset.

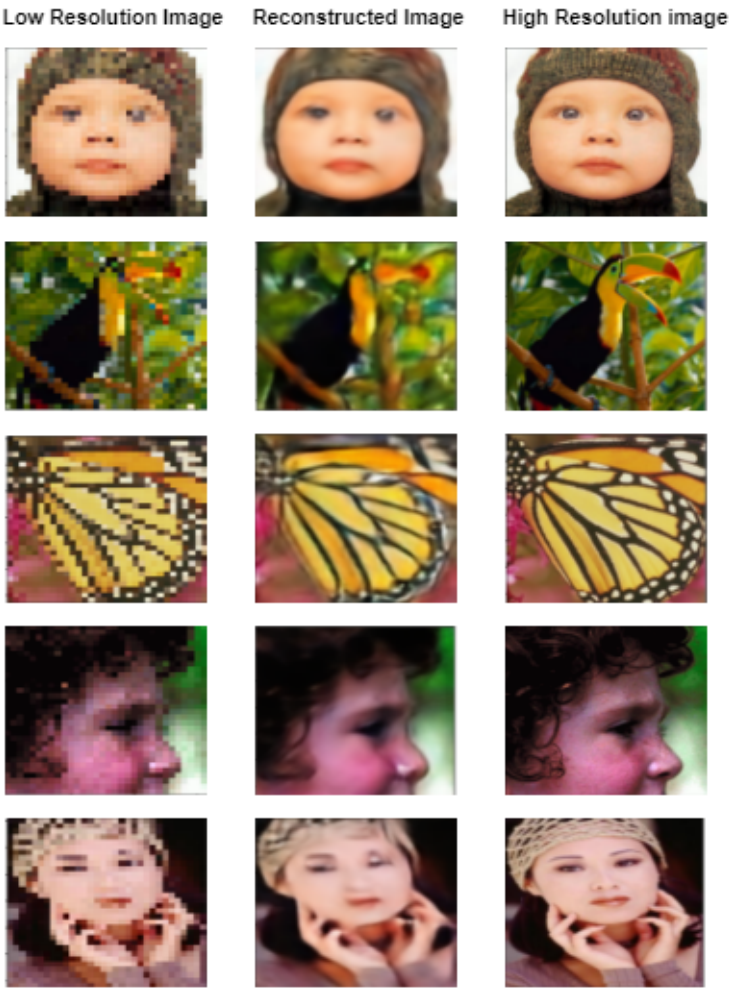

Figure 4.1: SR result for set5 dataset

For the five images, the model had average PSNR of 28.9310 dB and SSIM of 0.72. The butterfly iamge and the woman image had the least PSNR as they had more details which the model could not retrieve due to being low resolution.

Similarly, the face images from the custom face dataset had also been tested. The images had the highest PSNR value of 29.3256 with SSIM value of 0.7686. Some of the results are shown below:

Figure 4.2: SR result for custom face dataset

The figure 4.2 shows some result of SR model on faces. The above figure shows a group of three images together: the left most being the low-resolution image of 32X32, the center one being the reconstructed image by the SR model and the right most being the original HR image of size 128X128.

