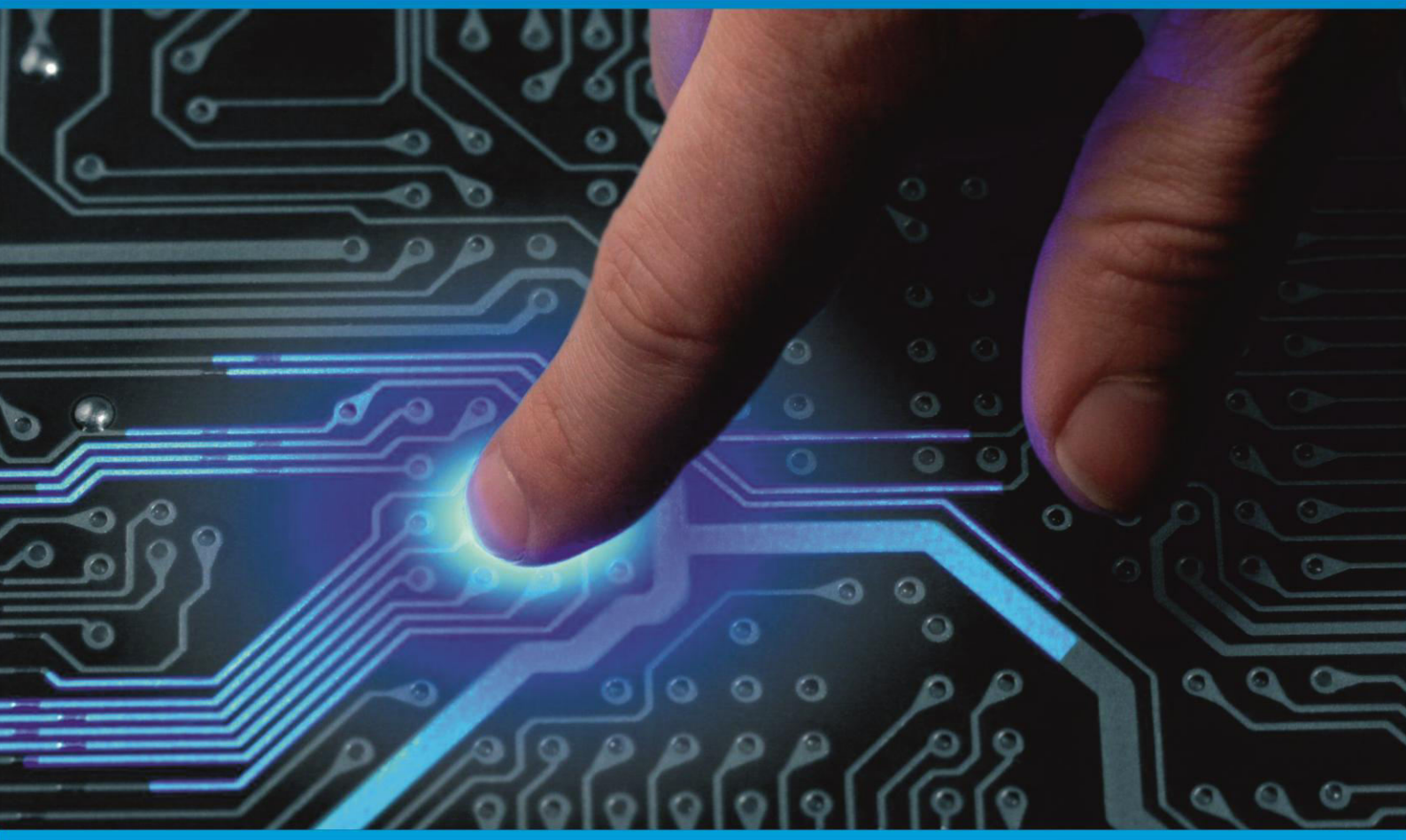
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 4, April 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904199|

Finding Missing Person Using Machine Learning

Neha Gholape [*1], Ashish Gour[*2], Shivam Mourya[*3], Dr. Seema Ladhe[4]

Department of Information Technology, PVPPCOE, Sion Mumbai, Maharashtra, India

ABSTRACT: Face recognition has been one of the most interesting and important research fields in the past two decades. The reasons come from the need of automatic recognition and surveillance systems, the interest in human visual systems on face recognition, and the design of human-computer interface, etc. The process of identification is done by a face recognition system based on facial features and their actions. In this paper we are working on the concept of KNN algorithm with open CV, and a Haar cascade classifier and various machine learning python libraries. This algorithm helps to extract the key features of a face like eyes, nose, mouth and that image gets correlated with the image available in the data set, and afterwards classification of it takes place.

