**Intelligent Video Surveillance Using Deep Learning System**

**Project Report**

Project Work Phase – II (EAI 852)

**BACHELOR OF TECHNOLOGY (CSE)**

|  |  |
| --- | --- |
| PROJECT GUIDE:  **Mr. Ashish Bishnoi** | SUBMITTED BY:  **Aviral Jain (TCA1959029)**  **Saksham Jain (TCA1959030)** |

May, 2023



**COLLEGE OF COMPUTING SCIENCES & IT**

**TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD**

**ACKNOWLEDGEMENT**

First and foremost, we would like to express our sincere gratitude to our project guide Dr. Saurabh Pathak Sir and Ashish Bishnoi Sir who has helped us in this endeavor and has always been very cooperative and without his help, cooperation, guidance and encouragement, this project couldn’t have been what it evolved to be.

We extend our heartfelt thanks to our faculty for their guidance and constant support as well as, for providing us the necessary information regarding the project.

We are also thanks to our parents for their cooperation and encouragement.

At last but not the least, gratitude to all our friends who helped us (directly or indirectly) to complete this project within the limited time frame.

Aviral Jain Saksham Jain

TCA1959029 TCA1959030

**DECLARATION**

We hereby declare that this Project Report titled **Intelligent Video Surveillance Using Deep Learning System** submitted by us and approved by our project guide, the College of Computing Sciences and Information Technology (CCSIT), Teerthanker Mahaveer University, Moradabad, is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

|  |  |  |
| --- | --- | --- |
| **Student Name:** | Aviral Jain |  |
| **Student Name:** | Saksham Jain |  |
| **Project Guide :** | Mr. Ashish Bishnoi |  |

Table of Contents

[1 Project Title 5](#_Toc134132315)

[2 Problem Statement 5](#_Toc134132316)

[3 Project Description 6](#_Toc134132317)

[3.1 Scope of the Work 6](#_Toc134132318)

[3.2 Project Modules 7](#_Toc134132319)

[3.3 Context Diagram (High Level) 7](#_Toc134132320)

[4 Implementation Methodology 8](#_Toc134132321)

[5 Technologies to be used 16](#_Toc134132322)

[5.1 Software Platform 16](#_Toc134132323)

[5.2 Hardware Platform 23](#_Toc134132324)

[5.3 Tools, if any 23](#_Toc134132325)

[What can Python do? 23](#_Toc134132326)

[6 Advantages of this Project 24](#_Toc134132327)

[7 Assumptions, if any 25](#_Toc134132328)

[8 Future Scope and further enhancement of the Project 25](#_Toc134132329)

[9 Project Repository Location 26](#_Toc134132330)

[10 Definitions, Acronyms, and Abbreviations 27](#_Toc134132331)

[11 Conclusion 27](#_Toc134132332)

[12 References 28](#_Toc134132333)

**Appendix**

**A: Screen Shots**

# Project Title

Intelligent Video Surveillance Using Deep Learning System is a python application which is used for Image Classification. It captures the video and gives the result for the activity in the videos. It uses CNN approach for classification purpose and provide with the recognized output.

# Problem Statement

In current era, where so many security measures are apply for everyone’s safety such as police, department for every affairs. So many cities are installing CCTV for recording every movement in the city and have record for every incident happening in the city. But it has some faults in it, as it cannot alarm the officer or the department at real time while any abnormal activity is occurring or happening

So the main problem is arrive:

How to detect abnormal activities at real time using CCTV surveillance?

To create this project on Image Classification, there are 4 distinct phases:-

* Collect and preprocess the data.
* Train the model.
* Test and refine the model.

Input Image

Deep Learning Model

Output Result

(Classified Image)

# Project Description

The goal of this project is to develop a python project that can perform image classification in a specific domain. The script will allow users to record a real-time video through their device and classify it into one of several pre-defined categories based on the video content.

The project will involve collecting and preprocessing a large dataset of images that are relevant to the specific domain, training a machine learning model using the dataset, and integrating the model into the python script using multiple libraries. The output will also include a user interface that displays the real-time video and the predicted class label.

The script will have several potential use cases, such as identifying abnormal human activity in a region. By using machine learning to perform image classification, the script will be able to provide more accurate and efficient results than traditional methods, such as manual labeling or keyword searches.

The success of the project will be measured by the accuracy and efficiency of the image classification model, as well as the usability and functionality of the python script. The final deliverables will include the trained machine learning model, the python code, and a report documenting the design and implementation of the project, as well as its potential use cases and future improvements.