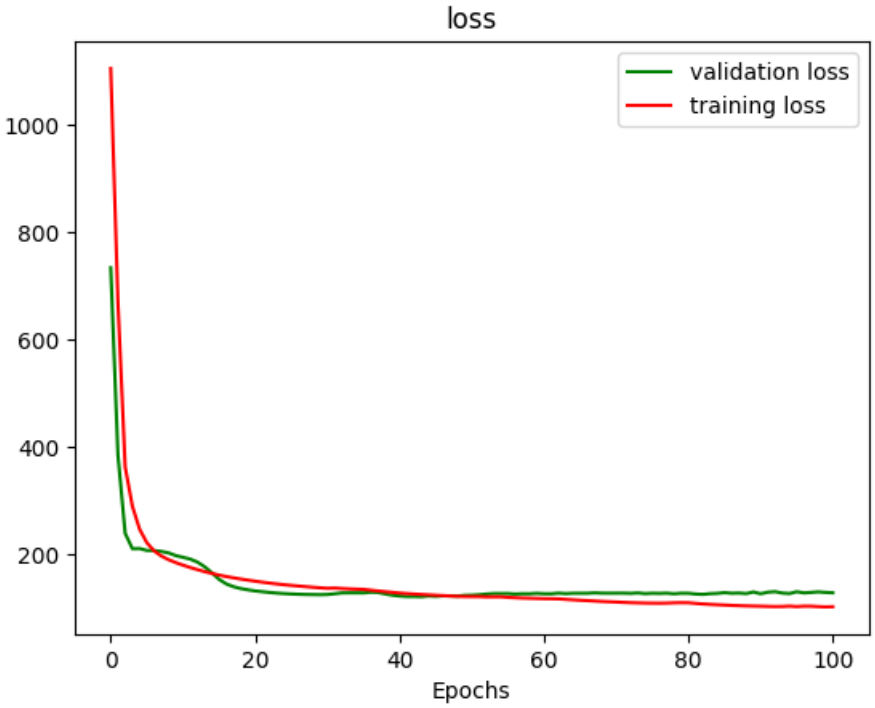

Figure 4.3: Training and Validation loss for SR model

The figure 4.3 show the loss and validation loss of the model. In the model, Mean Squared Error(MSE) is used as loss function. The loss graph was seen generally declining across 100 epochs as it should. The validation loss unlike trianing loss has some fluctuations.

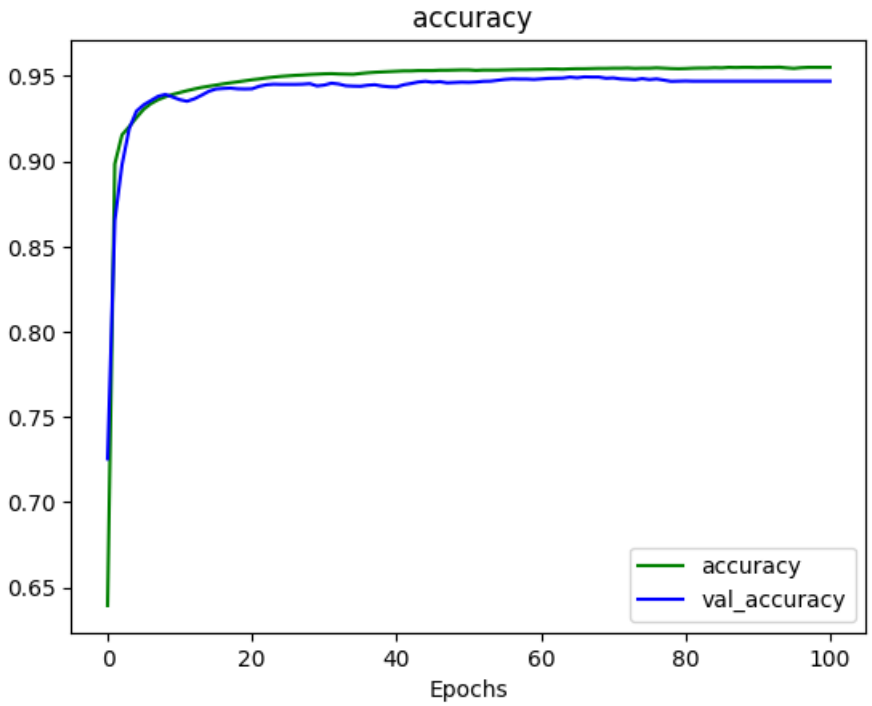

Figure 4.4: (a) Training and Validation accuracy for SR model

The figure 4.4 shows the training accuracy and validation accuracy of the SR model. The accuracy of the model is The accuracy of the model is measured using the metric of Peak Signal-to-Noise Ratio(PSNR) and Structural Similarity Index(SSIM). The model creates a super resolved image from a low resolution image with accuracy of 95.0%.
For the validation of the data, a different standard dataset called set5 is used. with set5 dataset a PSNR of 28.9310 dB and SSIM of 0.72