KEYWORDS: Face Detection, KNN, Face Recognition, OpenCV, Numpy, Pandas, Numpy, PIL-imaging library, Mysql. connector

I. INTRODUCTION

In this report, we focus on image, video and live web cam based face recognition. Given a picture/video taken from a digital camera, we'd like to know if there is any person inside, where his/her face is located, and who he/she is. Towards this goal, we generally separate the face recognition procedure into three steps: Face Detection, Feature Extraction, and Face Recognition. This study developed an algorithm for real-time face detection and recognition with a complex background that is efficient and resilient. Ada Boost, cascade classifier, Local Binary Patent (LBP), Haar-like features, face image pre-processing and Principal Component Analysis (PCA) are a series of signal processing methods. The PCA algorithm is used to recognize faces efficiently. This rhythm reaches 99.2% for correct facial recognition and a true positive level of 98.8% for face detection.

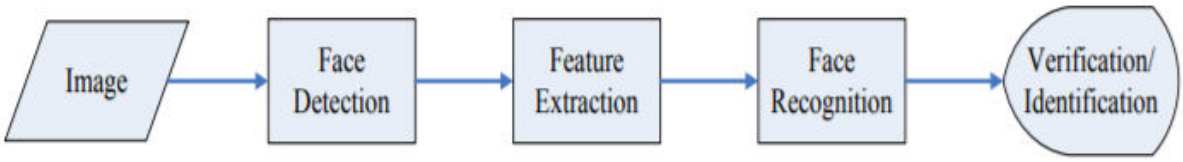


Figure 1: Configuration of a general face recognition structure

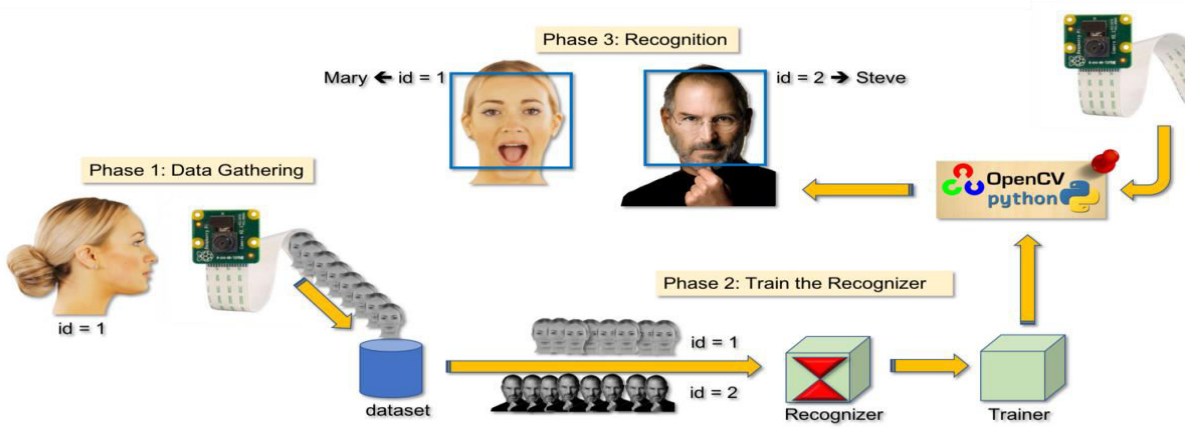
The main function of this step is to determine (1) whether human faces appear in a given image, and (2) where these faces are located at. The expected outputs of this step are patches containing each face in the input image. In order to make further face recognition systems more robust and easy to design, face alignment is performed to justify the scales and orientations of these patches. Besides serving as the pre-processing for face recognition, face detection could be used for region-of-interest detection, retargeting, video and image classification, etc. Feature Extraction: After the face detection step, human-face patches are extracted from images. Directly using these patches for face recognition have some disadvantages, first, each patch usually contains over 1000 pixels, which are too large to build a robust recognition system1. Second, face patches may be taken from different camera alignments, with different face expressions, illuminations, and may suffer from occlusion and clutter. To overcome these drawbacks, feature extractions are performed to do information packing, dimension reduction, salience extraction, and noise cleaning. After this step, a face patch is usually transformed into a vector with fixed dimension or a set of fiducially points and their corresponding locations. In order to achieve automatic recognition, a face database is required to build. For each person, several images are taken and their features are extracted and stored in the database. Then when an input face

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904199|

image comes in, we perform face detection and feature extraction, and compare its feature to each face class stored in the database.



