“Motion Detection & Surveillance System using Python & Open-CV”

Minor-Project Report Submitted

in

Partial Fulfilment of requirement for the award for the Degree of

BACHELOR OF TECHNOLOGY

in

DATA SCIENCE

by

Anurag Selote
Pranita Ambulkar
Shubham Mowade
Shreyash Bagdear
Sakshi Dhage
Tushar Nandurkar

Guide

Dr. Swapnili Karmore

(*HOD, Associate Prof, Data Science Department*)

Department of Data Science

G H RAISONI INSTITUTE OF ENGINEERING AND TECHNOLOGY NAGPUR

(An Autonomous Institute Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur) NBA & NAAC Accredited with A+ Grade

2022-2023

G H RAISONI INSTITUTE OF ENGINEERING AND TECHNOLOGY NAGPUR

Department of Data Science

CERTIFICATE

This report of the project titled "**Motion Detection & Surveillance System using Python & Open-CV**" submitted by **Anurag Selote, Pranita Ambulkar, Sakshi Dhage, Shreyash Bagde, Shubham Mowade and Tushar Nandurkar** in the partial fulfillment of the degree of **Bachelors of technology** in **Data Science** during the academic year 2022-23, has been carried out under my supervision at the department of Data Science of G H Raisoni Institute of Engineering and Technology, Nagpur. The work is comprehensive, complete and fit for evaluation.

Dr. Swapnili Karmore

(HOD, Associate Prof, Data Science Department)

G H RAISONI INSTITUTE OF ENGINEERING AND TECHNOLOGY NAGPUR

Department of Data Science DECLARATION

(*for Individual Student*)

I, **Shubham Mowade** hereby declare that the project titled "**Motion Detection & Surveillance System using Python & Open-CV**" is my own work carried out under the guidance of **Dr. Swapnili Karmore** at **G H Raisoni Institute of Engineering and technology, Nagpur**. This Work in the same form is not submitted by me or anyone else for the award of the degree and the following student are associated with me:

1. Pranita Ambulkar
2. Sakshi Dhage
3. Anurag Selote
4. Shreyash Bagde
5. Tushar Nandurkar

Date:

Name & Sign of one Declaring Student

G H RAISONI INSTITUTE OF ENGINEERING AND TECHNOLOGY NAGPUR

Department of Data Science DECLARATION

We, **Anurag Selote, Pranita Ambulkar, Sakshi Dhage, Shreyash Bagde, Shubham Mowade and Tushar Nandurkar** are hereby declare that the project titled "**Motion Detection & Surveillance System using Python & Open-CV"** is our own work carried out under the guidance of **Dr. Swapnili Karmore** at **G H Raisoni Institute of Engineering and Technology, Nagpur**. This work in the same form is not submitted by us or anyone else for award of degree.

Date: Name & Sign of Student

G H RAISONI INSTITUTE OF ENGINEERING AND TECHNOLOGY NAGPUR

Department of Data Science ACKNOLEDGEMENT

We feel immense pleasure in mentioning our indebtedness to those who helped in the carrying of our project titled "**Motion Detection & Surveillance System using Python & Open-CV**" from the initial stage of electing the subject of research till the writing of conclusion, a student has turn to guide. We are, therefore to our guide, "**Dr. Swapnili Karmore**" Department of Data Science, who helped us along to complete this project. We are thankful to **Dr. Swapnili Karmore**, Head of Department of Data Science, GHRIET Nagpur, for their valuable contribution in fulfilling our requirement related to the project.

We would like to extend our gratitude to "**Dr. Vivek Kapur**", Director, GHRIET Nagpur for being a constant source for the inspiration.

Finally, we would like to thank all those who have contributed, directly or indirectly to make this project successful.