4.2 Face Recognition

The face recognition is the next part of the project. Once the image is super resolved using SR model, the reconstructed image is used as input for the feature extraction by another CNN model of VGG architecture called VGG16. For face recognition purposes 10 classes were used, "aishwarya rai","alia bhatt", "aamir khan", "anuska sharma", "bishwash", "hrithik roshan", "priyanka chopra", "ranbir kapoor", "salman khan" and "shahruk khan" i.e. faces of these people were used in the model thus model can recognize their faces. Any face outside these classes would be termed as 'Cannot Recognize'.

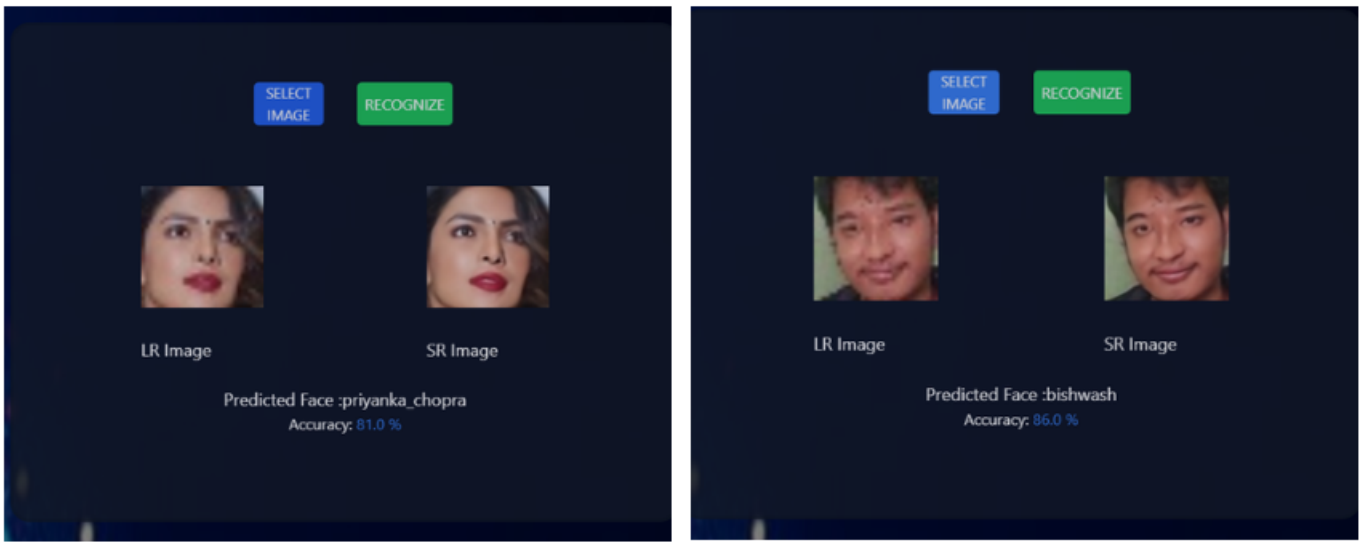

Figure 4.5: Images showing the Face Recognition model recognizing the faces

The figure 4.5 shows the sinlge image face recognition. As the faces were recognized by the model, it shows up their name.

