ABSTRACT

Suspicious Activity Detection using Convolutional Neural Networks (CNNs) is an innovative approach aimed at enhancing security and surveillance systems by automatically identifying and flagging suspicious behaviors in real-time. This paper presents a deep learning-based model leveraging the power of CNNs to detect suspicious activities from video feeds, thereby providing a robust solution for modern security challenges. The proposed system employs CNNs to analyse visual data and identify patterns associated With suspicious activities, such as unauthorized access, loitering, or aggressive behavior. By training the model on a diverse dataset containing various normal and suspicious activities, the system learns to distinguish between routine and potentially dangerous behaviors accurately.

Key contributions of this work include the development of an efficient CNN architecture optimized for real-time processing and high accuracy in detection. The model is evaluated using extensive video datasets from public surveillance systems, demonstrating its effectiveness in various real-world scenarios. Additionally, the system incorporates advanced preprocessing techniques to handle different lighting conditions, camera angles, and environmental noise, ensuring reliable performance across various settings.

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The Suspicious Activity Detection System aims to enhance security in surveillance environments by utilizing machine learning to automatically identify suspicious behaviors in real-time. to automatically identify suspicious behaviors such as unauthorized access, loitering, face covering, and aggressive actions in real-time. Leveraging **Convolutional Neural Networks (CNNs)**, the system analyzes video footage by breaking it into frames and classifying behaviors as normal or suspicious based on predefined patterns. It adapts to various environmental factors, including camera angles and lighting, making it suitable for diverse surveillance settings. Integrating with existing CCTV infrastructure, it provides security personnel with a user-friendly dashboard for monitoring, reporting, and flagging threats. With robust preprocessing to minimize false positives, it ensures secure data handling in compliance with privacy regulations. The system improves surveillance effectiveness, reduces human error, and promotes safer environments..

1.2 PROJECT PURPOSE

The purpose of the **Suspicious Activity Detection System** is to enhance surveillance security by automating threat detection. Traditional methods relying on human operators can lead to fatigue and oversight. This system uses **Convolutional Neural Networks (CNNs)** to identify suspicious behaviors like unauthorized access, loitering, face covering, and aggression in real-time, improving efficiency and response times.

By automating detection, the system reduces human error, improves response times, and ensures continuous monitoring in complex environments. It integrates with existing infrastructure, equipping security personnel to better monitor, report, and address potential threats. Ultimately, the **Suspicious Activity Detection System** promotes safer environments with a reliable, scalable solution for modern security needs..

1.3 PROJECT FEATURES

The Suspicious Activity Detection System offers several key features aimed at enhancing security. It automatically detects suspicious behaviors, such as unauthorized access, loitering, face covering, and aggression, using Convolutional Neural Networks (CNNs). The system processes video footage in real-time, breaking it down into individual frames for detailed analysis, and is adaptable to various environmental factors like lighting and camera angles. Seamlessly integrating with existing CCTV infrastructure, it provides a user-friendly dashboard for security personnel to monitor, report, and flag potential threats. Advanced preprocessing techniques reduce false positives and negatives, ensuring accuracy. Additionally, the system ensures secure data handling to comply with privacy regulations and is scalable to meet growing surveillance needs.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 LITERATURE SURVEY

The literature on Composite Behavioral Modeling for Identity Theft Detection highlights the limitations of traditional surveillance systems relying on manual observation and basic algorithms. It proposes using CNNs for real-time video analysis to detect suspicious behaviors like unauthorized access. While CNNs offer improved accuracy, they face challenges like high computational demands and the need for large datasets.

Future advancements could focus on integrating real-time analytics, adapting machine learning models to evolving fraud tactics, and improving privacy and bias issues in AI Systems.

2.2 PROBLEM STATEMENT

An automated system using Convolutional Neural Networks (CNN) to analyze video footage, classify suspicious activities, enhancing security without the need for constant human oversight. It helps in Differentiating between normal and suspicious human activities.

2.3 EXISTING SYSTEM

Existing suspicious activity detection systems rely on manual monitoring and basic algorithms like motion detection and object tracking, which are error-prone. Variations in lighting, camera angles, and background clutter result in high false positive and negative rates, reducing their efficiency.

Moreover, they lack machine learning capabilities, preventing adaptation to evolving threats. As a result, these systems are limited in scalability and consistency, highlighting the need for more advanced, automated solutions for reliable and efficient surveillance.