Anurag Selote
Pranita Ambulkar
5Sakshi Dhage
Shreyash Bagde
Tushar Nandurkar
Shubham Mowade

INDEX

CHAPTER I

INTRODUCTION

INTRODUCTION

In recent years, motion detection has attracted a great interest from computer vision researchers due to its promising applications in many areas, such as video surveillance, traffic monitoring or sign language recognition. However, it is still in its early developmental stage and needs to improve its robustness when applied in a complex environment: Several techniques for moving object detection have been proposed among them the three representative approaches are temporal differencing, background subtraction and optical flow.

In recent years, motion detection has attracted a great interest from computer vision researchers due to its promising applications in many areas, such as video surveillance, traffic monitoring or sign language recognition. However, it is still in its early developmental stage and needs to improve its robustness when applied in a complex environment: Several techniques for moving object detection have been proposed among them the three representative approaches are temporal differencing, background subtraction and optical flow.

The following Technologies will get used in this project:

1. Open-Source CV

Open-Source Computer Vision is a machine learning software library which provides a real-time optimized Computer vision library and tools. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. It can be used in python for image processing, motion detection, face recognition, and other function.

2. Python

Python is a high-level general-purpose programming language. It is object oriented which helps programmers to write a clear, logical code for a problem. Open CV supports python than can process images and videos to detect movements, identify objects, recognize faces and much more.

CHAPTER II

LITRATURE SURVEY & REVIEW

LITRATURE SURVEY & REVIEW

Literature review is needed before any project is begun. The review will help to understanding the scope of the project and the need to build the project. The review comes from the reading on the websites and from the books. The information from there view will be used to start the project with an excellent idea. The paper discusses about the detection of motion in a monitored area by using the frame difference methods. The technique discussed in the paper is based upon the combination of two different frame differencing methods which are the Background subtraction method and Consecutive frame subtraction method. The technique is based upon the background images acquired from the previously captured frames. The current image is compared pixel by pixel to an image which is actually the mean of the background images taken. The motion is detected by analyzing the difference in the pixel value of the current image with respect to the background images and the difference in pixel values is then checked against a threshold value. If the difference is greater than the threshold value set by the user, can be inferred that the motion is detected in the current camera frame else if the difference is less than the threshold, the current image is not detected with the presence of any motion and the system analyzes the next image for motion. Finally, the motion detection is indicated by an alarm or by using a graphical method. The technique is implemented using MATLAB and the results obtained depicts that the methods lead to avoidance of false positive alarms, but the amount of time required to train the system to avoid false alarms is very high which is a limitation to this technique. The improvement that can be done is to Use machine learning algorithms which can help learn the system at a faster pace and the limitation to this system is overcome. The paper proposes computer vision techniques that can be used to design a visual surveillance home security system to protect against intrusions and theft.

The paper discusses about using the combination of motion detection and face recognition techniques to build the system. The motion detection module uses frame difference methods to detect the presence of motion in the frame. The current frame us compared with the previous frame's pixel wise. The difference in pixel values is noted and used to detect motion. If the difference is greater than 15 percent from a set threshold value, then it indicates that the motion has been detected and it triggers the face recognition module. The face recognition module uses the Eigen face recognition algorithm to recognize faces. The face detection module differentiates the intruders from the authorized users. A database of authorized users is provided by the owners of the property. If a person is detected and his face is not recognized in the authorized person's database, it indicates that the person can be an intruder or thief and alarm is triggered and an email is generated to notify the owners about intrusion. The system is tested during day and night

at various times, and it was noted that the system is less responsive during nighttime to capture motion and the recognition rate is not high enough. Still the system is highly robust, reliable and efficient. The system can be improved by making the face recognition module less sensitive to variations in light intensity and increasing the recognition ratio. The paper [3] aims at using facial expression and biometrics technologies to build an intelligent surveillance system that identifies and recognizes the intentions of a person to commit any act of theft, robbery or any other criminal activity.