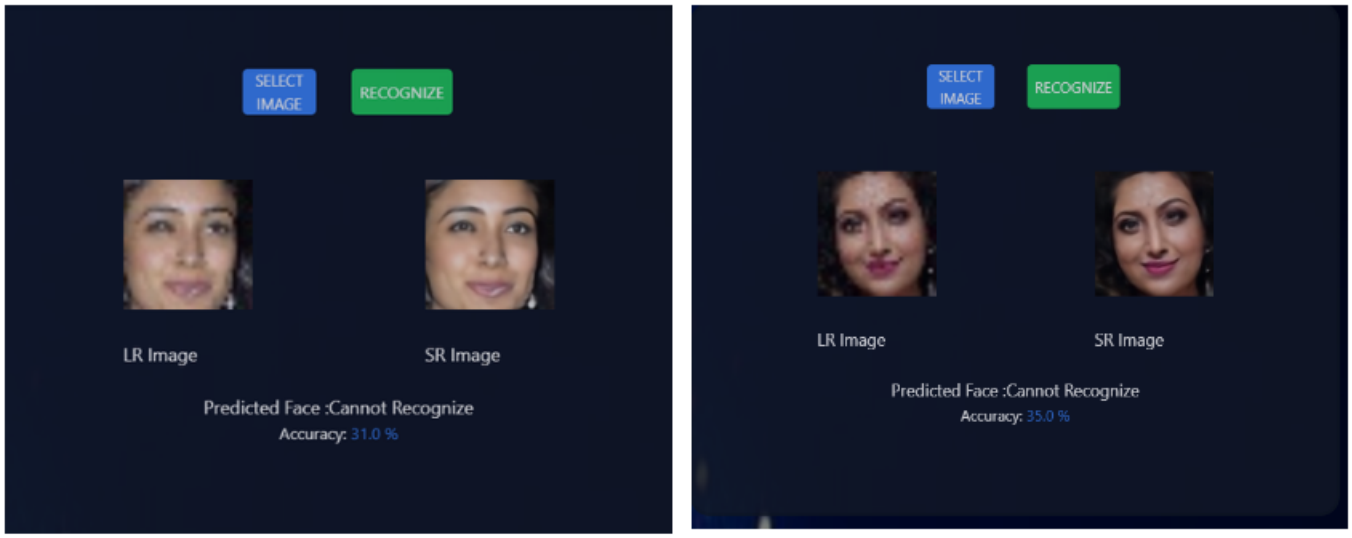

Figure 4.6: Images showing the Face Recognition model being unable to recognize the face

The figure 4.6 shows face images being not recognized. Softmax classification is based on probability distribution, so a threshold is set and any image that model finds has accuracy below threshold will not be recognized.

