ISSN [Online]: 2583-2654

Anomaly detection in crowds using Multi Video Information

Vinutha B.A[1], Malavika R[2], Monisha R[3], Monisha R[4] and Monika R[5]
Department of Computer Science And Engineering ,
Dr.T.Thimmaiah Institute Of Technology ,VTU

Abstract: At present, the existing abnormal event detection models based on deep learning mainly focus on data represented by a vectorial form. But here, Anomaly detection in crowds using Multi video information is a system capable of detecting unusual activities in crowds from real-world data captured from multiple cameras. The detection is achieved by classifying the distinct movements of people in crowds, and those patterns can be different and can be classified as normal and abnormal activities. Statistical features are extracted from the data set collected by applying sliding time window operations. A model for classifying movements is trained by using Deep learning technique.

The system was tested by using two data sets collected from CCTV during social events gathering. Results show that data can be used to detect anomalies in crowds as an alternative to video sensors with significant performances. Our approach is the first to detect any unusual behavior in crowd with non-visual data, which is simple to train and easy to deploy.

Security is always a main concern in every domain, due to a rise in crime rate in a crowded event or suspicious lonely areas. Abnormal detection and monitoring have major applications of computer vision to tackle various problems. Due to growing demand in the protection of safety, security and personal properties, needs and deployment of video surveillance systems can recognize and interpret the scene and anomaly events play a vital role in intelligence monitoring.

Keywords:Abnormal Event Detection , re-sized , gray scaled , convolutional neural network, image classification , testing and training , predict performance , accuracy .

I. INTRODUCTION

Public places like airports, train stations, theme parks and shopping centers are typically crowded. The flow of people is growing consistently each year with the development of urbanization.

This growth causes an increase in the risk of crowd stampedes in public places. To avoid these dangerous situations, the crowd should be monitored, and for detecting in a timely manner, any occurrence of unusual situations. Public safety and security are main reasons for crowd analysis. Human operators continuously observe the visual screens for detecting any event of interest, which becomes challenging to tackle all the time. This motivates researchers to develop an automatic system for detecting anomalous activities insides crowds and facilitating the operators.

There has been extensive research made in the area of anomalies detection in crowds by the computer vision and signal processing communities.

Recently, some attempts have been made by exploiting deep learning models to avoid any complex hand-crafted feature extraction and processing methods. Despite substantial research work are success in this area, there are deficiencies like ground truth availability, anomaly type etc as discussed in.The computer vision community still faces a number of challenges to develop efficient anomaly detection systems. This includes unavailability of cameras, adverse weather conditions, night vision issues etc. All these obstacles often result in missing the target event and occurrence of interest of instances.

In order to deal with such issues, surveillance cameras with better accuracy are good alternatives for certain problems as discussed in. The proliferation of smart phones has enabled the signal processing community to analyze and comprehend the dynamics of crowds. The data processed from these smart phones can be exploited to reveal the information about individual behavior and groups dynamics inside a crowd. Smart phones are equipped with sensors like accelerometer, gyroscope, Bluetooth, proximity sensor, camera, ambient light sensor etc. These sensors acquire data which helps in understanding the behavior of individuals in crowds.

A lot of work can be found related to single human activity recognition systems. These systems are also capable of classifying different activities like walking, sitting, laying, sleeping etc.

The main objective of the system developed and proposed by this paper is to detect unusual activities in a crowd by simultaneously processing non-visual data synchronously acquired from a bunch of individuals smart phones. To our knowledge, this is the first work that exploits non-visual data acquired from mobile accelerometer and gyroscopes to detect group based anomaly in crowds.

One of the main contributions of this paper is the use of smart phones embedded with rich sensors to gather data collectively and detect anomalous behavior in crowds. Moreover, this paper provides a complete data set along with video based ground truth, acquired from accelerometer, gyroscope and multiple fixed cameras, which could be used to analyze behavior of individuals in the crowd for research purposes.

II. SYSTEM ARCHITECTURE AND DESIGN

A system architecture diagram as shown in the below figure 2.1 that would be used to show the relationship between different components. Usually they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them.

2.1 Video Data Sets

It is a collection of data usually presented in tabular form. Each column will represent a particular variable. Each row corresponds to a given member of the data set. The action recognition system works on a subset of a data set called UCF-11.This is basically a collection of 11 video clips organized into different categories based on the action they depict. We can take example of biking, golf , trampoline jumping.