The paper discusses about the detection of motion in a monitored area by using the frame difference methods. The technique discussed in the paper is based upon the combination of two different frame differencing methods which are the Background subtraction method and Consecutive frame subtraction method given in fig [1]. The technique is based upon the background images acquired from the previously captured frames. The current image is compared pixel by pixel to an image which is actually the mean of the background images taken. The motion is detected by analyzing the difference in the pixel value of the current image with respect to the background images and the difference in pixel values is then checked against a threshold value. If the difference is greater than the threshold value set by the user, can be inferred that the motion is detected in the current camera frame else if the difference is less than the threshold, the current image is not detected with the presence of any motion and the system analyzes the next image for motion. Finally, the motion detection is indicated by an alarm or by using a graphical method. The technique is implemented using Matlab and the results obtained depicts that the methods lead to avoidance of false positive alarms, but the amount of time required to train the system to avoid false alarms is very high which is a limitation to this technique. The improvement that can be done is to use machine learning algorithms which can help learn the system at a faster pace and the limitation to this system is overcome.

The paper proposes computer vision techniques that can be used to design a visual surveillance home security system to protect against intrusions and theft. The paper discusses about using the combination of motion detection and face recognition techniques to build the system. The motion detection module uses frame difference methods to detect the presence of motion in the frame. The current frame us compared with the previous frames' pixel wise. The difference in pixel values is noted and used to detect motion. If the difference is greater than 15 percent from a set threshold value, then it indicates that the motion has been detected and it triggers the face recognition module. The face recognition module uses the Eigen face recognition algorithm to recognize faces. The face detection module differentiates the intruders from the authorized users. A database of authorized users is provided by the owners of the property. If a person is detected and his face is not recognized in the authorized person's database, it indicates

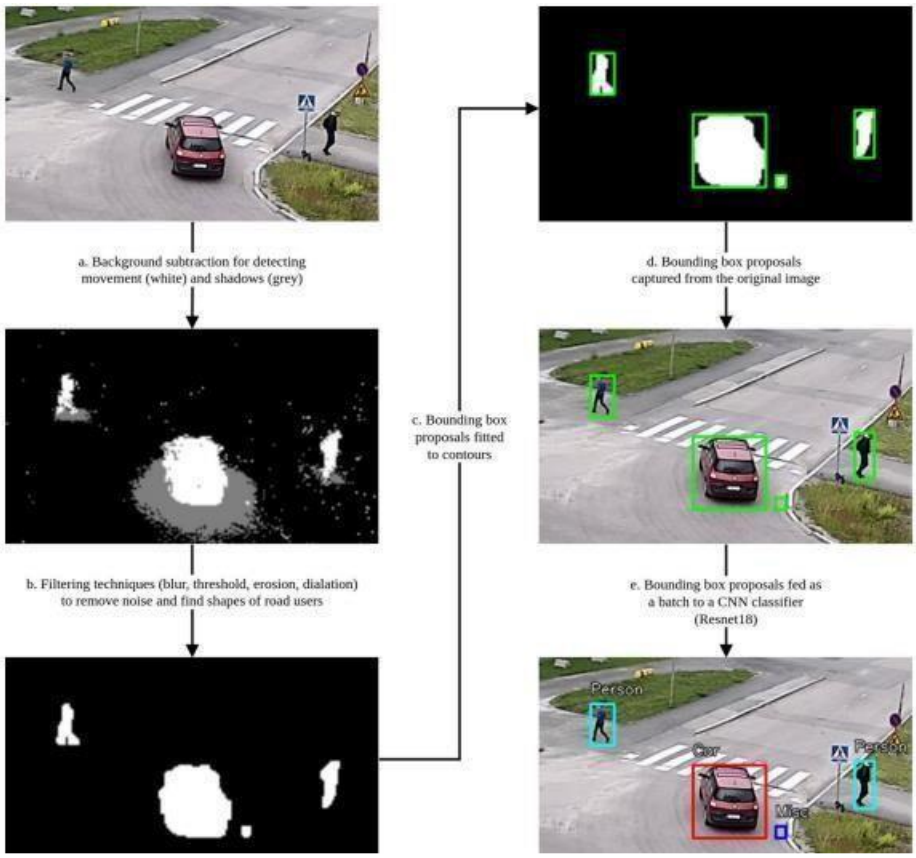
that the person can be an intruder or thief and alarm is triggered and an email is generated to notify the owners about intrusion. The system is tested during day and night at various times, and it was noted that the system is less responsive during nighttime to capture motion and the recognition rate is not high enough. Still the system is highly robust, reliable and efficient. The system can be improved by making the face recognition module less sensitive to variations in light intensity and increasing the recognition ratio. The paper [3] aims at using facial expression and biometrics technologies to build an intelligent surveillance system that identifies and recognizes the intentions of a person to commit any act of theft, robbery or any other criminal activity.