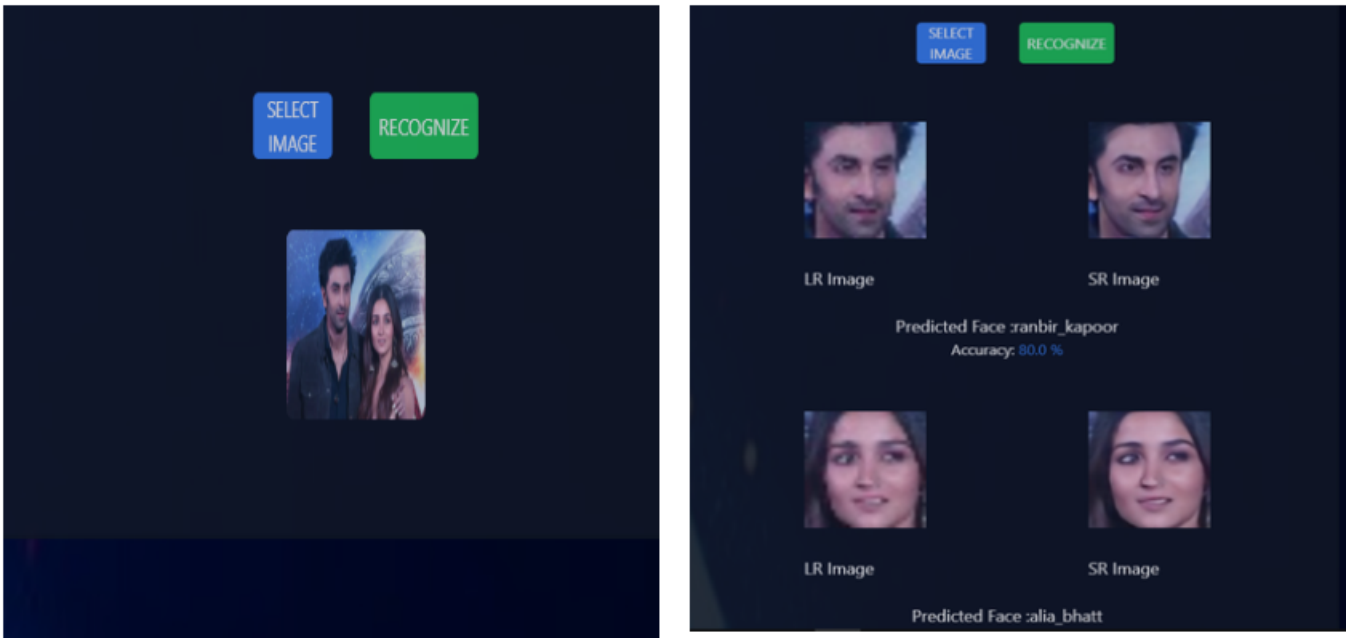

Figure 4.7: Image showing multiple face detection and recognition

The above figure 4.7, shows the situation when an image with multiple faces is used in the application. The application will first use Haar Cascade model to detect multiple faces in th eimage then application will corp them out super resolve them and recognize them using Face Recognition model.

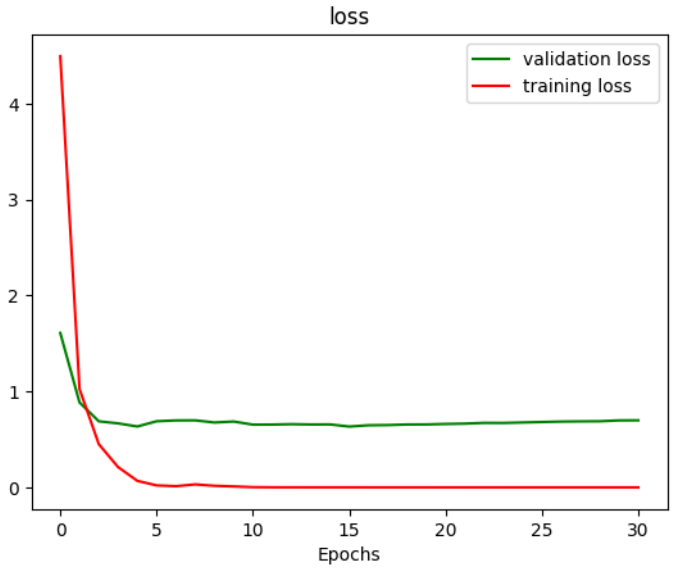

Figure 4.8: Graph showing loss and validation loss value of the FR model

Figure 4.8 shows the classification loss in training and validation process while training the model. The model uses categorical crossentropy loss, which measures the difference between the predicted probability distribution and the actual probability distribution of the target classes.

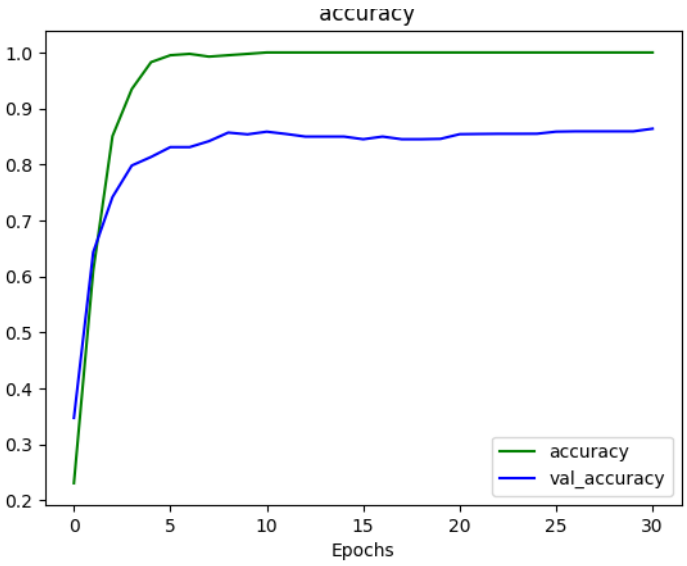

Figure 4.9: Graph showing accuracy and validation accuracy value of the FR model

Figure 4.9 shows the accuracy metric for the FR model.The accuracy of model is 86.32%

