IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 7, July 2021

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

Developing Application for Converting Grayscale Sketches Images into Colorful Image Using DCNN

Naumanurrahman Shaikh Mujiburrahman, Shreya Kiran Mahajan, Kazi Nausheen Firdous Nafees, Jagruti Rajendra Patil, Ashwini Bajirao Kapse

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India

ABSTRACT: Lot of work is going on in the field of the sketch to image conversion using Deep Convolutional Neural Network. In this paper we have successfully created an application which provide the way for forensic researchers so they can upload a sketch and get the colorful image of that photo we have also successfully implemented an algorithm using Convolutional Neural Network which provides the most matching image of the culprit which matches with more accuracy. This application provide a face-recognition algorithm Using KNN that provides security for the use of application and database. This application is useful in many from face creation from a sketch to face recognition.

KEYWORDS: DCNN, KNN, GrayScale, Numpy Array, Loss , Training, Regression, Face Recognition, Face Lock, Tkinter

I. INTRODUCTION

In the world of forensics, forensic labs gets the sketches of potential culprits which are the grayscale images those can be hard to use in face recognition algorithm due to lack of images features in the images. By this algorithm we can convert the sketch in to images which then can be used for features extraction and those features can be used for face recognition on the dataset we have available of culprits and give the closest matching image. This algorithm is developed by the Rejin Joy* Electrical and Computer Engineering University of Florida Rishabh Singh* Electrical and Computer Engineering University of Florida. This algorithm uses the Deep Convolutional Neural Networks for the converting the grayscale image into the colorful images which highlight the Image features.

The aim of the application we have created is to provide an end to end process for converting grayscale images into colorful images. The Idea behind creating this application is to provide the method for conversion which can also be done by the person with no knowledge of the programing. We have also provided the security using the face lock system that is the most efficient way to stop any intruder to use the application or to access the database of application. The further process in the application is that it provides a face recognition feature also which takes the image and search with that image in the dataset.

A. Convolutional

While programming with a Convolutional Neural Network, the input given is a array(tensor) with shape (number of images) x (image width) x (image height) x (image depth). Then after passing through a convolutional layer, array becomes abstracted to a feature map, with shape (number of images) x (feature map width) x (feature map height) x (feature map channels). A convolutional layer within a neural network must have these following attributes:

- Convolutional kernels are defined by width and height.
- The number of input channels.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

- The number of output channels.
- The depth of the Convolution filter (the input channels) must be equal to the number of channels (depth) of the input feature map.
- Activation Function

B. Weights

Each neuron during a neural network computes an output value by applying a selected function to the input values coming from the receptive field within the previous layer. The function that's applied to the input values is decided by a vector of weights and a bias (typically real numbers). Learning, during a neural network, progresses by making iterative adjustments to those biases and weights. The vector of weights and thus the bias are called filters and represent particular features of the input (e.g., a specific shape).

C. Activation Functions

Activation layers apply a non-linear operation to the output of the opposite layers like convolutional layers or dense layers.

• ReLu Activation: ReLu or Rectified linear measure computes
the function $f(x)=max(0,x) to threshold point when the activation is 0.

• Linear Activation: It works as a linear regression output, It calculates the output using the linear parameters as in Linera Regrssion.

D. Optimizer

• Adam (Adaptive moment estimation): is an update to RMSProp optimizer in which the running average of both the gradients and their magnitude is used. In practice Adam is currently recommended as the default algorithm to use, and often works slightly better than RMSProp. In this implementation Adam also shows general high accuracy. We've used Adam in this implementation because we were using the mean square error as metric and Adam optimizer works well with the mean square error.

E. K Nearest Neighbor Algorithm

It is a supervised Machine Learning algorithm which is used for the classification it calculates the Euclidean Distance between two features and classifies or label the object to the label of most nearest neighbors Labels.

II. METHODOLOGY

A. DCNN

A Deep Convolutional Neural Networks is the neural network used for complex calculations. In our project we use the DCNN to convert the numpy array of the grayscale image into the numpy array of the colored image which involves the heavy matrix calculation done through the network. In input layer we provide the matrix or numpy we have created our training sketches images the weights are randomly initialized and learning rate is set, we have used the means square error to find the difference between output we want and output we have. The Neural Network Performs Linear calculation on the array and produces a output array in feed forward process that we used to calculate the error between the Expected and real output this error and learning rate then used for backward propagation to update the weights of the Neural Network to minimize this error this process is repeated for 10 epochs and sample each image we have total of 72000 images, finally we stop for the minimum error high accuracy in our output we have got. This a Regression Problem.