2.4 DISADVANTAGES OF EXISTING SYSTEM

1. **Manual Monitoring**: Labor-intensive and prone to human error.
2. **Rigid Rule-Based Methods**: Inflexible and unable to adapt to new behaviors.
3. **Limited Scalability**: Ineffective in complex, dynamic environment.

2.5 PROPOSED SYSTEM

The proposed system for suspicious activity detection uses **Convolutional Neural Networks (CNNs)** to enhance the efficiency and accuracy of surveillance. Unlike traditional systems that depend on manual monitoring and basic computer vision techniques, this system automates the detection process by leveraging deep learning models. CNNs allow the system to analyze video feeds in real-time, significantly reducing the need for human intervention and improving response times.

The system detects various suspicious activities like unauthorized access, loitering, and aggressive behavior by learning from a comprehensive dataset of normal and suspicious behaviors. CNNs extract and analyze features from video frames, enhancing detection accuracy and identifying subtle anomalies effectively.

One of the key advantages of the proposed system is its ability to handle diverse environments, including varying lighting conditions and camera angles, through advanced preprocessing. This ensures consistent performance and reduces false positives and negatives. The system is also scalable and can be updated with new data, allowing it to adapt to evolving security threats and maintain its effectiveness over time.

2.6 ADVANTAGES OF PROPOSED SYSTEM

1. Improved Accuracy.
2. Real-Time Processing.
3. Reduced False Positives and Negatives.
4. Automation of Detection.
5. Scalability.

3. SYSTEM STUDY

3. SYSTEM STUDY

3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

3.1.1 ECONOMICAL FEASIBILITY

This system is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.1.2 TECHNICAL FEASIBILITY

This system is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.1.3 OPERATIONAL FEASIBILITY

This system is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. SYSTEM REQUIREMENTS

4. SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

• Operating System	: Windows 10 or above
• Coding Language	: Python
• Frame Work	: Tkinter
• Back End	: Python
• Data Base	: MY SQL

4.2 HARDWARE REQUIREMENTS

• Processor	**:** intel-i5
• Hard Disk	**: 512GB** or above
• Monitor	**:** SVGA
• Mouse	**:** Optical Mouse
• RAM	**:** 8 GB or Above

5. SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 INTRODUCTION

Architecture defines the components, modules interfaces and data for a system to satisfy specified requirements. One should see as the applications of the systems theory to product development.