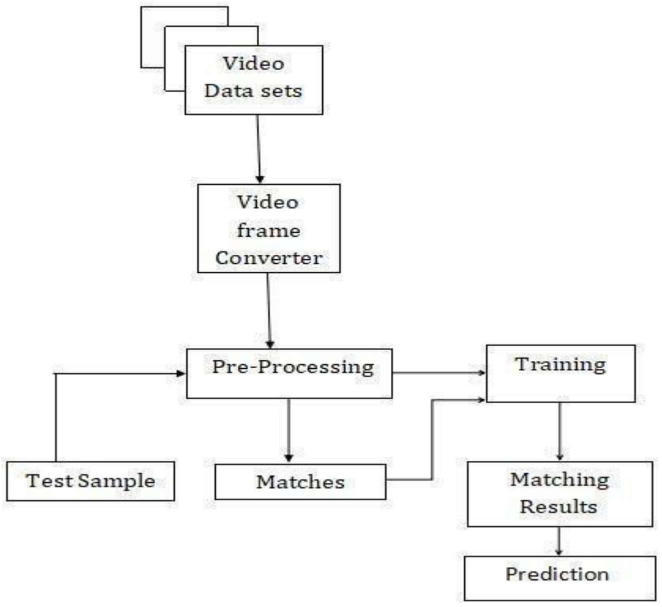


Fig 2.1: System Architecture

2.2 Video Frame Converter

It allows you to convert the video to an image sequence. For example lets upload a 3 second video of size 3MB and 53 frames per second. And choose the part of the video that you want to convert.

It will display a sequence of image of size 3KB and frame size =720x1280 = 9,21,600. where, 720 is height and 1280 is width. since we cannot process large data we take each frame as 10,000 element as representative of the actual frame size

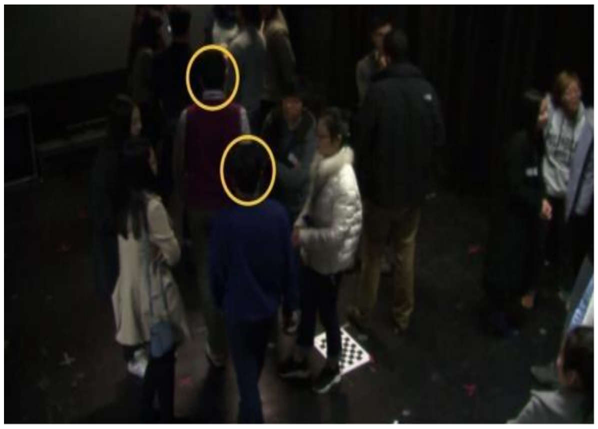


Fig 2.2: Example Dataset

2.3 Pre-Processing

The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images example: rotation, scaling, translation are classified among pre-processing methods here since similar techniques are used. Here we transform the 10,000 frame in order to extract its feature.

2.4 Training

Training data is the data that used to train an algorithm or machine learning model to predict the outcome you design your model to predict.

2.5 Test Sample

The test set should be used to see how well your model performs on unseen data. once the model is trained it is possible to carry out model testing. At this time test set of data is loaded.

2.6 Matches

This process is used to find a piece of matching information in a large sets of data.

2.7 Matching Result

The current model will run with the training data set and produces result, which is then compared with the target for each input vector in the training data sets.

2.8 Prediction

It refers to an output of an algorithm after it has been trained on a historical data set and applied to new data when forecasting the likelihood of a particular outcome.

It can be applied to any type of unknown event, regardless when it occurred they often used to detect crimes and identify suspects after the crime has taken place.

Example: we may use a model to determine email is spam or non-spam similarly we use it here for crime detection. where prediction is the comparison of future behavior.

III. ALGORITHM

CNN Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. CNN is also computationally efficient.It uses special convolution and pooling operations and performs parameter sharing. This enables CNN models to run on any device, making them universally attractive. There is an input image that we‘re working with. We perform a series convolution + pooling operations, followed by a number of fully connected layers. If we are performing multiclass classification the output is softmax.

3.1 Algorithm Architecture

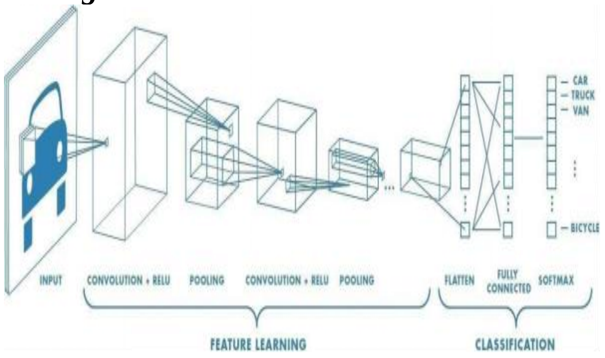


Fig 3.1 : Algorithm Architecture

From the above figure (3.1) we will be explaining about feature extraction and its classification as follows:

3.1.1 Feature Extraction

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power. After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

3.1.2 Classification

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space. Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and back propagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique**.**

IV METHODOLOGY

Methodology is a collection of methods, practices, processes, techniques, procedures, and rules. It is contextual framework for research, a coherent and logical scheme based on views, believes and values, that guides the choices researchers or other users make.