## Scope of the Work

In the proposed system, we have done abnormal activity analysis with many parameters and factors including object detection, Motion detection. Using machine learning algorithm and CNN algorithm, we are predicting the type of abnormal activity for the given real-time video. As an Outcome of our solution we can detect the rime prone areas and anomaly activities areas on the basis of available factors which will facilitate in taking preventive actions against crime in such areas and hence crime will gradually decrease.

The project will involve collecting and pre-processing a large dataset of images that are relevant to the specific domain, training a machine learning model using the dataset, and integrating the model into the python script using Tensor Flow and Keras.

The main objective of this project is to provide a solution for accurate and efficient image classification in a specific domain, which can be used in various industries such as security, automotive and crime detection.

This project can serve as a foundation for further research and development in the field of machine learning and image classification. The trained model can be further optimized to improve accuracy and speed, and the app can be extended to include additional features such as image segmentation or object detection.

Overall, this project has the potential to make significant contributions to the field of mobile image classification and has the potential to be adopted in various industries for a range of applications.

## Project Modules

**Data Collection and Pre-processing Module:**

This module involves collecting and pre-processing a large dataset of images that are relevant to the specific domain. The data pre-processing step involves data cleaning, normalization, and augmentation, which are essential for the success of the machine learning model.

**Machine Learning Model Training Module:**

This module involves training a machine learning model using the pre-processed dataset. The model will be designed to perform image classification, with the ability to classify images into several pre-defined categories. The model training module will also include model evaluation and optimization.

**Model Integration Module:**

This module involves integrating the trained machine learning model into the python exe using pyinstaller. This will involve loading the model into the python and using it to perform image classification on input video. The model integration module will also include model optimization and memory management.

**Testing and Refinement Module:**

This module involves testing the script to ensure that it is functioning correctly and refining the model or script as necessary to improve accuracy or usability. The testing and refinement module will also include features such as error handling, logging, and debugging.

## Context Diagram (High Level)

Interactive UI

Application

**0 – Level DFD**

# Implementation Methodology

**STEPS INVOLVED IN FACE RECOGNITION**

* **Detect motion and object:** When given a frame/video, the CNN algorithm first detects for faces in each frame.
* **Classify into one of the 13 age classes** : In the third layer of CNN, the class of the activity is determined and falls under either of the 13 classes [(Abuse), (Arrest), (Assault), (Arson), (Burglary), (Explosion), (Fighting), (Normal),( Accident),( Robbery),( Shooting),( Shop Lifting),( Stealing)].
* **Put the results on the frame and display it:**  The result is displayed on the frame containing the classes and outcome percentage using OpenCV. The resulting frame consists of a square box around the video with the estimated class and respective accuracy.

**Codes for Fast Image Retrieval**

To create the hash-like binary codes it provides effective framework for fast image retrieval.

**SegNet and Bayesian SegNet**

SegNet is real-time semantic segmentation architecture for scene understanding.

**DeepYeast**

Deep Yeast may be an 11-layer convolutional neural network trained on biaural research pictures of yeast cells carrying fluorescent proteins with totally different subcellular localizations. Python VS other languages for Object Detection: Object detection may be a domain-specific variation of the machine learning prediction drawback. Intel’s OpenCV library that is implemented in C/C++ has its interfaces offered during a} very vary of programming environments like C#, Matlab, Octave, R, Python and then on. Why Python codes are much better option than other language codes for object detection are more compact and readable code. Python uses zero-based indexing. Dictionary (hashes) support provided. Simple and elegant Object-oriented programming Free and open multiple functions can be package in one module more choices in graphics packages and toolsets Supervised learning also plays an important role. The utility of unsupervised pre-training is usually evaluated on the premise of what performance is achieved when supervised fine-tuning. This paper reviews and discusses the fundamentals of learning as well as supervised learning for classification models, and also talks about the mini batch stochastic gradient descent algorithm that is used to fine-tune many of the models. Object Classification in Moving Object Detection Object classification works on the shape, motion, color and texture.

**Shape-Based**

A mixture of image-based and scene based object parameters such as image blob (binary large object) area, the as pectration of blob bounding box and camera zoom is given as input to this detection system. Classification is performed on the basis of the blob at each and every frame. The results are kept in the histogram.

**Motion-Based**

When an easy image is given as an input with no objects in motion, this classification isn't required. In general, non- rigid articulated human motion shows a periodic property; therefore this has been used as a powerful clue for classification of moving objects. Based on this useful clue, human motion is distinguished from different objects motion. ColorBased- though color isn't an applicable live alone for police investigation and following objects, but the low process value of the colour primarily based algorithms makes the coloura awfully smart feature to be exploited. As an example, the color histogram based technique is employed for detection of vehicles in period. Color bar chart describes the colour distribution in a very given region that is powerful against partial occlusions.