5.2 ARCHITECTURE

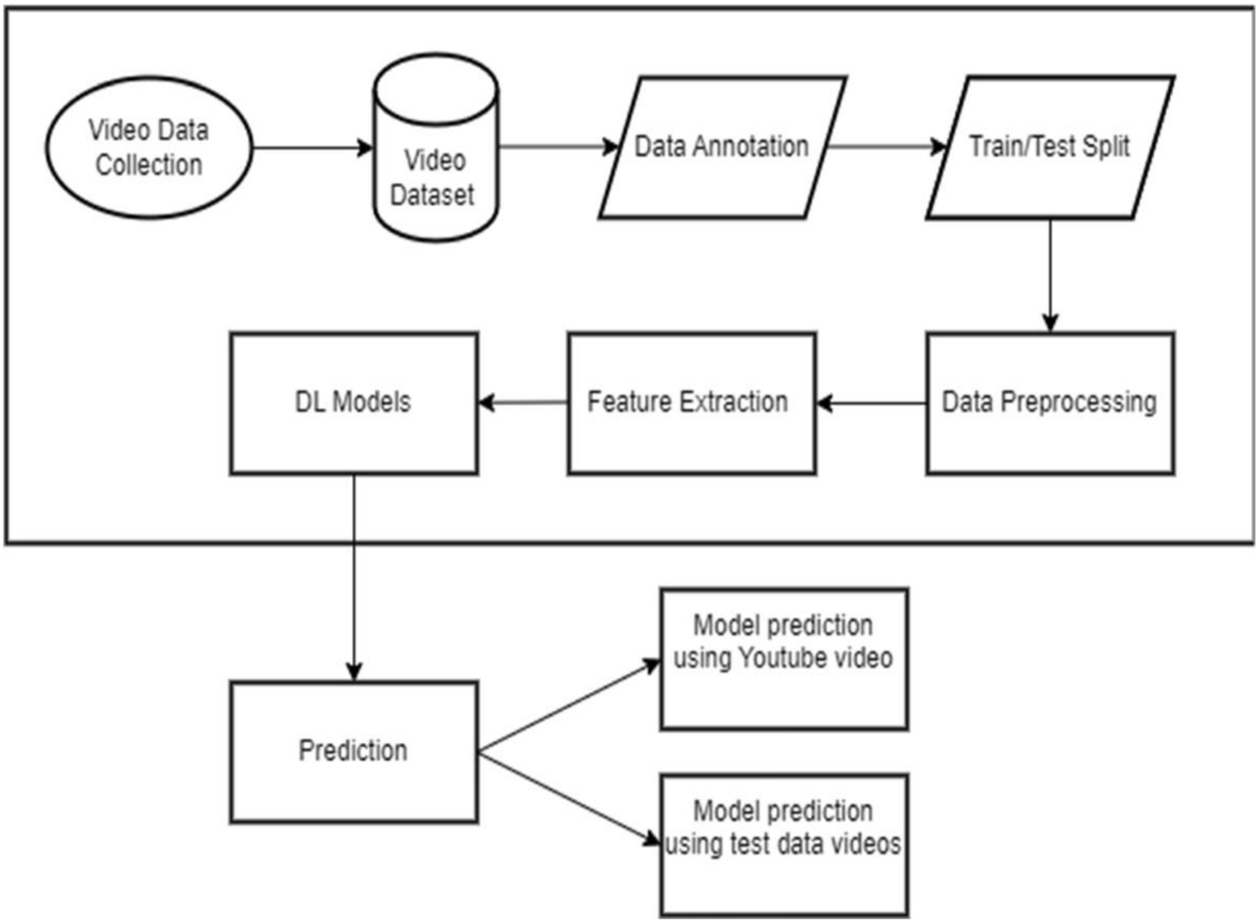


FIGURE 5.1 PROPOSED ARCHITECTURE

Video Data Collection: This initial step gathers video footage for analysis, potentially from multiple sources (e.g., CCTV cameras).

Video Dataset: Collected video data is stored in a dataset, which serves as the source for training and testing.

Data Annotation: Each video is annotated, likely labeling suspicious and non-suspicious behaviors, to create a supervised learning dataset.

Train/Test Split: The dataset is split into training and testing sets to evaluate model performance.

Data Preprocessing: This step prepares the data for feature extraction, including operations like frame resizing, normalization, and possibly data augmentation.

Feature Extraction: Important features relevant to suspicious activity are extracted. This step likely uses techniques to highlight critical spatial and temporal aspects of behavior in the video frames.

DL Models (Deep Learning Models):The core model, which could be a Convolutional Neural Network (CNN) or a similar deep learning model, is trained on the processed data and features.

Prediction:The trained model generates predictions on new data, classifying activities as suspicious or non-suspicious.

Model Prediction Using YouTube/Test Data Videos: The architecture includes provisions for testing the model on videos from external sources (like YouTube) or additional test data, to validate its generalizability and robustness.

5.3 UNIFIED MODELING LANGUAGE (UML)

UML represents Unified Modeling Language. UML is an institutionalized universally useful showing dialect in the subject of article situated programming designing. The fashionable is overseen, and become made by way of, the Object Management Group.

The goal is for UML to become a regular dialect for making fashions of item arranged PC programming. In its gift frame UML is contained two noteworthy components: a Meta-show and documentation. Later on, a few type of method or system can also likewise be brought to; or related with, UML.

The Unified Modeling Language is a popular dialect for indicating, Visualization, Constructing and archiving the curios of programming framework, and for business demonstrating and different non-programming frameworks.

The UML speaks to an accumulation of first-rate building practices which have verified fruitful in the showing of full-size and complicated frameworks.

The UML is a essential piece of creating gadgets located programming and the product development method. The UML makes use of commonly graphical documentations to specific the plan of programming ventures.

GOALS OF UML:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.