There are generally three major steps as shown in the below figure (4.1):

Planning
Implementing
Analysis

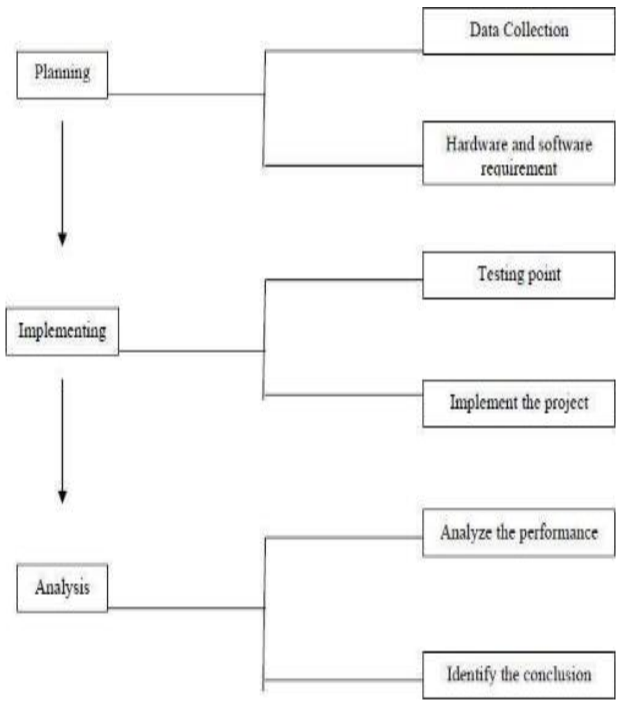


Fig 4.1: Steps of Methodology

4.1 Planning

To identify all the information and requirement such as hardware and software, planning must be done in the proper manner.

The planning phase has two main elements namely data collection and the requirements of hardware and the software as discussed above.

4.2 Data Collection

Machine learning needs two things to work, data (lots of it) and models. When acquiring the data, be sure to have enough features (aspect of data that can help for a prediction, like the surface of the house to predict its price) populated to train correctly your learning model. In general, the more data you have the better so make to come with enough rows.

The primary data collected from the online sources remains in the raw form of statements, digits and qualitative terms. The raw data contains error, omissions and inconsistencies. It requires corrections after careful scrutinizing the completed questionnaires.

The following steps are involved in the processing of primary data. A huge volume of raw data collected through field survey needs to be grouped for similar details of individual responses. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Therefore, certain steps are executed to convert the data into a small clean data set. This technique is performed before the execution of Iterative Analysis. The set of steps is known as Data Pre-processing. It includes –

Data Cleaning

Data Integration

Data Transformation

Data Reduction

Data Preprocessing is necessary because of the presence of unformatted real-world data. Mostly real-world data is composed of –

Inaccurate data (missing data) – There are many reasons for missing data such as data is not continuously collected, a mistake in data entry, technical problems with biometric and much more.

The presence of noisy data (erroneous data and outliers) - The reasons for the existence of noisy data could be a technological problem of gadget that gathers data, a human mistake during data entry and much more.

Inconsistent data - The presence of inconsistencies are due to the reasons such that existence of duplication within data, human data entry, containing mistakes in codes or names, i.e. violation of data constraints and much more.

4.3 Implemention

In this work, a business intelligent model has been developed, to classify different crimes, based on a specific business structure deal with Anomaly classification using a suitable deep learning technique. The model was evaluated by a scientific approach to measure accuracy. We are using Convolutional Neural Network (CNN) to build our model.

4.3.1 Convolutional Neural Network

A convolutional neural network (CNN) is a special architecture of artificial neural networks, proposed by Yann LeCun in 1988. CNN uses some features of the visual cortex. One of the most popular uses of this architecture is image classification. For example Facebook uses CNN for automatic tagging algorithms, Amazon—for generating product recommendations and Google for search through among users‘ photos. Let us consider the use of CNN for image classification in more detail. The main task of image classification is acceptance of the input image and the following definition of its class. This is a skill that people learn from their birth and are able to easily determine that the image in the picture is an elephant as shown in figure 4.3.1.But the computer sees the pictures quite differently:

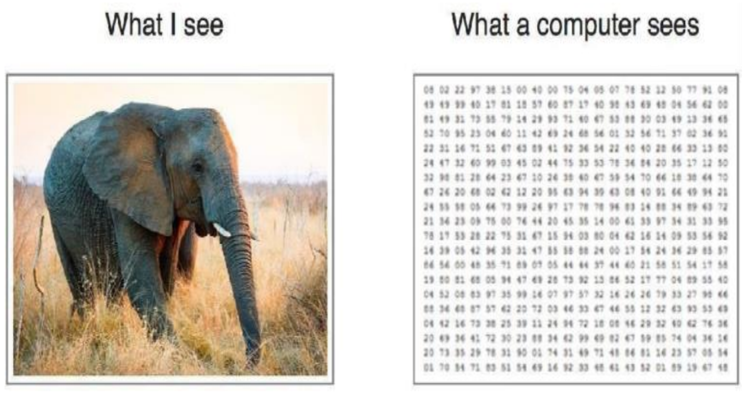


Fig 4. 3.1: Image Classification

Instead of the image, the computer sees an array of pixels. For example, if image size is 300 x 300. In this case, the size of the array will be 300x300x3. Where 300 is width, next. 300 is height and 3 is RGB channel values. The computer is assigned a value from 0 to 255 to each of these numbers. This value describes the intensity of the pixel at each point.