**Texture-Based**

The texture-based approaches with the assistance of texture pattern recognition work just like motion based approaches. It provides higher accuracy, by exploitation overlapping native distinction social control however might need longer, which may be improved exploitation some quick techniques. Proposed WORK Authors have applied period object detection exploitation deep learning and OpenCV to figure to work with video streams and video files. This will be accomplished using the highly efficient open computer vision. Implementation of proposed strategy includes caffe- model based on Google Image Scenery; Caffe offers the model definitions, optimization settings, pre- trained weights. Prerequisite includes Python 3.7, OpenCV 4 packages and numpy to complete this task of object detection. NumPy is the elementary package for scientific computing with Python. It contains among other things: a strong N-dimensional array object, subtle (broadcasting) functions tools for integrating C/C++ and fortran code, helpful linear algebra, Fourier transform, and random 36 number capabilities. Numpy works in backend to provide statistical information of resemblance of object with the image scenery caffemodel database. Object clusters can be created according to fuzzy value provided by NumPy. This project can detect live objects from the videos and images.

**LEARNING FEATURE HIERARCHY:**

Learn hierarchy all the way from pixels classifier one layer extracts features from output of previous layer, train all layers jointly.

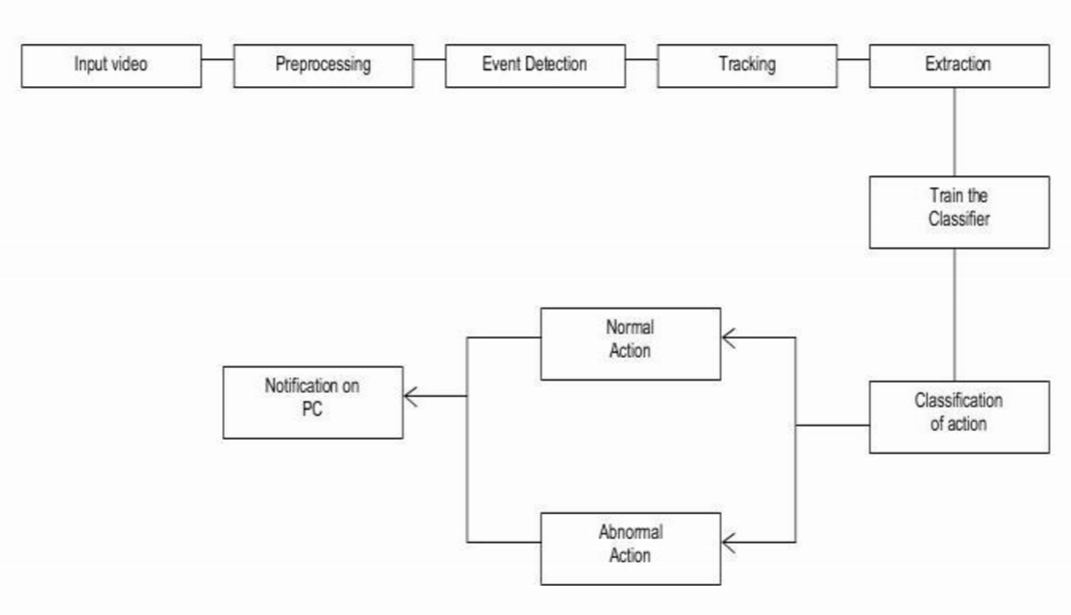
**Zero-One Loss**

The models given in these deep learning tutorials are largely used for classification. The major aim of training a classifier is to reduce the amount of errors (zero-one loss) on unseen examples.

**Negative Log**

Likelihood Loss Optimizing it for large models (thousands or millions of parameters) is prohibitively expensive (computationally) because the zero-one loss isn't differentiable. In order to achieve this maximization of the log-likelihood is done on the classifier given all the labels in a training set. The likelihood of the correct class and number of right predictions is not the equal, but they are pretty similar from the point of view of a randomly initialized classifier. As the likelihood and zero-one loss are different objectives but we should always see that they are co-related on the validation set but sometimes one will rise while the other falls, or vice-versa.