Precision, recall and F1 score for the evaluation of the model were used. The values for each class can be seen in following table 4.2. Precision value shows the exactness of classifier and recall value shows the completness of classifier. In table 4.2, we can see that 'salman khan' has the highest precision and 'aamir khan' has the least amount of precision. Similarly, 'bishwash' has the highest recall and 'aishwarya rai' and 'salman khan' has the lowest recall.

class	Label	Precision	Recall	F1 Score
0	aishwarya rai	0.920	0.622	0.742
1	alia bhatt	0.776	0.900	0.833
2	aamir khan	0.720	0.831	0.771
3	anushka sharma	0.792	0.934	0.857
4	bishwash	0.909	0.938	0.923
5	hrithik roshan	0.774	0.857	0.814
6	priyanka chopra	0.875	0.656	0.750
7	ranbir kapoor	0.794	0.909	0.847
8	salman khan	0.933	0.622	0.747
9	shahruk khan	0.840	0.712	0.771

Table 4.2: Precision, Recall and F1 Score for classes used in FR Model

		PREDICTED CLASSES									
		0	1	2	3	4	5	6	7	8	9
ACTUAL CLASS	0	23	2	2	2	1	2	2	0	1	2
	1	0	45	0	4	0	0	1	0	0	0
	2	0	0	54	1	0	3	0	6	1	0
	3	0	2	0	57	1	0	0	0	0	1
	4	0	2	0	0	30	0	0	0	0	0
	5	0	0	5	1	0	48	0	2	0	0
	6	2	4	2	1	0	0	21	0	0	2
	7	0	0	3	0	0	0	0	50	0	2
	8	0	0	6	1	1	4	0	4	28	1
		0	3	3	5	0	5	0	1	0	42

Figure 4.10: Confusion Matrix of the classes

4.3 Systems Accuracy

For checking the efficiency of pipeline system, the models were trained with different available dataset and their corresponding accuracy were observed. A set of 100 low resolution images was used as input for the given combinations.

The SR model was once trained with SuperResDT dataset and FR model with custom face dataset which was only accurate for around 72 images out of 100. Similarly, the SR model was trained with Custom Face Dataset and FR model with it too which did increase the accuracy to around 75%.Finally, a combination of face dataset for SR model and a dataset of the face dataset but with output from the SR model. This combination turned out to be the most succesful of all with 82.43

Super Resolution Model	Face Recognition Model	Accuracy(%)
SuperResDT Dataset	Custom Face Dataset	72.63
Custom Face Dataset	Custom Face Dataset	75.36
Custom Face Dataset	SR Output Dataset	82.43

Table 4.3: Table showing accuracy of system with different combination of dataset

The pipeline system had the most accuracy when the face recognition model had been trained on the dataset created using outputs from Super Resolution model.

4.4 Discussion

The accuracy of the system as whole to recognize a low resolution image is quite decent. The PSNR and SSIM value for the Super resolution model is also good enough to create face image with features enough to recognize the face. The accuracy of the method can be further increased when the greater number of training images in different lighting conditions and physical appearances can be used. The PSNR value for custom face dataset was 29% and the face recognition model had accuary of 86.32%. The web application asks the user for a input of low resolution image which at first is reconstructed into a higher resolution image and then is recognized using the face recognition model. The web application can also detect multiple faces in a single image using Haar Cascade. The face detected then can be recognized using the face recognition model.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

This work aims to improve the face recognition of low resolution images by using a pipeline model of first upsmapling the low resolution image to high resolution image and then using the newly reconstructed high resolution image as input for feature extraction model and classify it. The residual network CNN model takes 32X32 image as an input and produces a 128X128 HR image as output which is fed to face recognition model. The model extracts the features, pools them and finallyuse softmax classification to classify the image.