CHAPTER III

METHODOLOGY

METHODOLOGY

The road user detection algorithm proposed here consists of motion detection and classification, which are applied in series. The approach has been specifically tuned for ITS applications, with low computational cost and scalability in mind. In addition, multiple well-established image processing methods such as blur, erosion, and dilation are utilized to filter the contents of the images. Workflow of the algorithm is presented in Figure displaying the steps taken to reach detections for an image.



Fig[1]

When an 8-bit three channel RGB image is fed into the algorithm, the input image is first resized and processed with motion detection. the image is transformed into a gray-scale image where each moving pixel is presented in white, and shadows are presented in gray (a). Since the acquired gray scale image is typically noisy, the image is blurred, thresholder into a binary image, and finally eroded and dilated (b). Remaining areas after the filtering are fitted with bounding boxes (c). Using these bounding box coordinates, bounding box proposals are captured from the original RGB image (d). Te acquired bounding box proposals are fed into a classification CNN, which gives the class of each bounding box proposal as an output (e). Consequently, the output of the image coordinates and types of the visible moving objects. Further details of the operation are provided in the following sections.

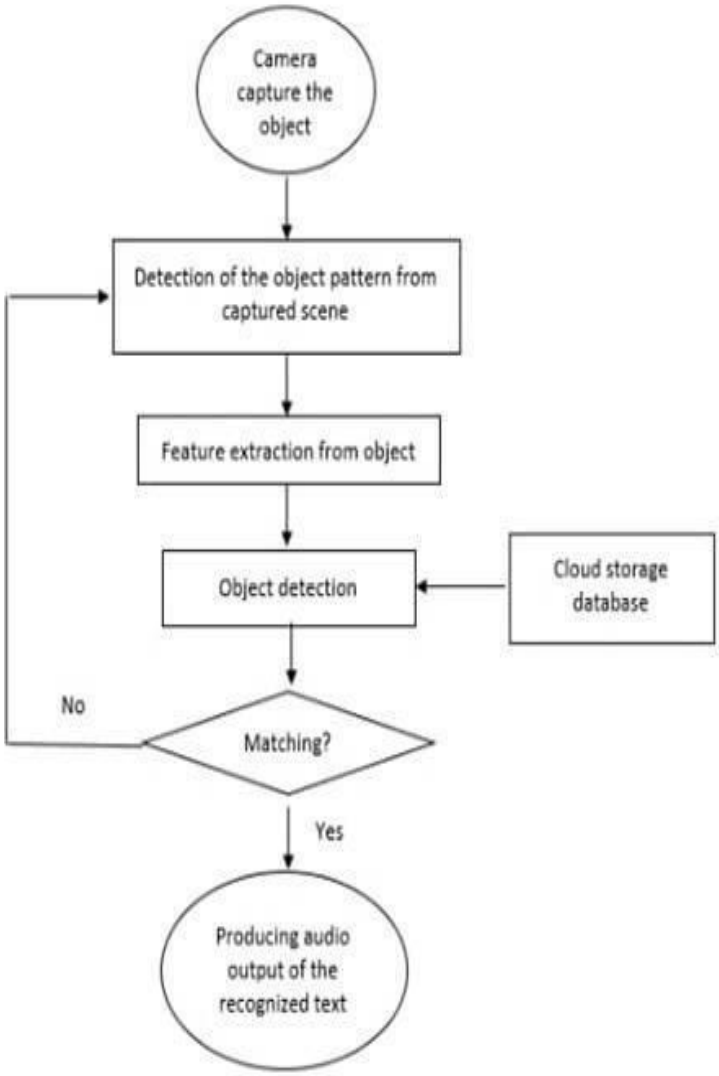


Fig. Block Diagram Representation

Figure explains the overview of the project, the step-by-step process to detect motion. First, we need a camera to capture the video then image acquisition will be done in which background subtraction of a scene will be executed. Motion detection algorithm will be written in python and a recognition engine will be used to detect the movement.