Network Architecture

A 9-layered network is used for the generation of the colorful image. The following layers are used in the Convolutional Network we have designed.

Found 72526 images for training input.
Found 72516 images for training output .
Optimizer:”Adam”

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

Layer (type)	Shape
input_data	(shape=[None, 100, 100, 3], name='input')
conv2d (Conv2D)	(convnet, 32, 5, activation='relu')
conv2d (Conv2D)	(convnet, 64, 5, activation='relu')
conv2d (Conv2D)	(convnet, 128, 5, activation='relu')
conv2d (Conv2D)	(convnet, 128, 5, activation='relu')
conv2d (Conv2D)	(convnet, 64, 5, activation='relu')
conv2d (Conv2D)	(convnet, 32, 5, activation='relu')
dropout	(convnet, 0.8)
conv2d (Conv2D)	(convnet,3, 5, activation='linear')

Layers:
• Convolution: Convolutional layers convolve around the image to detect edges, lines, blobs of colors and other visual elements. Convolutional layers hyper parameters are the number of filters, filter size, stride, padding and activation functions for introducing non-linearity.

B. Face Classification and Search

We have built searching algorithm using Convolutional Neural Network which takes the generated image and search for same face in the available data set and return the information about that person. For this we have collected the photos of person and create a dataset with folder named with people name and then use that data set to train convolutional neural network for classifying the faces of the peoples. It is a classification network so we use the loss metric categorical_crossentropy which provides the same process for back propagation and weights updating and then it assign prper weights for this we have just used the keras model predefined network with our hyper parameters. We save the created model and can update it on new entry every time if the new person is arrived or entered in the database the model can be updated, form previous model using transfer learning it will increase the accuracy.
We only had one image of a person so trained it over more number of epochs.

Network Architecture

In this method we are using Dense Arch Network architecture with n_classes
Model: " DenseArchs(n_classes)"
Found 72526 images belonging to 72526 classes.

C. Face Recognition

For the security purpose we have added a face lock before loging in into the system. We have used K nearest neighbor algorithm for face recognition purpose in the start we just stored the images of the peoples which we want add as the admin or trusted person. The algorithm then converts those images into the numpy array and save them as .npy files, so when someone tries to login in the application it clicks the picture of the person converts it into the numpy array and run the K Nearest Neighbor algorithm and if the distance between saved numpy array and the new generated numpy array is less than the threshold we have set it will allow the user to pass through the security if the face recognition is not working well due to some problem you can try the old school login from database method.

In the login through database we use mysql databse and SQL is used for the identification and verification using the entries in database.

III. IMPLEMENTATION

A. Dataset Creation For Training

In order to use the dataset we needed we used the Celebrity Dataset from the Kaggle it contains almost 200000 images of the celebrity faces. We have done preprocessing on the dataset and created the dataset we need for the training of the Deep Convolutional Network. We have just used 72526 images from that.

The Steps for the data preprocessing are as follows.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

1. Cropping the face from the images:
   The images we have are all of the different sizes and have other body parts included to remove those clutters from the image and convert them into equal sizes we use harcascadefrontalface.xml to extract faces from image and resize them into 100*100 and save them in the folder called crop.
2. Converting Images into the sketches:
   To train a neural network we need the crop sketches for that we used grayscale manipulation for the we read the images from crop folder and inverted them into grayscale from color images then remove blur from image using GaussianBlur then use cv2,divide function for final conversion and save those images into folder call sketches.
3. Converting images into training ray or extracting features:
   To convert those into arrays we first read the array the images using cv2.imread and convert them into numpy array one by one then we add those arrays in our matrix for the training purpose and save it as a numpy array as .npy file for the further use in the model training.
   We do this process for folders we have created in previous step first is in the images in crop which are our output array. And then same process for the images in sketches folders which are our input given in the input layer.

And finally after preprocessing we save those numpy arrays.

B. DCNN

While training the Deep Convolutional Neural Network we pass the both numpy arrays we have saved after data preprocessing. One array is used as the training input and another array is the training output. Then we split it into training and validation. we set batch size is equal to 64 and the image shape is defined as 100x100.
We the start building our neural network.
For model purpose we used tflrearn conv2d with 9 layers.