II. ALGORITHM AND LIBRARIES USED

A. KNN Algorithm
B. PIL-Imaging library
C. OpenCV
D. Tkinter
E. Numpy
F. Pandas
G. Mysql.connector (database connectivity)

A. KNN Algorithm

KNN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones.

```
def train(submitted_by: str):
    """
    Trains a KNN Model on the submitted cases.

    Args:
        submitted_by: str

    Returns:
        dict - {
            "status": bool - whether the functional call was successful or not
            "message": str - message returned on each case
        }
    """
    model_name = 'classifier.pkl'
    if os.path.isfile(model_name):
        os.remove(model_name)

    try:
        labels, key_pts = get_train_data(submitted_by)
        if len(labels) == 0:
            return {"status": False, "message": "No cases submmited by this user"}
        le = LabelEncoder()
        encoded_labels = le.fit_transform(labels)
        classifier = KNeighborsClassifier(n_neighbors=len(labels),
                                          algorithm='ball_tree',
                                          weights='distance')
        classifier.fit(key_pts, encoded_labels)

        with open(model_name, 'wb') as file:
            pickle.dump((le, classifier), file)
        return {"status": True, "message": "Model Refreshed"}
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904199|

B. PIL-Imaging library

Pil Library is Python Imaging Library which is an open source library for python. The Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

```
from tkinter import*
from tkinter import ttk
from PIL import Image, ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
import os
import numpy as np


class Train:
    def __init__(self, root):
        self.root = root
        self.root.geometry("1530x790+0+0")
        self.root.title("FINDING MISSING PERSON USING AI")

        title_lbl = Label(self.root, text = "TRAIN DATA SET", font = ("
        title_lbl.place(x=0, y=0, width = 1530, height = 45)


        img_top = Image.open(r"C:\Users\gaur2\OneDrive\Documents\FINAL
        img_top = img_top.resize((1530, 425), Image.ANTIALIAS)
        self.photoimg_top = ImageTk.PhotoImage(img_top)
```

C. OpenCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

```
import cv2
import time
import datetime
import imutils

curr_path = os.getcwd()

print("Loading face detection model")
proto_path = os.path.join(curr_path, 'model', 'deploy.prototxt')
model_path = os.path.join(curr_path, 'model', 'res10_300x300_ssd_iter_140000.caffemodel')
face_detector = cv2.dnn.readNetFromCaffe(prototxt=proto_path, caffeModel=model_path)

print("Loading face recognition model")
recognition_model = os.path.join(curr_path, 'model', 'openface_nn4.small2.v1.t7')
face_recognizer = cv2.dnn.readNetFromTorch(model=recognition_model)

recognizer = pickle.loads(open('recognizer.pickle', "rb").read())
le = pickle.loads(open('le.pickle', "rb").read())

print("Starting test video file")
vs = cv2.VideoCapture('testvideo.mp4')
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904199|

D. Tkinter

Tkinter is the most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest.

E. Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Array operations can be performed only on array so firstly we converted our data in multidimensional array format for further operations.

```
from tkinter import*
from tkinter import ttk
from PIL import Image, ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
import os
import numpy as np



class Train:
    def __init__(self, root):
        self.root = root
        self.root.geometry("1530x790+0+0")
        self.root.title("FINDING MISSING PERSON USING AI")

        title_lbl = Label(self.root, text = "TRAIN DATA SET", font
        title_lbl.place(x=0, y=0, width = 1530, height = 45)


        img_top = Image.open(r"C:\Users\gaur2\OneDrive\Documents\F
        img_top = img_top.resize((1530, 425), Image.ANTIALIAS)
        self.photoimg_top = ImageTk.PhotoImage(img_top)
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

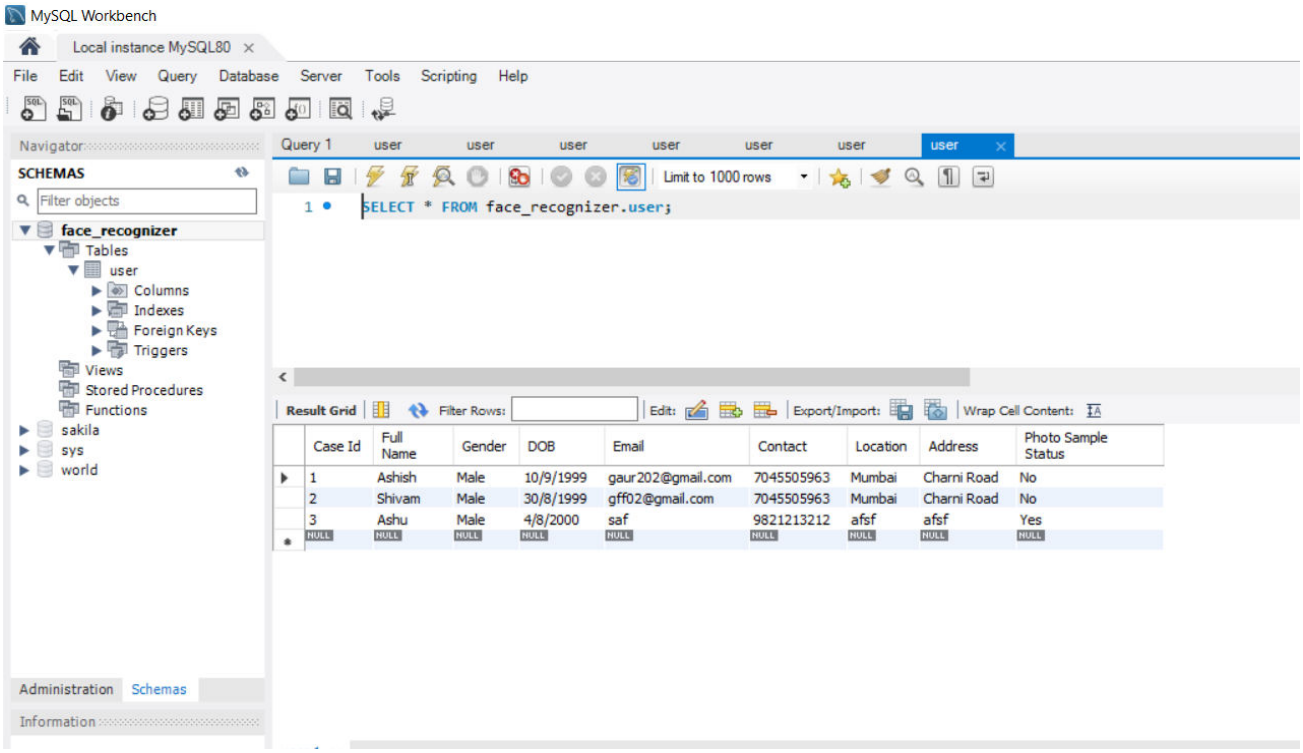
| DOI: 10.15680/IJIRCCE.2021.0904199|

F. Pandas

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on the top of the NumPy library. Pandas is fast and it has high-performance & productivity for users.

G. Mysql.connector

It is a driver for connecting to a MYSQL database server through the open database connectivity Application Program Interface (API).For this system we wrote a python program to access the MYSQL database using API.



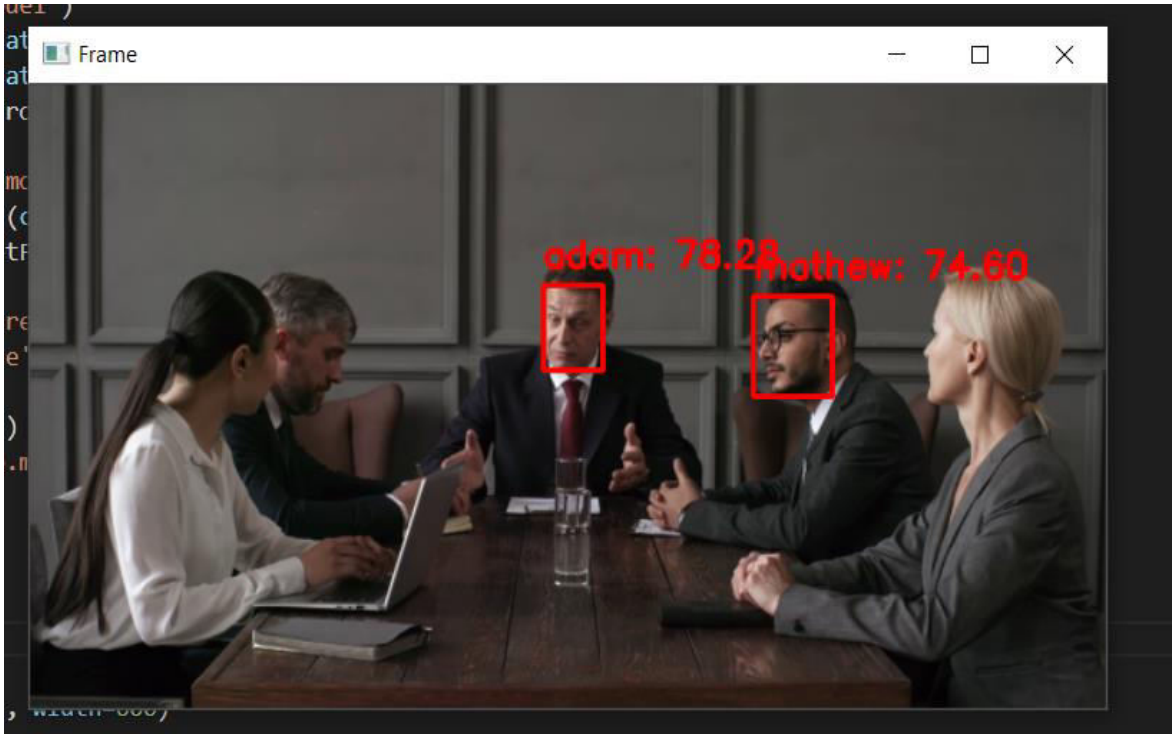
III. CONCLUSION

In this paper we have presented an experiment for face identification using the KNN method. KNN is one of the simplest algorithms that can be used for classification. The sources of data come from manual shooting which is divided into 30 classes. The face identification using the KNN method consists of two stages, such as the training phase and the testing phase. Based on the results by changing the parameter k value obtained results are different for each parameter. The results give accuracy 81% for k=1, give accuracy 53% for k=2, give accuracy 47% for k=3. From the results, it shows that the value of k greatly affects the level of accuracy of the system. The parameter k value and accuracy are inversely proportional, the greater k value gives the smaller accuracy of the identification system. From this research can be concluded that the higher k value is the smaller accuracy that we get for face identification using KNN.