To solve this problem the computer looks for the characteristics of the base level. In human understanding such characteristics are for example the trunk or large ears. For the computer, these characteristics are boundaries or curvatures. And then through the groups of convolutional layers the computer constructs more abstract concepts.

In more detail: the image is passed through a series of convolutional, nonlinear, pooling layers and fully connected layers, and then generates the output.

The Convolution layer is always the first. The image (matrix with pixel values) is entered into it. Imagine that the reading of the input matrix begins at the top left of image. Next the software selects a smaller matrix there, which is called a **filter** (or neuron, or core) as shown in the below figure(4.3.2). Then the filter produces convolution, i.e. moves along the input image. The filter‘s task is to multiply its values by theoriginal pixel values. All these multiplications are summed up.

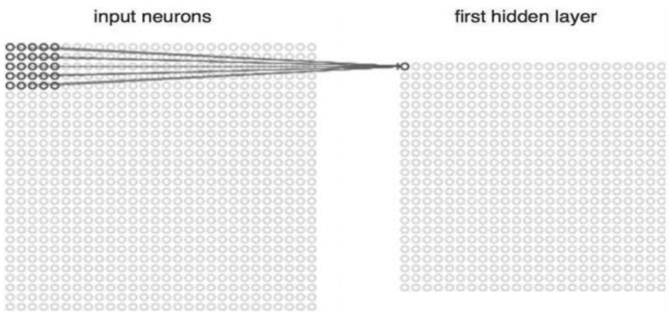


Fig 4. 3.2: Image Matrix

One number is obtained in the end. Since the filter has read the image only in the upper left corner, it moves further and further right by 1 unit performing a similar operation. After passing the filter across all positions, a matrix is obtained, but smaller then an input matrix.

The network will consist of several convolutional networks mixed with nonlinear and pooling layers. When the image passes through one convolution layer, the output of the first layer becomes the input for the second layer. And this happens with every further convolutional layer.

The nonlinear layer is added after each convolution operation. It has an activation function, which brings nonlinear property. Without this property a network would not be sufficiently intense and will not be able to model the response variable (as a class label).

The pooling layer follows the nonlinear layer. It works with width and height of the image and performs a down sampling operation on them. As a result the image volume is reduced. This means that if some features (as for example boundaries) have already been identified in the previous convolution operation, than a detailed image is no longer needed for further processing, and it is compressed to less detailed pictures.

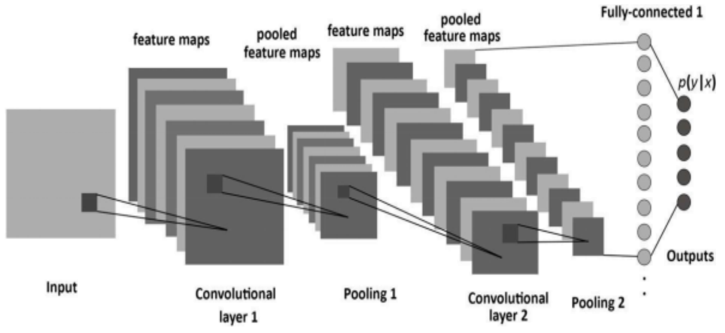


Fig 4. 3.3: Convolutional Neural Network

After completion of series of convolutional, nonlinear and pooling layers, it is necessary to attach a fully connected layer. This layer takes the output information from convolutional networks. Attaching a fully connected layer to the end of the network results in an N dimensional vector, where N is the amount of classes from which the model selects the desired class.

4.4 Modules

4.4.1 Image Acquisition And Pre-Processing

In this module we will get the data from the online source. Further we will resize the image for future use. Image re-sizing, or image scaling, is a geometric image transformation which modifies the image size based on an image interpolation technique. This image scaling process can increase or decrease the resolution of a target image so that the absolute size of image data is adjusted. Computers are able to perform computations on numbers and are unable to interpret images in the way that we do.

We have to somehow convert the images to numbers for the computer to understand. The image will be converted to gray scale (range of gray shades from white to black) the computer will assign each pixel a value based on how dark it is. All the numbers are put into an array and the computer does computations on that array. We then feed the resulting array for next step.