Flow Chart



Abnormal Action

Classification of action

Alert on System

Normal Action

Train the Classifier

Extraction

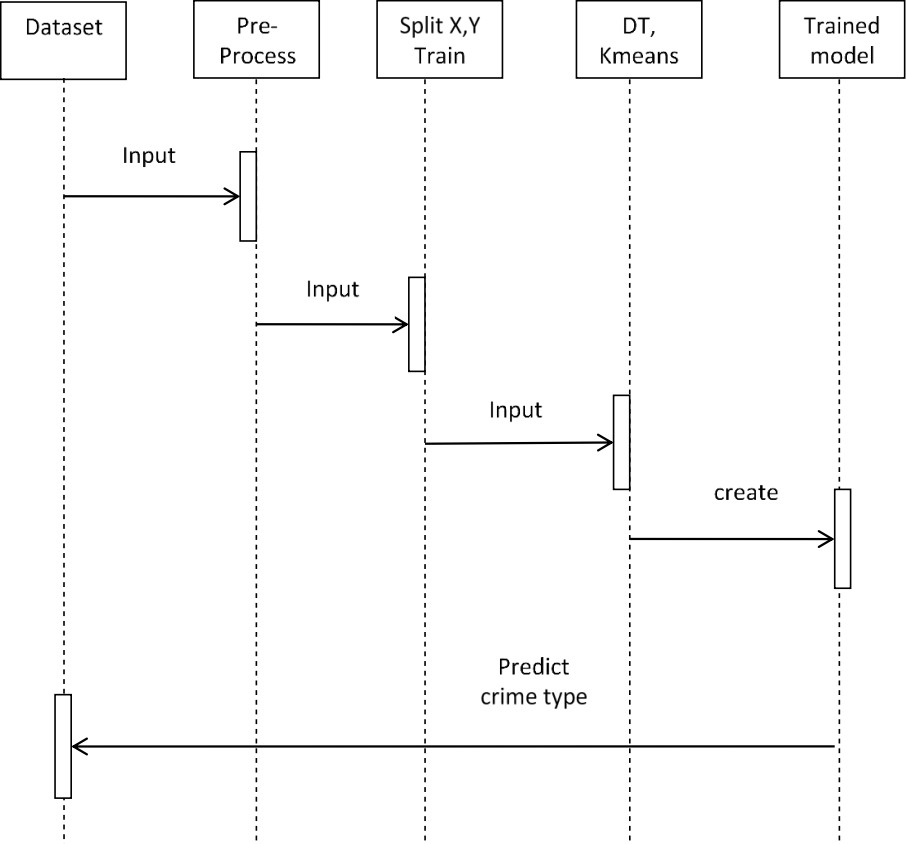
Tracking

Event Detection

Preprocessing

Input video

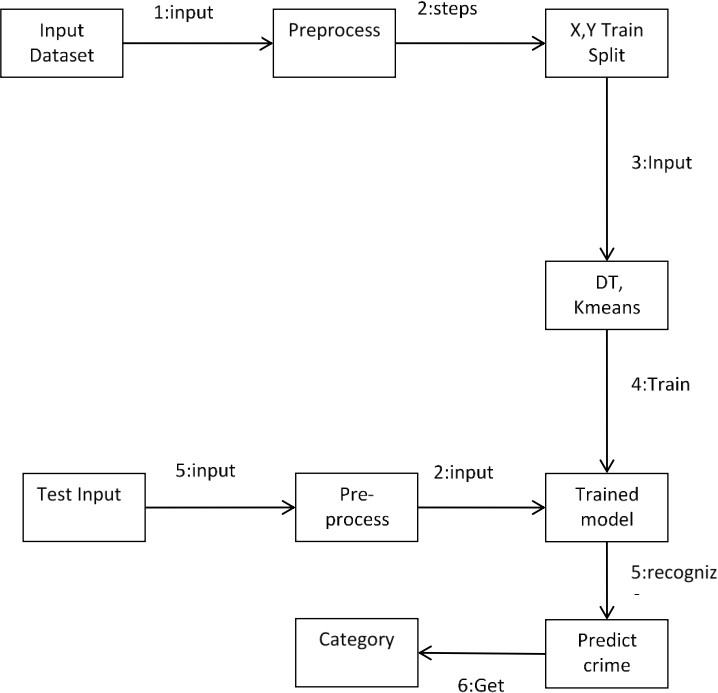
SEQUENCE DIAGRAM:



Predict abnormal activity type

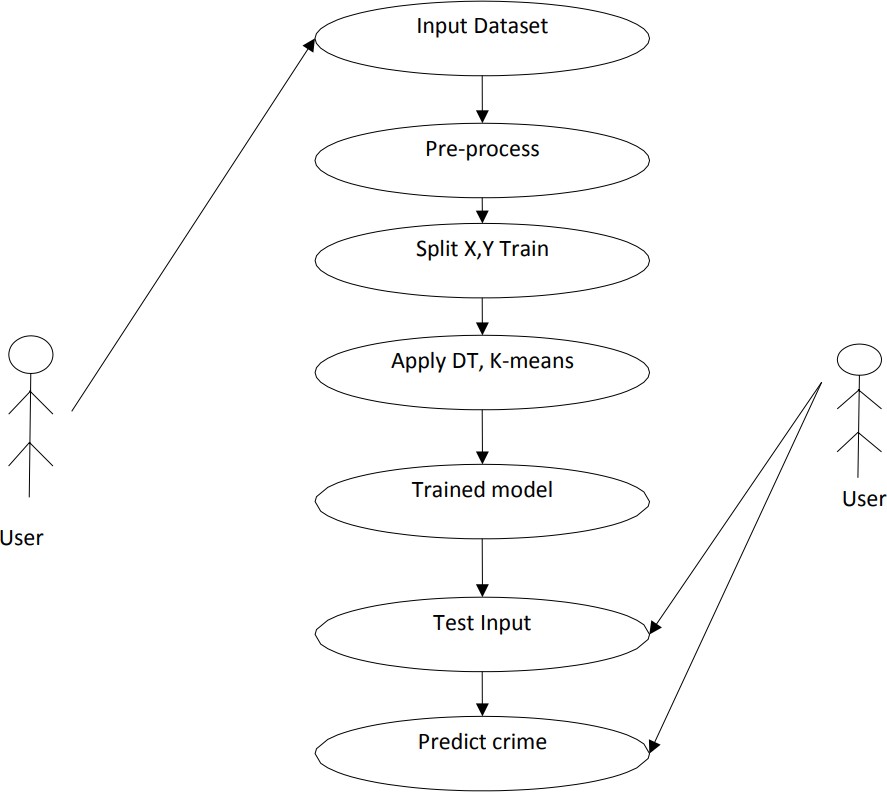
Auto encoder

COLLABORATION DIAGRAM:



Abnormal activity

Algorithm



abnormal activity

Algorithm

USE CASE DIAGRAM (ML Model):

# Technologies to be used

## Software Platform

* **VS Code**

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

* **Tensor flow**

Tensor flow is an open-source software library framework, which was used to implement object detection and recognition.

* **Keras**

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the Tensor Flow library.

* **OpenCV**

OpenCV is huge open-source library for the computer vision, machine learning, image processing and now it plays a major one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To identify image pattern and its various features we use vector space and perform mathematical operations on these features.

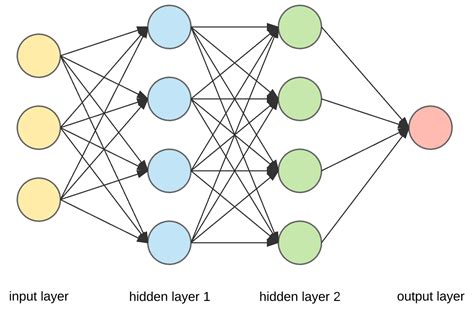
* **Other python libraries used**

Numpy – To process the image matrices

os – To access the file system to read the image from the train and test directory from our machines.

**3.6.4 Algorithm used**

Convolutional neural network (CNN) is a type of neural network for working with images, this type of neural network takes input from an image and extract features from an image and provide learnable parameters to efficiently do the classification, detection and a lot more tasks.



Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

* Convolutional layer
* Pooling layer
* Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

* **OpenCV: -** OpenCV is Open Source Computer vision. This library is capable of processing real-time frame and video while also boasting analytical capabilities. OpenCV is used to display text on the picture using putText ().

imshow () is used for the frame output.

The cv2.dnn.blobFromframe function returns a blob which is our input frame after mean subtraction, normalizing, and channel swapping.

Features of OpenCV Library:

Using OpenCV library, you can-

•Read and write frames

•Capture and save videos

•Process frames (filter, transform)

•Perform feature detection

•Detect specific objects such as faces, eyes, cars, in the videos or frames.

•Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc.

**Libraries in OpenCV**

**Numpy:**

NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines. Furthermore, NumPy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices. The implementation is even aiming at huge matrices and arrays. Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object Sophisticated (broadcasting) functions.
* Tools for integrating C/C++ and FORTRAN code.
* Useful linear algebra, Fourier Transform, and random number capabilities.

**Numpy Array:**

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

**SciPy:**

SciPy (Scientific Python) is often mentioned in the same breath with NumPy. SciPy extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier transformation and many others. NumPy is based on two earlier Python modules dealing with arrays. One of these is Numeric. Numeric is like NumPy a Python module for high-performance, numeric computing, but it is obsolete nowadays. Another predecessor of NumPy is Numarray, which is a complete rewrite of Numeric but is deprecated as well. NumPy is a merger of those two, i.e. it is built on the code of Numeric and the features of Numarray.