5.3.1 USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

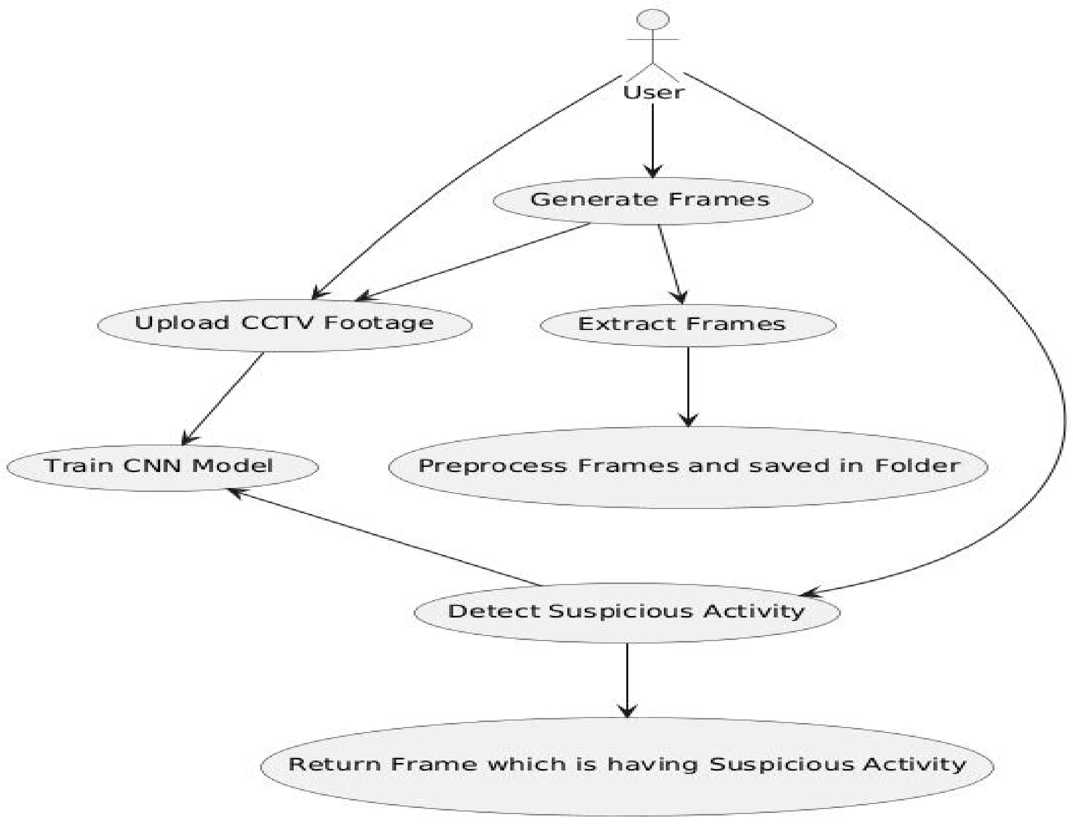

User
Generate Frames
Upload CCTV Footage
Extract Frames
Train CNN Model
Preprocess Frames and saved in Folder
Detect Suspicious Activity
Return Frame which is having Suspicious Activity

FIGURE 5.2 USE CASE DIAGRAM

5.3.4 FLOWCHART DIAGRAM:

A flowchart diagram is a visual tool that shows the steps in a process or workflow. It uses shapes like arrows and rectangles to represent different actions and decisions. Flowcharts help simplify complex tasks and make it easier to understand how things work. They are useful for planning and identifying areas for improvement. Overall, flowcharts help communicate processes clearly to others.