In this work, the performance of the Super resolution model is evaluated using set5 standard dataset while the face recognition dataset is evaluated using custom face dataset. Metrics like PSNR and SSIM are used for evalutating the super rsolution model's performance. Furthermore,the obtained results from the face recognition modelare validated from the evaluation metrics of confusion matrix. The accuracy of face recognition for classifying the face images is 86.32% for the custom face dataset.

5.2 Limitations

The project only take fixed size of images as low resolution image and high resolution image. The images used are only cropped out face images, and in case of presence of noise and other artifacts in the image the classifier may not be able to predict the image or may not be able to recognize it. The SR model used is only good for moderate upscaling(2x,4x), for extreme upscaling like 8x and 16x the model may not be applicable. The quality of the super-resolved image could significantly affect the face recognition accuracy. If the super-resolved image is not of good quality, the recognition system may fail to recognize the face accurately. The SRResNet works only on spatial resolution so the image still may have lesser spectral resolution.

5.3 Recommendation

The accuracy and recognition rate can be increased by making some improvements.Some of the classification errors are due to error in dataset. This project can be extended to recognize images of other resolution too and can also be mended in such a way that it can detect faces in real time and crop them out by itself.

Bibliography

[1] P. H. Hennings-Yeomans, S. Baker, and B. V. Kumar. Simultaneous super-resolution and feature extraction for recognition of low-resolution faces. In *2008 IEEE Conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[2] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] P. Li, L. Prieto, D. Mery, and P. J. Flynn. On low-resolution face recognition in the wild: Comparisons and new techniques. *IEEE Transactions on Information Forensics and Security*, 14(8):2000–2012, 2019.

[4] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[5] X. Tan and B. Triggs. Face recognition from low-resolution images using gabor wavelets. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–289. IEEE, 2005.

[6] X. Tan and B. Triggs. Face recognition from low resolution images using subspace analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):649–62, 2007.

[7] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[8] S. Wang, Y. Zhu, Z. Liu, C. Liu, W. Zuo, and L. Zhang. Low-resolution face recognition via learning deep discriminative representations with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2380–2389, 2018.

[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[10] J. Yang, Z. Zhou, and Y. Ma. Low-resolution face recognition via sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1745–1752. IEEE, 2011.

[11] E. Zangeneh, M. Rahmati, and Y. Mohsenzadeh. Low resolution face recognition using a two-branch deep convolutional neural network architecture. *Expert Systems with Applications*, 139:112854, 2020.

[12] X. Zhang, X. Gao, and X. Li. Face recognition from low resolution images using super-resolution based on markov random field. In *2011 International Conference on Computer Vision*, pages 853–858. IEEE, 2011.