CHAPTER IV

DESIGN AND IMPLEMENTATION

DESIGN & IMPLEMENTATION

4.1. IMPLEMENTATION

1. Importing the package

The deep Learning model, we imported the modules that are used in the deep Learning .

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os

# Importing Deep Learning Libraries

from keras.preprocessing.image import load_img, img_to_array
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense,Input,Dropout,GlobalAveragePooling2D,Flatten,Conv2D,BatchNormalization,Activation,MaxPooling2D
from keras.models import Model,Sequential
from keras.optimizers import Adam,SGD,RMSprop
```

Fig : Importing of the modules for Deep Learning model

2. Importing the Datasets

Then. We upload the datasets, that is the Data of facial emotion recognition.

```
picture_size = 48
folder_path = "../input/face-expression-recognition-dataset/images/"


expression = 'happy'

plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img(folder_path+"train/"+expression+"/"+
                   os.listdir(folder_path + "train/" + expression)[i], target_size=(picture_size, picture_size))
    plt.imshow(img)
plt.show()
```

Fig 9: Importing the datasets

3. Pre-processing

Then after we have to train the data set and validate a data .

```
batch_size  = 128

datagen_train  = ImageDataGenerator()
datagen_val = ImageDataGenerator()

train_set = datagen_train.flow_from_directory(folder_path+"train",
                                              target_size = (picture_size,picture_size),
                                              color_mode = "grayscale",
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              shuffle=True)


test_set = datagen_val.flow_from_directory(folder_path+"validation",
                                              target_size = (picture_size,picture_size),
                                              color_mode = "grayscale",
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              shuffle=False)
```

Fig 10: Pre-processing on the facial data

4. Building a Model

To Implement the trained data on real data for that we made a CNN network for good result that is nearly 4 layer .

```
from keras.optimizers import Adam,SGD,RMSprop


no_of_classes = 7

model = Sequential()

#1st CNN layer
model.add(Conv2D(64,(3,3),padding = 'same',input_shape = (48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#2nd CNN layer
model.add(Conv2D(128,(5,5),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))

#3rd CNN layer
model.add(Conv2D(512,(3,3),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))
```

```python
#4th CNN layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())

#Fully connected 1st layer
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))


# Fully connected layer 2nd layer
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(no_of_classes, activation='softmax'))



opt = Adam(lr = 0.0001)
model.compile(optimizer=opt,loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Python

Fig 11: Using CNN Network for classification

5. Model Accuracy Testing

Here we validate the model.

```python
from keras.optimizers import RMSprop,SGD,Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau

checkpoint = ModelCheckpoint("./model.h5", monitor='val_acc', verbose=1, save_best_only=True, mode='max')

early_stopping = EarlyStopping(monitor='val_loss',
                               min_delta=0,
                               patience=3,
                               verbose=1,
                               restore_best_weights=True
                               )

reduce_learningrate = ReduceLROnPlateau(monitor='val_loss',
                                        factor=0.2,
                                        patience=3,
                                        verbose=1,
                                        min_delta=0.0001)

callbacks_list = [early_stopping,checkpoint,reduce_learningrate]

epochs = 48

model.compile(loss='categorical_crossentropy',
              optimizer = Adam(lr=0.001),
              metrics=['accuracy'])
```

```
history = model.fit_generator(generator=train_set,
                              steps_per_epoch=train_set.n//train_set.batch_size,
                              epochs=epochs,
                              validation_data = test_set,
                              validation_steps = test_set.n//test_set.batch_size,
                              callbacks=callbacks_list
                              )
```

```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/48
225/225 [==============================] - 180s 799ms/step - loss: 1.7711 - accuracy: 0.3183 - val_loss: 1.7775 - val_accuracy: 0.3482
Epoch 2/48
225/225 [==============================] - 22s 99ms/step - loss: 1.4261 - accuracy: 0.4517 - val_loss: 1.3999 - val_accuracy: 0.4814
Epoch 3/48
225/225 [==============================] - 22s 97ms/step - loss: 1.2760 - accuracy: 0.5113 - val_loss: 1.2810 - val_accuracy: 0.5163
Epoch 4/48
225/225 [==============================] - 22s 100ms/step - loss: 1.1831 - accuracy: 0.5490 - val_loss: 1.2041 - val_accuracy: 0.5491
Epoch 5/48
225/225 [==============================] - 23s 100ms/step - loss: 1.1257 - accuracy: 0.5724 - val_loss: 1.2334 - val_accuracy: 0.5278
```