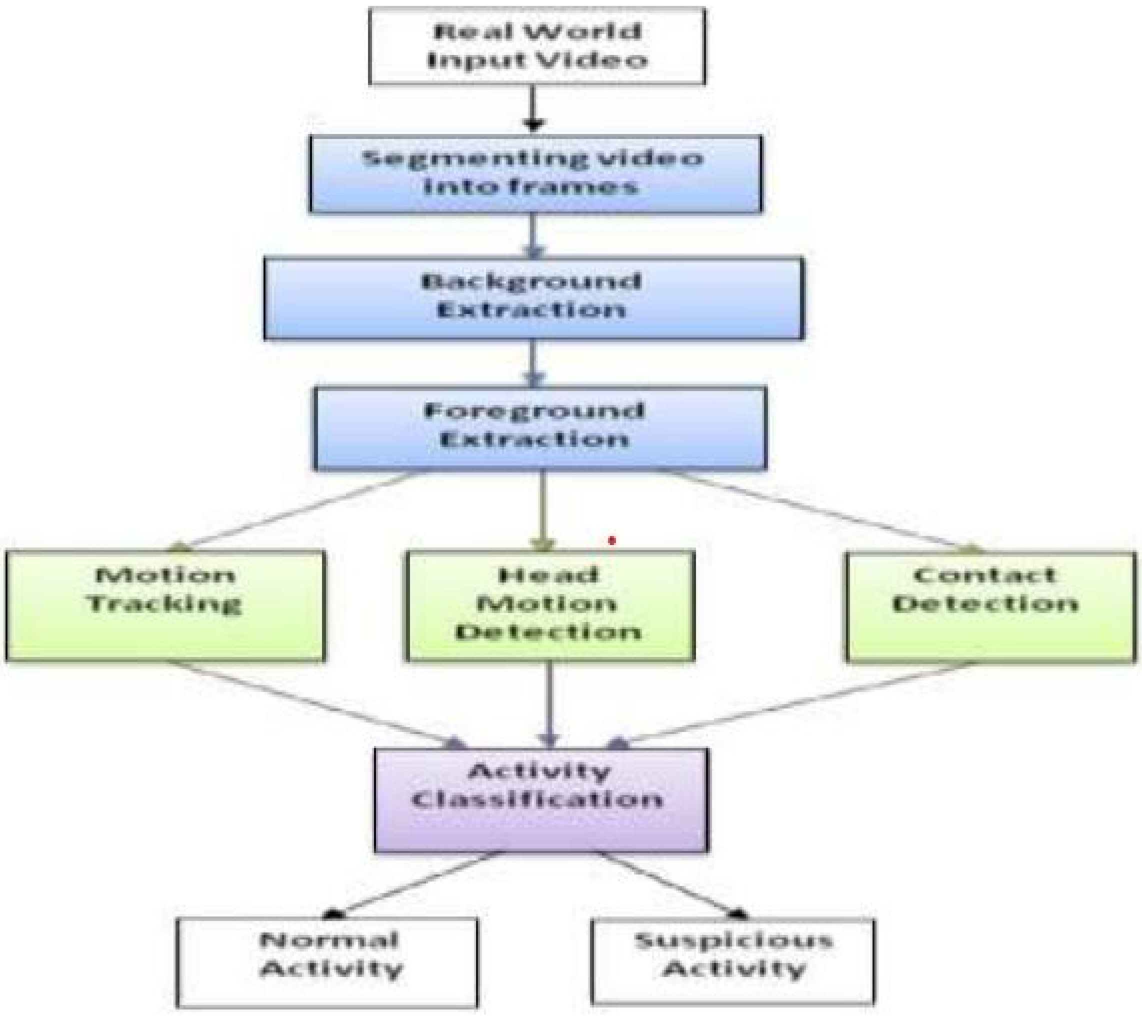


FIGURE:5.5 **FLOWCHART DIAGRAM**

5.3.5 DATAFLOW DIAGRAM:

A data flow diagram (DFD) is a simple drawing that shows how information moves in a system. It uses arrows to show where data goes, circles to represent processes that use the data, and rectangles for places that store data. DFDs help people see how different parts of a system work together and how data is shared. They are helpful for planning and improving systems by making it clear what data comes in and goes out of each process. Overall, DFDs make it easy to understand how data is handled in a system.