Layer1: Input layer Conv2D with activation function RELU with input shape = (100, 100, 3).
Layer2: It is a Conv2D with activation function RELU with shape = (convnet, 32, 5, activation='relu').
Layer3:A Conv2D with activation function RELU with shape = (convnet, 64, 5, activation='relu').
Layer4: It is Conv2D with activation function RELU with shape = (convnet, 128, 5, activation='relu') .
Layer5: It is a convolutional layer Conv2D with activation function RELU with shape = (convnet, 128, 5, activation='relu').
Layer6: It is a Conv2D with activation function RELU with shape = (convnet, 64, 5, activation='relu').
Layer7:It is a Conv2D with activation function RELU with shape = (convnet, 32, 5, activation='relu').
Layer8:It is a Dropout prameters set as dropout(convnet, 0.8).
Layer9:It is a Conv2D with activation function Linear conv_2d(convnet,3, 5, activation='linear').

Then we passed the network to the Regression model with optimizer set as adam learning_rate=0.00054 and loss set as mean_square and name=targets. We built a DNN model fo tflearn by passing all the values and then train that model over the input we have provided earlier for number of epochs as 10 and saved the model as sketch for the use in the next module.

C. Face Classification and Search

While building the model for the face recognition classification and searching we directly treated the problem as a classification problem with multiple classes for which we created different folders for person images and put those images into the folders then by the default structure of the folder keras cnn model name the image labels. It reads the images from the folders with labels named as the folder names and convert them into the numpy array with size 160*160*3. Then we split those arrays in training and test data set.
We now create model network using DenseArchs(n_classes) where n_classes are equal to number of persons we have in data set. We them compile the model using keras.fit with loss set as cross entropy loss cause we are dealing with classifying problem and then we train it over number of epochs equal to 6 and save model for the future prediction.

D. Face Recognition

For security purpose we have added this feature. For this we use K nearest neighbor first we add the images of the person we want to add as admin or authorized person and label it as the persons name then our program take that image

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

extract the features of the image like lips positions chick bones nose position and save all the features as the labeled named by you.
So whenever someone tries to login we use open cv to capture the frame we extract the features from that frame also and compare them to the features we have we use the K nearest neighbor to find the distance between the features we have and feature we have extracted live then we use that distance to assign the person if he is in the authorized persons list or not the threshold for the matching is set very high to increase the precision of our classifier and decrease the recall.
The precision provide the power to not wrongly classify it helps the classifier to always classify if it is that person for real and not to classify if it is not that person.

IV. RESULTS AND DISCUSSION

A. *DCNN*

After completion of 10 epochs Accuracy and loss recorded for Training and Validation are as follows.

Training Step: 11339 | total loss: 1143.77698 | time: 375.252s

| Adam | epoch: 010 | loss: 1143.77698 - acc: 0.633 -- iter: 72512/72526

Training Step: 11340 | total loss: 1135.71045 | time: 444.442s

| Adam | epoch: 010 | loss: 1135.71045 - acc: 0.626 | val_loss: 1083.86912 - val_acc: 0.743 -- iter: 72526/72526

B. *Face Classification and Search*

After completion of 6 epochs Accuracy and loss recorded for Training and Validation are as follows.