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904199|



REFERENCES

1. Ranjan , R., Bansal, A., Zheng, J., Xu, H., Gleason, J., Lu, B., et al. (2015). A Fast and Accurate System for Face Detection,. JOURNAL OF LATEX CLASS, 14(8).
2. Mantoro, T., Ayu, M. A., & Suhendi. (2018). Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods. 2018 6th International Conference on Multimedia Computing and Systems (ICMCS).
3. Rachmawanto, E. H., Anarqi, R. G., Setiadi, D. I., & Sari, A. (2018).Handwriting Recognition Using Eccentricity and Metric Feature Extraction Based on K-Nearest Neighbors. 2018 International Seminar on Application for Technology of Information and Communication(iSemantic).
4. Fransiska, R. (2016). Voice Recognition Using k Nearest Neighbor and Double Distance Method. 2016 International Conference on Industrial Engineering, Management Science and Application (ICIMSA).
5. Asana, I., Widyantara, I., Wirastuti, N., & Adnyana, I. (2017). Metode Contrast Stretching untuk Perbaikan Kualitas Citra pada Proses Segmentasi Video. OJS Unud, 16(2).
6. Natesan, M., & Bhuvaneshwaran, K. (2014). Image Enhancement Using Multilevel Contrast Stretching and Noise Smoothening Technique for CT Images. International Journal of Scientific and Engineering Research, 5(5), 713-718.
7. Mahmud , F., Khatun, T., Zuhori, S., Afroge, S., Aktar, M., & Pal, B.(2015). Face recognition using Principal Component Analysis and Linear Discriminant Analysis. International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)
8. Paul , L., & Sumam, A. (2012). Face Recognition Using Principal Component Analysis Method. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1(9).

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.488

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details