Fig 11: Testing Model Accuracy

6. Model Accuracy

The accuracy of model is represented in graphical form

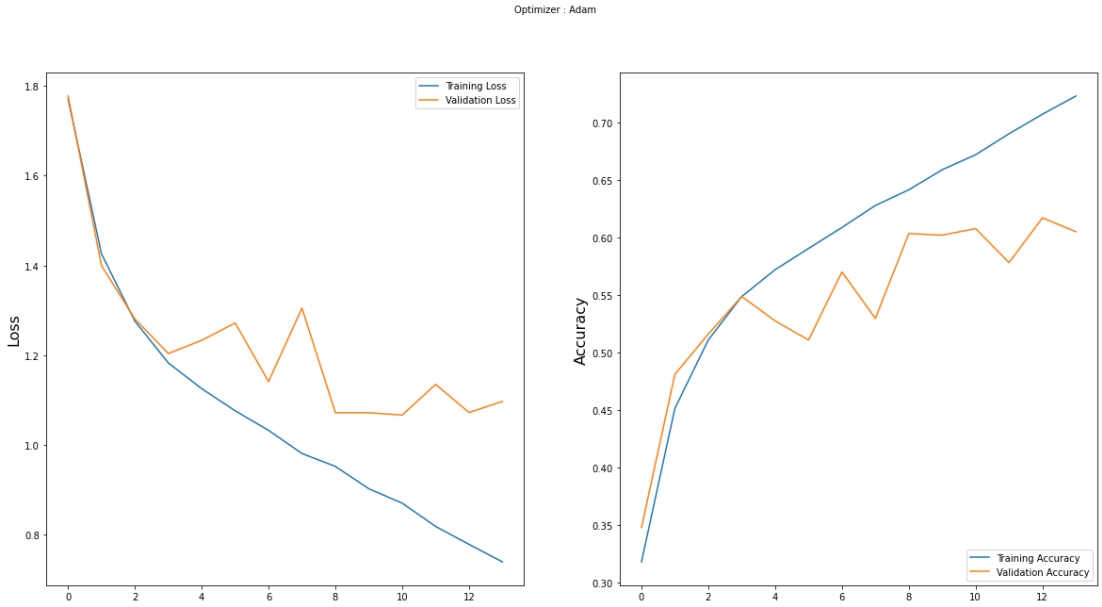


Fig 12: Accuracy graph

7. Implementing model on real time

Now we have a model with good accuracy , finally we can implement that model on videos using web cam.

main.py > ...

```python
from keras.models import load_model
from time import sleep
from tensorflow.keras.preprocessing.image import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np

face_classifier = cv2.CascadeClassifier(r'haarcascade_frontalface_default.xml')
classifier =load_model(r'C:\Users\Lenovo\Desktop\Minor\model.h5')

emotion_labels = ['Angry','Disgust','Fear','Happy','Neutral', 'Sad', 'Surprise']

cap = cv2.VideoCapture(0)



while True:
    _, frame = cap.read()
    labels = []
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray)

    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),2)
        roi_gray = gray[y:y+h,x:x+w]
        roi_gray = cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)



        if np.sum([roi_gray])!=0:
            roi = roi_gray.astype('float')/255.0
            roi = img_to_array(roi)
            roi = np.expand_dims(roi,axis=0)
```