The Python Alternative to Matlab:

Python in combination with Numpy, Scipy and Matplotlib can be used as a replacement for MATLAB. The combination of NumPy, SciPy and Matplotlib is a free (meaning both "free" as in "free beer" and "free" as in "freedom") alternative to MATLAB. Even though MATLAB has a huge number of additional toolboxes available, NumPy has the advantage that Python is a more modern and complete programming language and - as we have said already before - is open source. SciPy adds even more MATLAB-like functionalities to Python. Python is rounded out in the direction of MATLAB with the module Matplotlib, which provides MATLAB-like plotting functionality.

* **Convolutional Neural Network: -** A Convolutional Neural Network is a deep neural network (DNN) widely used for the purposes of frame recognition and processing and NLP. Also known as a ConvNet, a CNN has input and output layers, and multiple hidden layers, many of which are convolutional. In a way, CNNs are regularized multilayer perceptions.

The convolutional neural network for this python project has 3 convolutional layers:

Convolutional layer; 96 nodes, kernel size 7

Convolutional layer; 256 nodes, kernel size 5

Convolutional layer; 384 nodes, kernel size 3

It has 2 fully connected layers, each with 512 nodes, and a final output layer of softmax type.

1. A first fully connected layer that receives the output of the third convolutional layer and contains 512 neurons, followed by a ReLU and a dropout layer.

2. A second fully connected layer that receives the 512-dimensional output of the first fully connected layer and again contains 512 neurons, followed by a ReLU and a dropout layer.

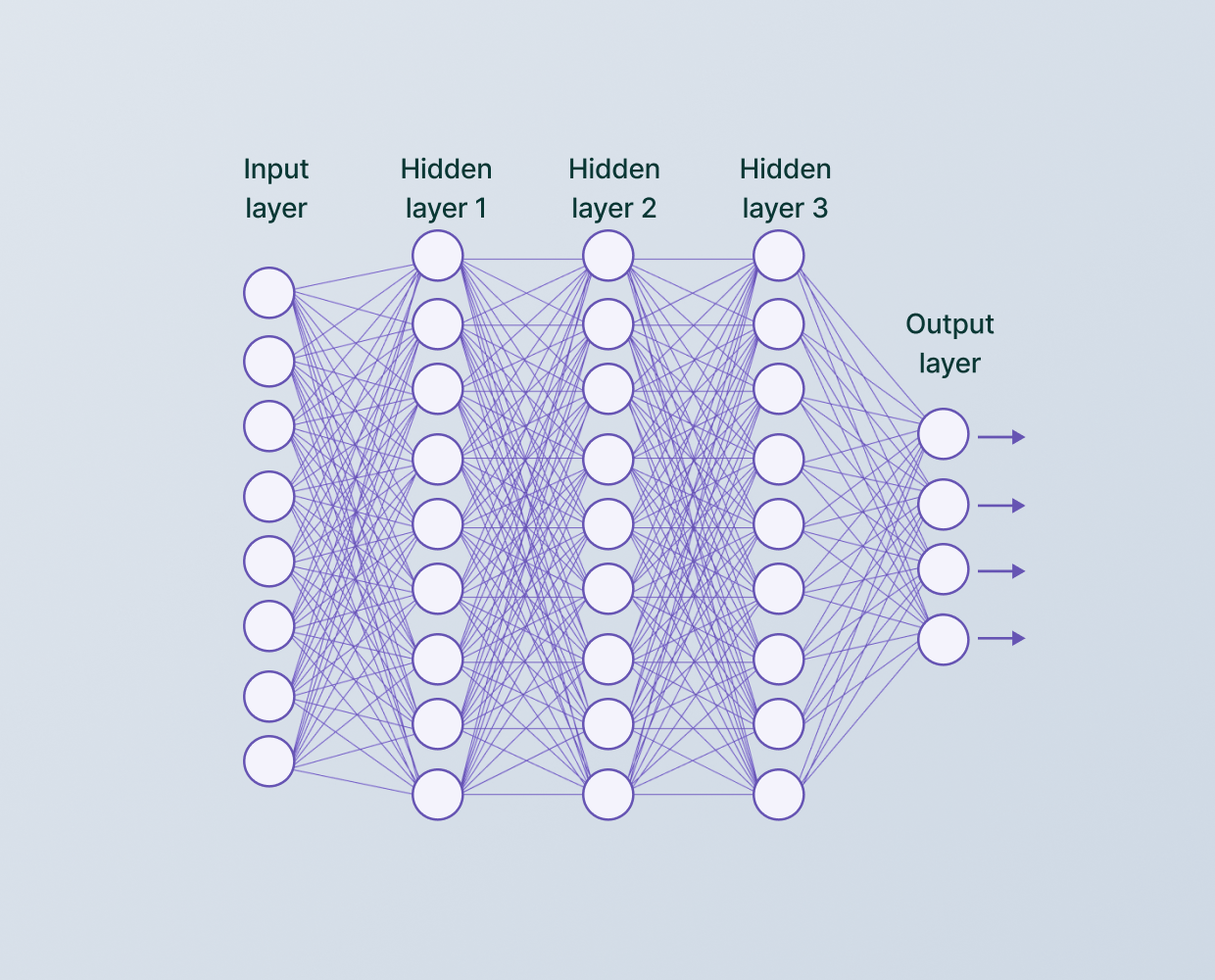
3. A third, fully connected layer which maps to the final classes for age or gender.