4.4.2 Data Preparation And Model Construction

Many a times, people first split their data set into two Train and Test. After this, they keep aside the Test set, and randomly choose X% of their Train data set to be the actual Train set and the remaining (100-X)% to be the Validation set, where X is a fixed number, the model is then iteratively trained and validated on these different sets. So we will follow the same method to prepare data for training and testing phase. We are building our model by using Convolutional neural network. Convolutional neural networks (CNN) are a special architecture of artificial neural networks, proposed by Yann LeCun in 1988. CNN uses some features of the visual cortex. Now that we‘re done pre-processing, we can start implementing our neural network. We‘re going to have 3 convolution layers with 2 x 2 max- pooling.

Max-pooling -A technique used to reduce the dimensions of an image by taking the maximum pixel value of a grid. This also helps reduce over fitting and makes the model more generic. After that, we add 2 fully connected layers. Since the input of fully connected layers should be two dimensional, and the output of convolution layer is four dimensional, we need a flattening layer between them. At the very end of the fully connected layers is a softmax layer.

4.4.3 Model Training

After model construction it is time for model training. we are able to build an artificial convolutional neural network that can recognize images. Split the data set into train and test data set. finally we will build and train the model using training data set.

4.4.4 Model Testing And Evaluation

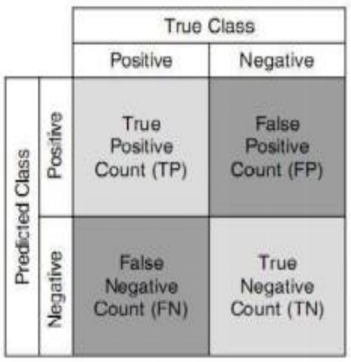
Once the model has been trained it is possible to carry out **model testing.** During this phase a test set of data is loaded. This data set has never been seen by the model and therefore its true accuracy will be verified. Finally, the saved model can be used in the real world. The name of this phase is **model evaluation**. This means that the model can be used to evaluate new data.

4.4.5 Analysis

In this final phase, we will test our classification model on our prepared image data set and also measure the performance on our data set. To evaluate the performance of our created classification and make it comparable to current approaches, we use accuracy to measure the effectiveness of classifiers.

One might even try multiple model types for the same prediction problem, and then, would like to know which model is the one to use for the real-world decision making situation, simply by comparing them on their prediction performance (e.g., accuracy). To measure the performance of a predictor, there are commonly used performance metrics, such as accuracy, recall etc.

First, the most commonly used performance metrics will be described, and then some famous estimation methodologies are explained and compared to each other.



$$\text{True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{True Negative Rate} = \frac{TN}{TN + FP}$$

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

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Fig 4.4.1: Confusion Matrix and Formulae

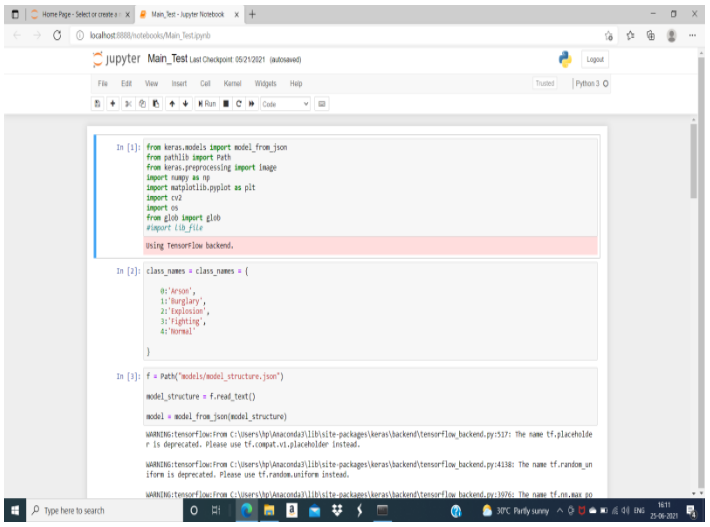
As being seen in above figure (4.4.1), the numbers along the diagonal from upper-left to lower-right represent the correct decisions made, and the numbers outside this diagonal represent the errors.

"The true positive rate (also called hit rate or recall) of a classifier is estimated by dividing the correctly classified positives (the true positive count) by the total positive count. The false positive rate (also called a false alarm rate) of the classifier is estimated by dividing the incorrectly classified negatives (the false negative count) by the total negatives. The overall accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the total number of samples.