Appendix

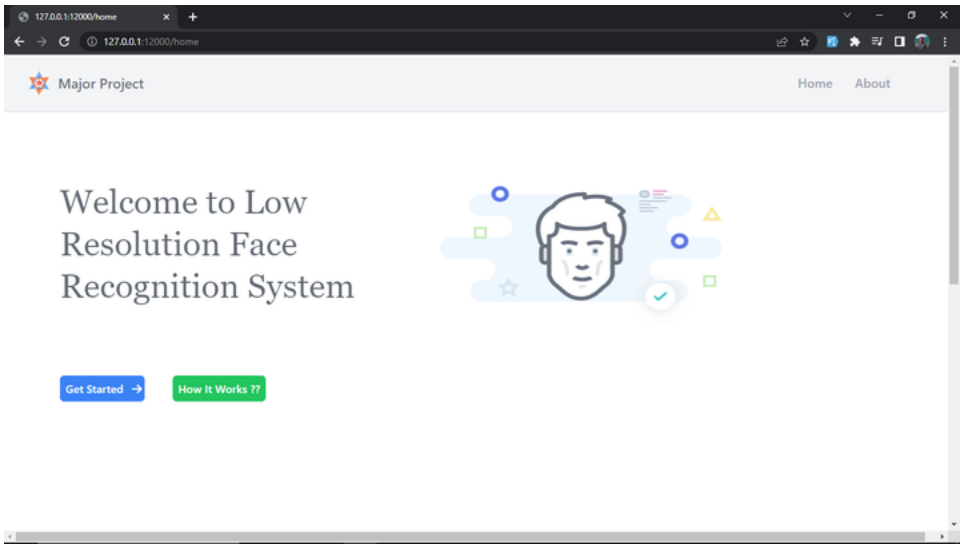

Figure 5.1: Homepage of our web application

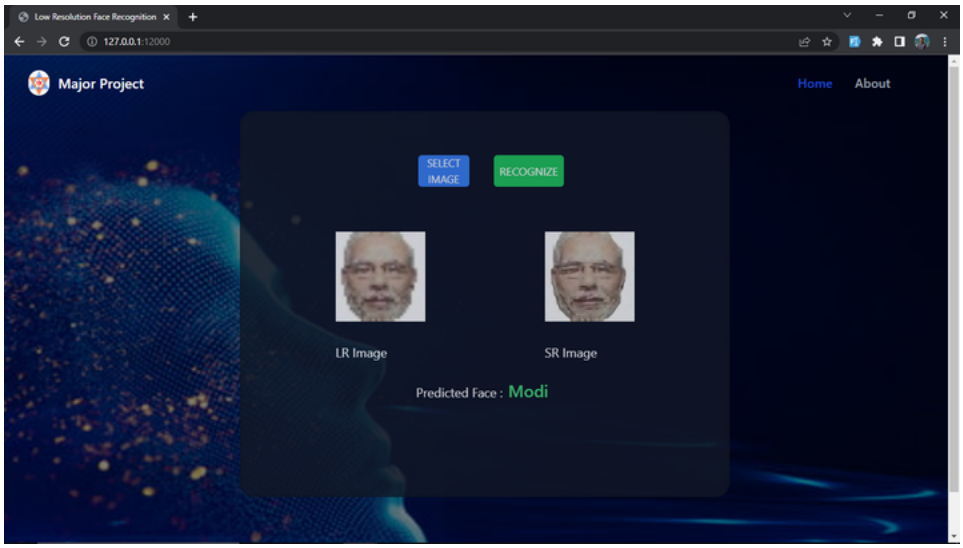

Figure 5.2: webpage of the web application

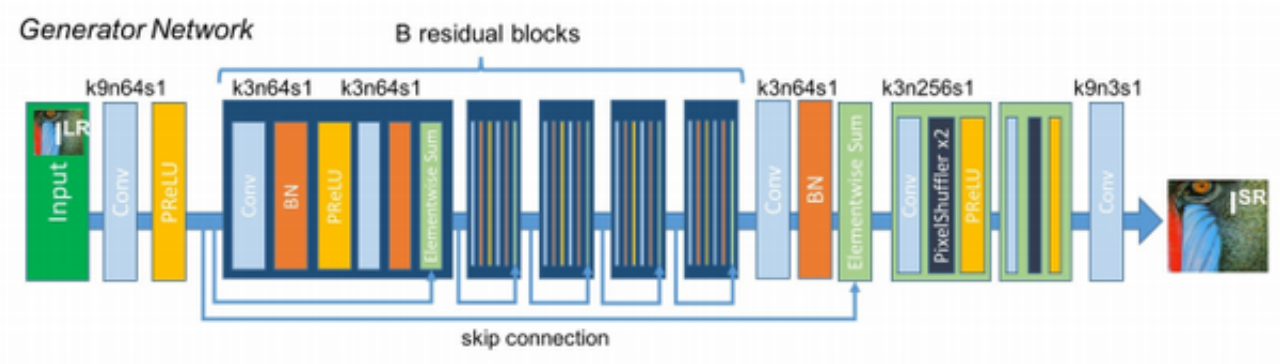

Figure 5.3: Architecture of SRResNet

Figure 5.4: Architecture of VGG16