**CNN ARCHITECTURE:**

CNNs are feedforward networks in that information flow takes place in one direction only, from their inputs to their outputs. Just as artificial neural networks (ANN) are biologically inspired, so are CNNs. The visual cortex in the brain, which consists of alternating layers of simple and complex cells (Hubel & Wiesel, 1959, 1962), motivates their architecture.

CNN architectures come in several variations; however, in general, they consist of convolutional and pooling (or subsampling) layers, which are grouped into modules. Either one or more fully connected layers, as in a standard feedforward neural network, follow these modules. Modules are often stacked on top of each other to form a deep model. It illustrates typical CNN architecture for a toy image classification task. An image is input directly to the network, and this is followed by several stages of convolution and pooling. Thereafter, representations from these operations feed one or more fully connected layers.

Finally, the last fully connected layer outputs the class label. Despite this being the most popular base architecture found in the literature, several architecture changes have been proposed in recent years with the objective of improving image classification accuracy or reducing computation costs. Although for the remainder of this section, we merely fleetingly introduce standard CNN architecture.



* **Tensor Flow: - Tensor** Flow is an end-to-end open source platform for machine learning. Tensor Flow is a rich system for managing all aspects of a machine learning system. . Tensor Flow allows developers to create dataflow graphs-structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor. The single biggest benefit Tensor Flow provides for machine learning development is abstraction. Instead of dealing with the nitty-gritty details of implementing algorithms, or figuring out proper ways to hitch the output of one function to the input of another, the developer can focus on the overall logic of the application. Tensor Flow takes care of the details behind the scenes. For face detection, we have a .py file- this is a protobuf file (protocol buffer); it holds the graph definition and the trained weights of the model.Tensor flow is an open source software library for high performance numerical computation. It allows simple deployment of computation across a range of platforms (CPUs, GPUs, TPUs) due to its versatile design also from desktops to clusters of servers to mobile and edge devices. Tensor flow was designed and developed by researchers and engineers from the Google Brain team at intervals Google’s AI organization, it comes with robust support for machine learning and deep learning and the versatile numerical computation core is used across several alternative scientific domains. To construct, train and deploy Object Detection Models Tensor Flow is used that makes it easy and also it provides a collection of Detection Models pre-trained on the COCO dataset, the Kitti dataset, and the Open Images dataset. One among the numerous Detection Models is that the combination of Single Shot Detector (SSDs) and Mobile Nets architecture that is quick, efficient and doesn't need huge computational capability to accomplish the object Detection.

## Hardware Platform

**Table 1**

|  |  |
| --- | --- |
| Operating System | Windows, IOS, Linux |
| RAM | Minimum 12GB |
| Internal Storage | 12 GB |

## Tools, if any

* **Python:** Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale.Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.
* What can Python do?
* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.
* **Google Colab:** Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

What Colab Offers You?

As a programmer, you can perform the following using Google Colab.

* Write and execute code in Python
* Document your code that supports mathematical equations
* Create/Upload/Share notebooks
* Import/Save notebooks from/to Google Drive
* Import/Publish notebooks from GitHub
* Import external datasets e.g. from Kaggle
* Integrate PyTorch, Tensor Flow, Keras, OpenCV
* Free Cloud service with free GPU

# Advantages of this Project

The integration of CNN classification in a python code provides several benefits, including:

* Improved accuracy: CNN models are known for their high accuracy in image classification tasks. By integrating a CNN model in an Android app, the accuracy of image classification can be significantly improved, resulting in more reliable and accurate results.
* Real-time classification: CNN models can be optimized for real-time classification on Android devices. This means that the app can provide instant results, making it ideal for use cases such as object recognition in autonomous vehicles and facial recognition in security systems.
* Reduced server load: By deploying a CNN model on a python code, the python can perform image classification tasks locally on the device, reducing the load on servers. This can result in faster response times, reduced latency, and lower costs.
* Offline functionality: Since the CNN model is deployed on the python, the model can perform image classification tasks even without an internet connection. This makes the model more versatile and usable in areas with limited internet connectivity.
* User privacy: By performing image classification tasks locally on the device, the model can provide greater user privacy. The app does not need to transmit images to a server, reducing the risk of sensitive data being intercepted or compromised.