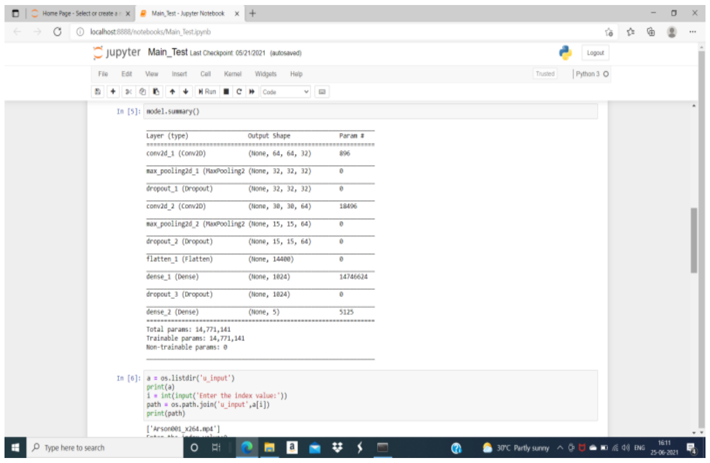
V RESULT AND ANALYSIS

A result is the final consequence of actions or events expressed qualitatively or quantitatively. Performance analysis is an operational analysis, is a set of basic quantitative relationship between the performance quantities.

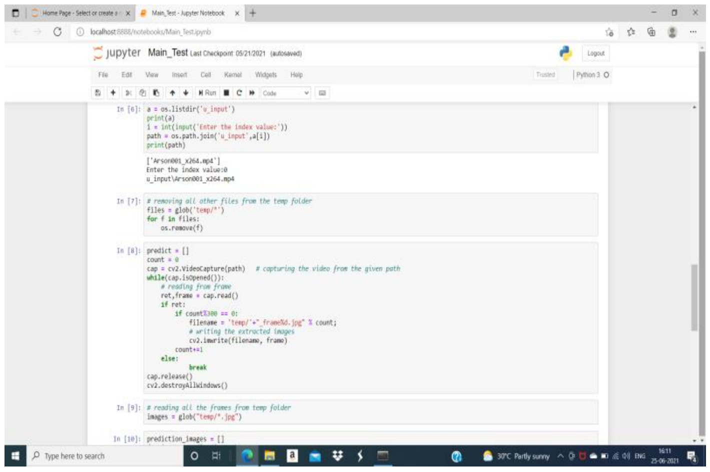
5.1 Screenshots


Home Page - Select or create a
Main_Test - Jupyter Notebook
localhost:8888/notebooks/Main_Test.ipynb
jupyter Main_Test Last Checkpoint: 05/21/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
30°C Partly sunny

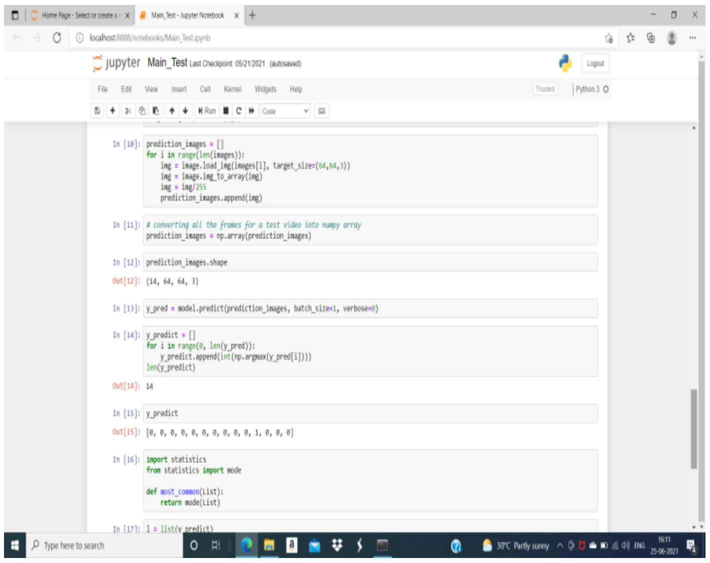
```
In [1]: from keras.models import model_from_json
        from pathlib import Path
        from keras.preprocessing import image
        import numpy as np
        import matplotlib.pyplot as plt
        import cv2
        import os
        from glob import glob
        #import lib_file

        Using TensorFlow backend.

In [2]: class_names = class_names = {

            0:'Arson',
            1:'Burglary',
            2:'Explosion',
            3:'Fighting',
            4:'Normal'

        }

In [3]: f = Path("models/model_structure.json")

        model_structure = f.read_text()

        model = model_from_json(model_structure)

        WARNING:tensorflow:From C:\Users\hp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.placeholde
        r is deprecated. Please use tf.compat.v1.placeholder instead.

        WARNING:tensorflow:From C:\Users\hp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.random_un
        iform is deprecated. Please use tf.random.uniform instead.

        WARNING:tensorflow:From C:\Users\hp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3976: The name tf.nn.max_po
```


Home Page - Select or create a
Main_Test - Jupyter Notebook
localhost:8888/notebooks/Main_Test.ipynb
jupyter Main_Test Last Checkpoint: 05/21/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
30°C Partly sunny