Fig 13: Implementation of model on real time

CHAPTER V

RESULTS AND CONCLUSION

LTS AND CONCLUSION

5.1 RESULTS

The “Smart webcam motion detection surveillance system” is a home/office security system that can be very useful in situations when security is a concern. The methods used by thieves and robbers to steal have improved dramatically as a result of technological advancements in the modern world. As a result, surveillance systems must evolve to keep pace with the changing world. Video surveillance and monitoring are the most recent technologies utilized in the fight against theft and destruction. It is possible to monitor and collect every inch and second of the region of interest utilizing technology. In digital video surveillance systems, motion detection is the most crucial function. It allows the camera to capture only when necessary, rather than all of the time, resulting in a significant decrease in storage capacity. When unexpected motion is detected, an alarm can be activated.

Personnel are no longer required to monitor at all times as a result of this. Computerized home-based security can develop a lot with the coming future. Future is promising and easier with innovative technologies.

5.2. CONCLUSION

This review paper gives brief survey on various motion detection techniques. The basic motion detection methods are background subtraction, optical flow and clustering. An effective motion assessment and monitoring system has been developed for the improvement of the motion detection ability.

This method cannot be used in real time motion detection system implementation. So, in order to develop precious and optimize system, motion detection and background subtraction system is .Video surveillance and monitoring are the most recent technologies utilized in the fight against theft and destruction. It is possible to monitor and collect every inch and second of the region of interest utilizing technology. In digital video surveillance systems, motion detection is the most crucial function. It allows the camera to capture only when necessary rather than all of the time, resulting in a significant decrease in storage capacity. When unexpected motion is detected, an alarm can be activated. This relieves personnel from constant monitoring. The motion detector fills the requirement for a low-cost, modest security system in everyday life. Computerized home-based security has a lot of potential in the future. With breakthrough technologies, the future seems bright and easier.

CHAPTER VI

REFERENCES

REFERENCES

- A motion detection system in python and Opencv IEEE Paper (Suraiya Parveen &Javeria Shah) Jamia Hamdard University, New Delhi.
- Emotion Detection using image processing in python (raghav Puri,archit Gupta, manas sikri, Mohit Tiwari, Nitish Pathak, Shivendra goeal) Bharti Vidyapeeth's College of Engineering, New Delhi, India.
- MATEC web of Conference 2017.
- Phillipp Wagner July 2012(Face recognition with Python.
- An Intelligent motion detection using Opencv , International journal of scientific research in science, engineering and technology.
- A survey on object detection and tracking algorithm by Rupesh Kumar Rout, CSE Department NITRourkela.
- Real time abnormal function detection in surveillance video by Massachusetts Instituite of technology Cambridge University.
- K. Kopecky and E. Winer, „„MetaTracker: Integration and abstraction of 3D motion tracking data from multiple hardware systems, "" Proc. SPIE, vol. 9095, pp. 909507-1– 909507-11, May 2014.
- R. Al-Ali, Imran A. Zualkernan, Assia Lasfer, Alaa Chreide, and Hadel Abu Ouda, "GRPSBased Distributed Home-Monitoring Using Internet-Based Geographical Information System," IEEE Trans. Consumer Electron., vol. 57, no.4, pp. 1688-1694, Nov.2011.
- Farrukh Hijaz, Nabeel Afzal, Talal Ahmad, and Osman Hasan, "Survey of Fall Detection and Daily Activity Monitoring Techniques," 2010 International Conference on Information and Emerging Technologies (ICIET), pp.1-6, June 2010.
- Y. Zhao, and Z. Ye, "A Low-Cost GSM/GPRS Based Wireless Home Security System," IEEE Trans. Consumer Electron., vol. 54, no. 2, pp. 567-572, May 2008.