Training: Accuracy=0.7678 Loss=0.6529 Validation: Accuracy=0.5678 Loss=0.4529

C. *Working of whole Application*

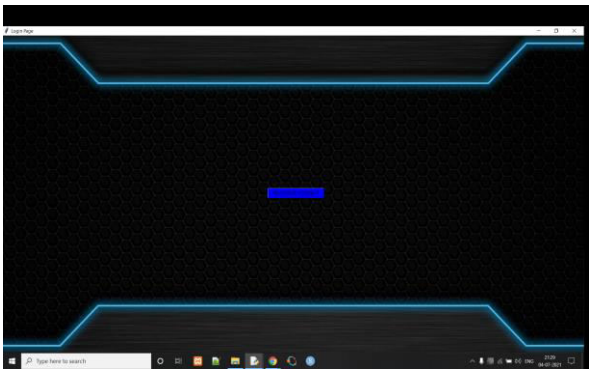

Fig1:-Face Recognition Page

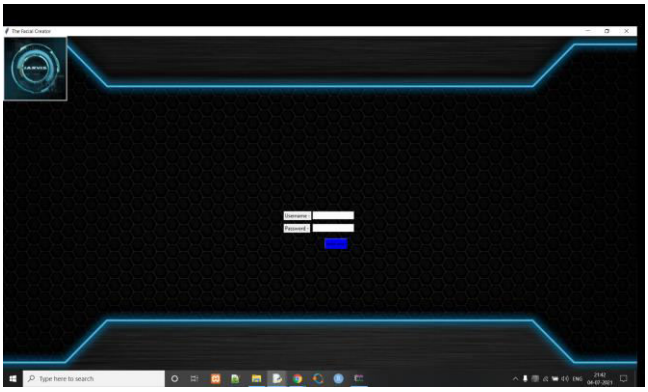

Fig2:-Home Page

Face Recognition successfully done as described above

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

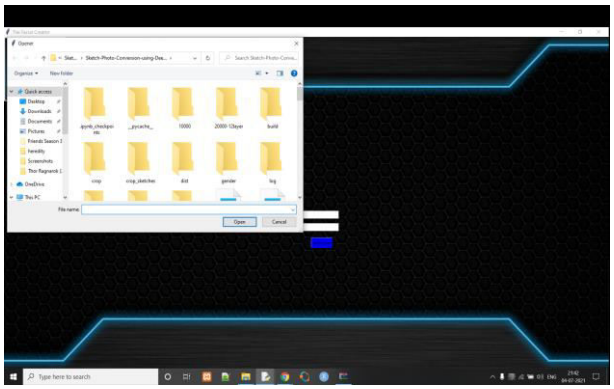

Fig3:-Image Uploading

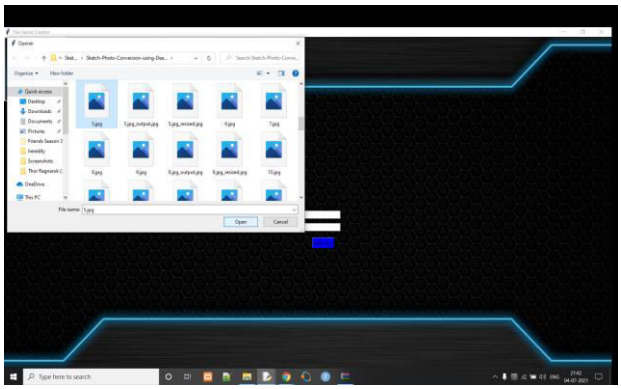

Fig4:-Image Uploading

Using the file dialog. Ask open filename function in tinkter we can choose the file from the local disk instead of uploading or giving through an argument cause the user might not always know the python and know how to run so we use this UI to make file choosing easy for the user