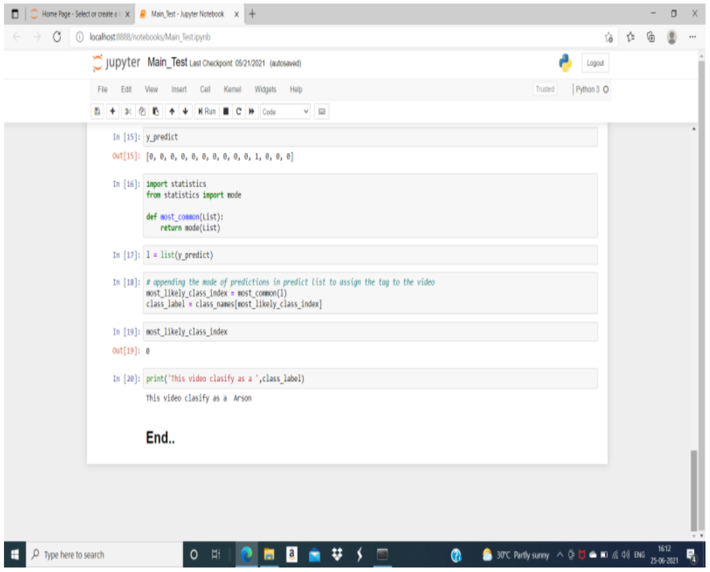
```
In [5]: model.summary()

        _________________________________________________________________
        Layer (type)                 Output Shape              Param #
        =================================================================
        conv2d_1 (Conv2D)            (None, 64, 64, 32)        896
        _________________________________________________________________
        max_pooling2d_1 (MaxPooling2 (None, 32, 32, 32)        0
        _________________________________________________________________
        dropout_1 (Dropout)          (None, 32, 32, 32)        0
        _________________________________________________________________
        conv2d_2 (Conv2D)            (None, 30, 30, 64)        18496
        _________________________________________________________________
        max_pooling2d_2 (MaxPooling2 (None, 15, 15, 64)        0
        _________________________________________________________________
        dropout_2 (Dropout)          (None, 15, 15, 64)        0
        _________________________________________________________________
        flatten_1 (Flatten)          (None, 14400)             0
        _________________________________________________________________
        dense_1 (Dense)              (None, 1024)              14746624
        _________________________________________________________________
        dropout_3 (Dropout)          (None, 1024)              0
        _________________________________________________________________
        dense_2 (Dense)              (None, 5)                 5125
        =================================================================
        Total params: 14,771,141
        Trainable params: 14,771,141
        Non-trainable params: 0
        _________________________________________________________________

In [6]: a = os.listdir('u_input')
        print(a)
        i = int(input('Enter the index value:'))
        path = os.path.join('u_input',a[i])
        print(path)

        ['Arson001_x264.mp4']
```


Home Page - Select or create a
Main_Test - Jupyter Notebook
localhost:8888/notebooks/Main_Test.ipynb
jupyter Main_Test Last Checkpoint: 05/21/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
30°C Partly sunny

```
In [6]: a = os.listdir('u_input')
        print(a)
        i = int(input('Enter the index value:'))
        path = os.path.join('u_input',a[i])
        print(path)

        ['Arson001_x264.mp4']
        Enter the index value:0
        u_input\Arson001_x264.mp4

In [7]: # removing all other files from the temp folder
        files = glob('temp/*')
        for f in files:
            os.remove(f)

In [8]: predict = []
        count = 0
        cap = cv2.VideoCapture(path)   # capturing the video from the given path
        while(cap.isOpened()):
            # reading from frame
            ret,frame = cap.read()
            if ret:
                if count%300 == 0:
                    filename = 'temp/'+"_frame%d.jpg" % count;
                    # writing the extracted images
                    cv2.imwrite(filename, frame)
                count+=1
            else:
                    break
        cap.release()
        cv2.destroyAllWindows()

In [9]: # reading all the frames from temp folder
        images = glob("temp/*.jpg")

In [10]: prediction_images = []
```



```
In [10]: prediction_images = []
         for i in range(len(images)):
             img = image.load_img(images[i], target_size=(64,64,3))
             img = image.img_to_array(img)
             img = img/255
             prediction_images.append(img)

In [11]: # converting all the frames for a test video into numpy array
         prediction_images = np.array(prediction_images)

In [12]: prediction_images.shape

Out[12]: (14, 64, 64, 3)

In [13]: y_pred = model.predict(prediction_images, batch_size=1, verbose=0)

In [14]: y_predict = []
         for i in range(0, len(y_pred)):
             y_predict.append(int(np.argmax(y_pred[i])))
         len(y_predict)

Out[14]: 14

In [15]: y_predict

Out[15]: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]

In [16]: import statistics
         from statistics import mode

         def most_common(List):
             return mode(List)

In [17]: l = list(y_predict)
```



```
In [15]: y_predict

Out[15]: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]

In [16]: import statistics
         from statistics import mode

         def most_common(List):
             return mode(List)

In [17]: l = list(y_predict)

In [18]: # appending the mode of predictions in predict list to assign the tag to the video
         most_likely_class_index = most_common(l)
         class_label = class_names[most_likely_class_index]

In [19]: most_likely_class_index

Out[19]: 0

In [20]: print('This video clasify as a ',class_label)

         This video clasify as a  Arson
```

End..