Overall, the integration of CNN classification in a python code provides several benefits, including improved accuracy, real-time classification, reduced server load, offline functionality, and user privacy. These benefits make CNN-based Android apps an attractive solution for a variety of image classification tasks in various industries such as healthcare, transportation, and security.

* Anomaly Detection improves surveillance efforts and helps hunt criminals and terrorists
* Anomaly Detection and Anomaly Recognition Technology is straightforward to integrate, and most solutions are compatible with the bulk of security software
* Anomaly Detection enables the automation of the identification process, which saves time and improves accuracy
* Improved Security, Anomaly Detection improves surveillance efforts and helps track and down criminals and Terrorists. Personal Security is also enhanced since there is nothing for hackers to steal or change, such as passwords.
* Anomaly Detection allows the identification process to be automated, thus saving time and increasing Accuracy.
* Easy to Integrate Face Detection and Face Recognition Technology is easy to Integrate, and Most Solutions are Compatible with the Majority of Security Software.

# Assumptions, if any

* Is dataset clean?
* Is dataset properly pre-processed?
* Proper internet connection with enough speed
* Python modules and libraries are install properly with specific versions

# Future Scope and further enhancement of the Project

* Some future improvements can be made to this project. To improve the accuracy, a dataset with more number of images can be given.
* The algorithm can be improved to an extent where it can be able to process and give the desired result on images with disturbances like lack of clarity, a different angle, or any kind of accessories on the face. For example, if the person is wearing sunglasses, or has makeup on the face, the algorithm will be able to give the desired result.
* We will look into a more complex CNN architecture and a more reliable image processing approach for estimating exact ages for future work. We can use this project for electronic customers, crowd behavior analysis.
* The future of anomaly recognition technology is bright. Security and surveillances are the major segments which will be deeply influenced. In the long run, robots using anomaly recognition technology may also come to foray. They can be helpful in completing the tasks that are impractical or difficult for human beings to complete.
* A facial recognition system is used to identify and verify a person from an image or video source. It uses biometric software’s along with AI enabled devices for mapping facial features and brings out the recognition step.

# Project Repository Location

| **S#** | **Project Artifacts (softcopy)** | **Location** | **Verified by Project Guide** | **Verified by Lab In-Charge** |
| --- | --- | --- | --- | --- |
|  | Project Synopsis Report (Final Version) | 3231 |  |  |
|  | Project Progress updates | 3231 |  |  |
|  | Project Requirement specifications | 3231 |  |  |
|  | Project Report (Final Version) | 3231 |  |  |
|  | Test Repository | 3231 |  |  |
|  | Project Source Code (final version) with executable | 3231 |  |  |
|  | Any other document | 3231 |  |  |

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| CV | Computer Vision |
| CNN | Convolutional Neural Network |
| FCN | Fully Convolutional Networks |
| SGD | Stochastic Gradient Descent |
| CoGAN | Coupled Generative Adversarial Network |
| BLOB | Binary Large Object |
| SciPy | Scientific Python |
| SVM | Support Vector Machine |
| ANN | Artificial Neural Networks |

# Conclusion

The proposed system aims to open a new door in the field of video surveillance and provide the result on the basis to detect abnormal activities. It will help the user to monitor any abnormal activities or suspicious events. It's been very difficult to monitor abnormal activities in various fields like security, crime prevention, traffic monitoring. It will help the user by sending an alert message when an abnormal condition is identified. The number of parameters currently included is an attempt to cover all the basic aspects of video surveillance and other overlooked parameters which deserve recognition. This work is interested in the recognition of abnormal human activities by providing a brief analysis of the recent research tasks in this field of video surveillance. We have implemented the CNN to detect abnormal activities. Finally, through this analysis of the recent research tasks in this field of video surveillance and provide the result on the basis to detect abnormal activities.

# References

<https://www.ijeat.org/wp-content/uploads/papers/v11i2/B32751211221.pdf>

<https://www.aibridgeml.ai/blog/face-detection-and-its-benefits>

<https://arxiv.org/pdf/2010.03791.pdf>

<https://www.mathworks.com/discovery/deep-learning.html>

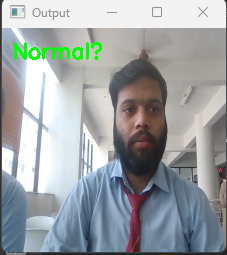
<https://sisu.ut.ee/imageprocessing/book/1>

<https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm>

<https://www.w3schools.com/python/>

**Annexure A**

**Screen Shots**

**