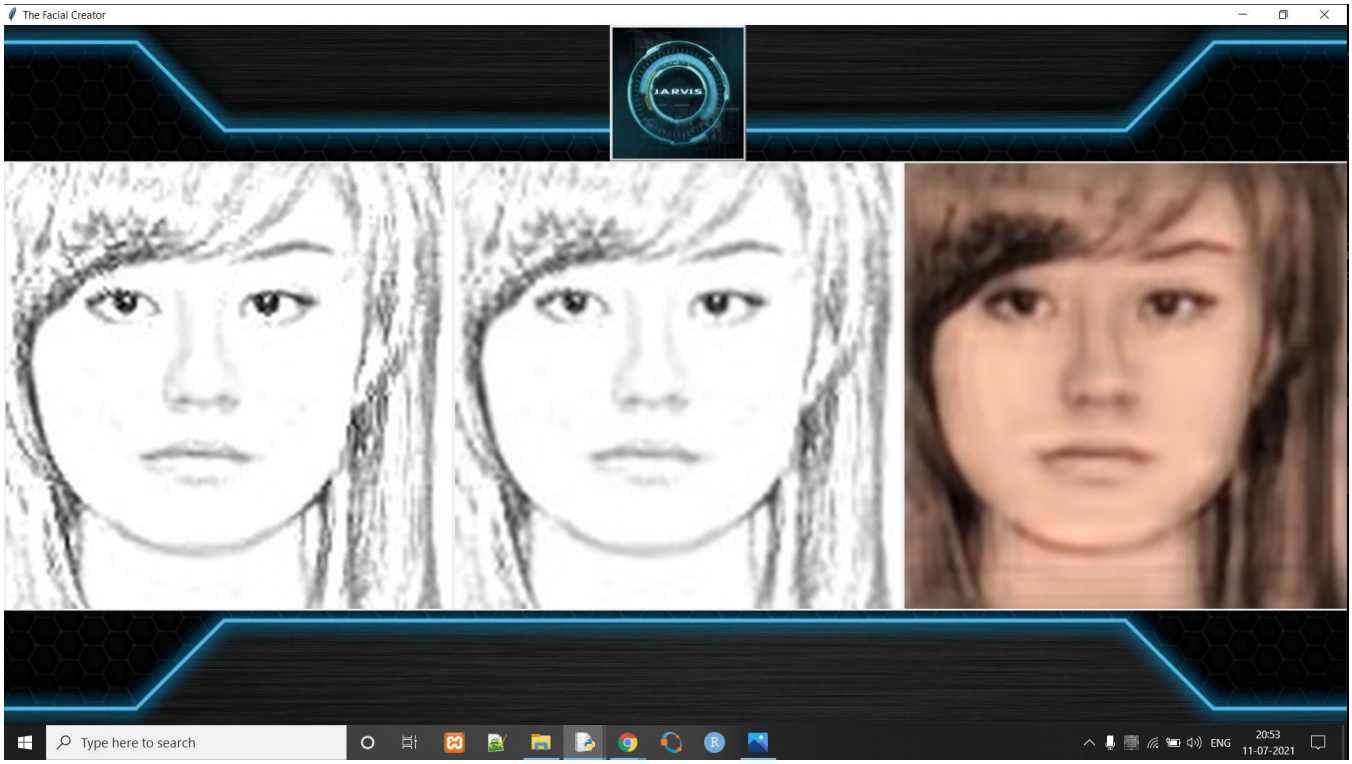

Fig5:-Result

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

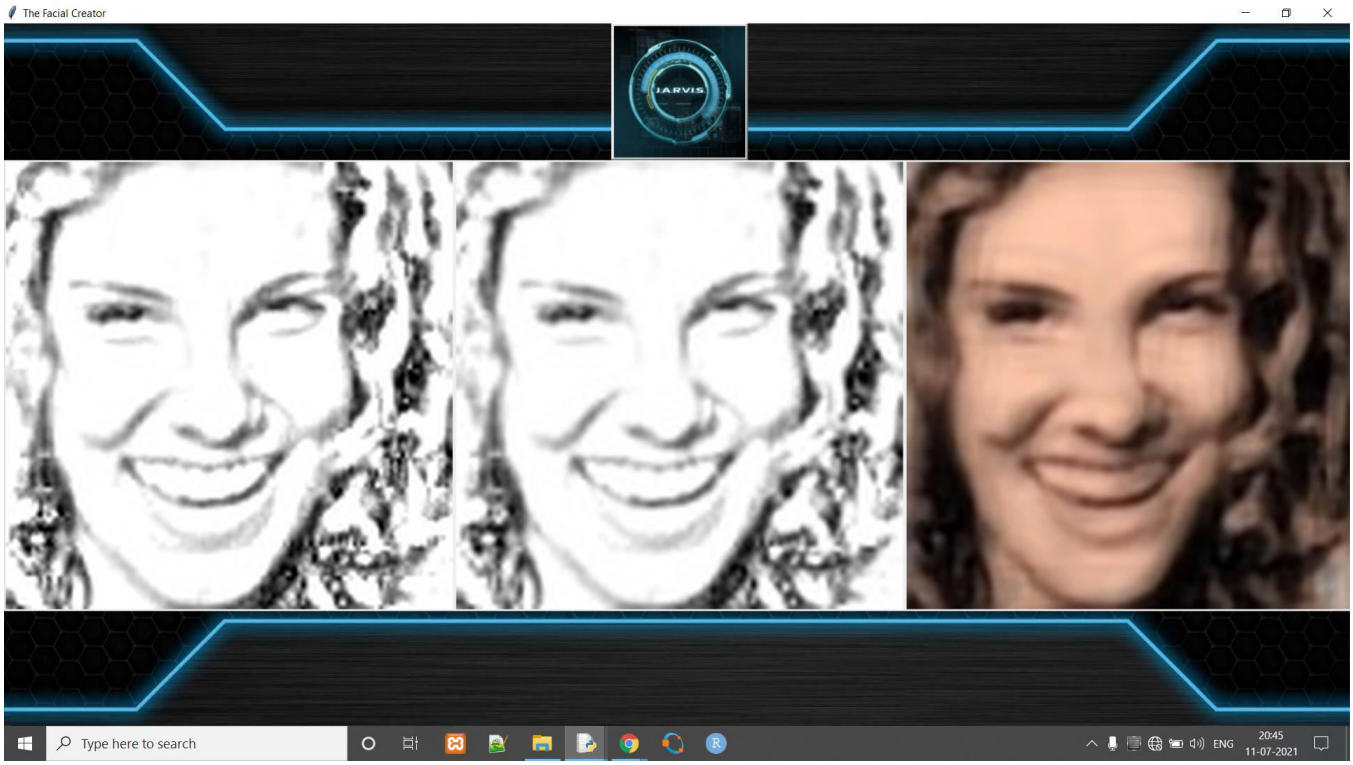

Fig6:-Result

The final Images we have got are clear and show all the features of human face.

V. CONCLUSION AND FUTURE SCOPE

A. *Conclusion*

After implementation of both DCNN for image generation and training a neural network for face classification and KNN face lock and face recognition we can conclude that we have successfully built the Application for the grayscale imagei into colorful images for feature extraction and finding the face with same features in the dataset.

B. *Future Scope*

This application can be used in the Forensic Lab for the face recognition from the sketches.The Accuracy of the model can be increased by training it on GPU due to low computational power we were not able to achieve more accuracy and after changing some hypermeters it can also be achivied.

REFERENCES

[1] Rob van Lier Yamur Gltrk, Umut Gl and Marcel A. J. van Gerven.Convolutional sketch inversion. In Computer Vision ECCV 2016 Workshops, 2016.
[2] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. In IEEE Transactions On Pattern Analysis And Machine Intelligence, 2014.
[3] Dacheng Tao Xinbo Gao, Nannan Wang and Wei Liu. Face sketch-photo synthesis and retrieval using sparse representation. In IEEE Transactions on Circuits and Systems for Video Technology, 2012.
[4] Xian Wu Shengyong Ding Liliang Zhang, Liang Lin and Lei Zhang. End-to-end photo-sketch generation via fully convolutional representation learning. In ICMR15, 2015.
[5] Nannan Wang Jiewei Zhang and Xinbo Gao. Face sketch-photo synthesis based on support vector regression. In 2011 18th IEEE International Conference on Image Processing, 2011.