VI CONCLUSION

This system uses supervised data set that means labeled data set we have to label the input videos by class name and give to CNN training model once CNN training model is built we can give a test video as a input the system has to classify the input video is contained any crime activities or not, if it found crime activity it has to send a alert message to police control system

Acknowledgement

We would like to thank Dr.S.Sreedhar Kumar , Mrs.Vinutha B.A , and Mrs.Mercy Flora for the useful help.

REFERENCES

[1] *Ahmad salihu Ben-mus, sanjay kumar singh, prateek Agarwal, **"Suspicious Recognition for video Surveillance System"**, International Journal on Internet and Distributed Computing Systems. Vol. 2, No. 2, PP.141-148, January 2014.*

[2] *Alkesh Bharati, Dr Sarvanaguru R.A, **"Crime Prediction and Analysis Using Machine Learning"**, International Research Journal of Engineering and Technology, Vol.05, PP.406-412, September 2018.*

[3] *Panagiotis Stalids, Theodoros Semertzidis, **"Examining Deep Learning Architectures for Crime Classification and prediction"**, In journal of Machine Learning Research Vol.3, PP.2825-2830, October 2018.*

[4] *Gallavee, Lati furkhan, Bhavani Thurai singham ,**"A framework for a video analysis is tool for suspicious event detection"**,Conference on Multimed Tools And Application, PP.109-123, April 25, 2007.*

[5] *S Prabakaran and Shilpa Mitra, **"Survey of Analysis of Crime Detection Techniques Using Data Mining and Machine Learning"**, International conference on Networks and soft computing, PP.406-412, September 2014.*

[6] *C. Lu, J. Shi, and J. Jia, **"Abnormal event detection at 150 FPS in MATLAB"**,International conference on Computer Vision , sydney, NSW, Australia, PP.2720-2727, December 2013.*

[7] *A. D. Giorno, J. A. Bagnell, and M. Hebert, **"A discriminative framework for anomaly detection in large videos"**, International conference on computer vision, Amsterdam, The Netherlands, PP.334-349, October 2016.*

[8] *J. Li, Y. Tian, T. Huang, and W. Gao, **"Probabilistic multi-task learning for visual saliency estimation in videos"**, International journal on Computer Vision, Vol. 90, No.2, PP.150-165, November 2010.*

[9] *Y. Fu, T. M. Hospedales, T. Xiang, S. Gong, and Y. Yao, **"Interestingness prediction by robust learning to rank"**, International Conference on computer vision, Zurich, Switzerland, PP.488-503, September 2014.*

[10] *M. J. Roshtkhari and M. D. Levine, **"Online dominant and anomalous behavior detection in videos"**, IEEE Conference Computer Vision and Pattern Recognition, Portland, OR, USA, PP. 2611-2618, June 2013.*

[11] *Ouye, Jun Deng, Zhenhuayu, Tao Liu And Lihong Dong, **"Abnormal Event Detection via Feature Expectation Sub graph Calibrating Classification in Video Surveillance Scenes"**, International journal on Multimedia ubiquitous engineering, Vol.10, No.9, PP.339-352, June 5,2020.*

[12] *W. Chu, H. Xue, C.Yao, and D. Cai, **"Sparse coding guided spatio-temporal feature learning for abnormal event detection in large videos"**, IEEE Transaction on multimedia, Vol. 21, No.1, PP.246-255, January 2019.*

[13] *Tanu Gupta, Vimala Nunavath and Sudip Roy, **"A Deep-CNN Based Frame work detect Abnormal Crowd-Motion Behavior in Videos for Predicting Crowd Disaster"**, IEEE International Conference on Systems, Man and Cybernetics Bari, Italy, PP.2877-2882, October 6, 2019.*

[14] *T. Fernando, S. Denman, S. Sridharan, and C. Fookes,* ***"Soft + Hardwired attention An LSTM framework for human trajectory prediction and abnormal event detection"****, Journal on Neural Network., Vol.108, PP.466-478, December 2018.*

[15] *Wang Huan, Huiwen Guo, Xinyu Wu,* ***"Saliency Attention Based Abnormal Event Detection in Video"****, International Journal on Multimedia Ubiquitous Engineering, Vol.10, No. 9, PP.339-352 ,October 2015.*

[16] *R. Ye and X. Li,****"Collective representation for abnormal event detection"****, Journal on Computer Science and Technology, Vol.32, No. 3, PP.470-654, May 2017.*