[6] K. He C. Dong, C. C. Loy and X. Tang. Learning a deep convolutional network for image superresolution. In Computer VisionECCV 2014, 2014.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

| DOI:10.15680/IJIRSET.2021.1007247|

[7] M. Savvides Y.-h. Li and V. Bhagavatula. Illumination tolerant face recognition using a novel face from sketch synthesis approach and advanced correlation filters. In International Conference on Acoustics, Speech, and Signal Processing, Institute of Electrical and Electronics Engineers (IEEE), 2006, 2006.
[8] LeCunn et al. Convolutional neural networks (lenet) deeplearning 0.1 documentation. In DeepLearning 0.1. LISA Lab, 2013.
[9] LeCunn et al. Lenet - 5 convolutional neural networks. In Proceedings of the IEEE, november 1998, 1998.
[10] Paul Viola and Michael J. Jones. Robust real time face detection. In International Journal of Computer Vision, 2004.
[11]Rejin Joy* Electrical and Computer Engineering University of Florida Email:rejinjoy18@ufl.edu
Rishabh Singh* Electrical and Computer Engineering University of Florida Email:rish283@ufl.edu
Sketch Inversion and Colorization of Gray-scale Photos and Videos using Convolutional Learning April 2017
[12] B. Karl, Z. Li, and A. K. Jain Matching forensic Sketches to mugshot photos, IEEE Trans Pattern Anal. Match. Vol. 33, no. 3, pp. 639-649, 2011.
[13] B. Klara and A. K. Jain Heterogeneous Face Recognition Using Kernel Prototype Similarities, IEEE Trans. pattern anal. Match. Intel. vol. 35, no. 6, pp. 1410-1422, June 2013
[14] A. M. Martinez and R. Benvento, The AR Face database, CVC Tech. Rep, June 2013
[15] O. M. Parkhill, A. Vivaldi, and A. Zisserman Deep Face recognition, in British Machine Vision Conference, 2015

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details