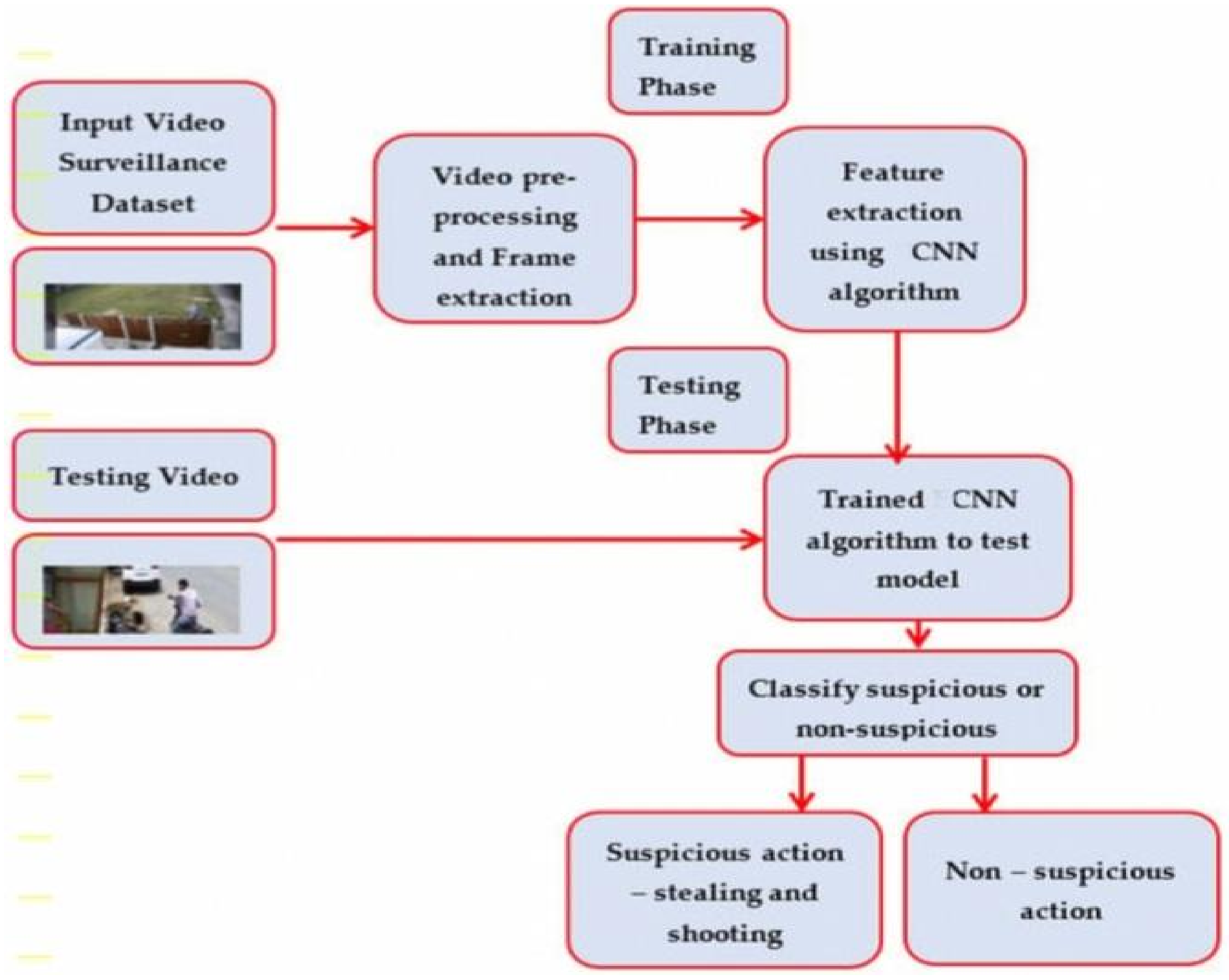

Training Phase
Input Video Surveillance Dataset
Video pre-processing and Frame extraction
Feature extraction using CNN algorithm
Testing Phase
Testing Video
Trained CNN algorithm to test model
Classify suspicious or non-suspicious
Suspicious action – stealing and shooting
Non – suspicious action

FIGURE:5.6 DATAFLOW DIAGRAM

6. IMPLEMENTATION

6. IMPLEMENTATION

6.1 MODULES

1. Upload CCTV Footage.

2. Generate Frames.

3. Detect Suspicious Activity.

6.2 MODULES DESCRIPTION

1. Upload CCTV Footage

This module allows users to upload video files collected from surveillance cameras and other relevant sensors. It serves as the initial step in the process, where users can upload their own videos for analysis. Note that the uploaded videos should feature individuals covering their faces or engaging in shoplifting activities, similar to the examples used in this project.

2. Generate Frames:

In this module, the uploaded video is processed to extract individual frames, which are then saved in a designated 'frames' folder. Each extracted frame is numbered for easy identification and access.

3. Detect Suspicious Activity:

This module initiates the monitoring of the extracted frames for suspicious activities. It utilizes Convolutional Neural Networks (CNNs) to automatically extract spatial and temporal features from the frames, identifying significant patterns indicative of suspicious behaviors. Detected behaviors are classified as normal or suspicious based on learned patterns. If any suspicious activity is detected, the right text area will display details of the specific frame containing such activity.

6.3 SOURCE CODE :

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from imutils import paths
import matplotlib.pyplot as plt
import datetime
from tkinter.filedialog import askopenfilename
import cv2
import shutil
import os
from imageai.Prediction.Custom import CustomImagePrediction
import os

main = tkinter.Tk()
main.title("Suspicious Activity Detection")
main.geometry("1200x1200")

global filename

execution_path = os.getcwd()
prediction = CustomImagePrediction()
prediction.setModelTypeAsResNet()
prediction.setModelPath("model.h5")
prediction.setJsonPath("model_class.json")
prediction.loadModel(num_objects=2)

def upload():
  global filename
  filename = askopenfilename(initialdir = "videos")
  pathlabel.config(text=filename)

def generateFrame():
  global filename
  text.delete('1.0', END)
  if not os.path.exists('frames'):
    os.mkdir('frames')
  else:
    shutil.rmtree('frames')
    os.mkdir('frames')
  vidObj = cv2.VideoCapture(filename)
  count = 0
  success = 1
  while success:
    success, image = vidObj.read()
    if count < 500:
      cv2.imwrite("frames/frame%d.jpg" % count, image)
```

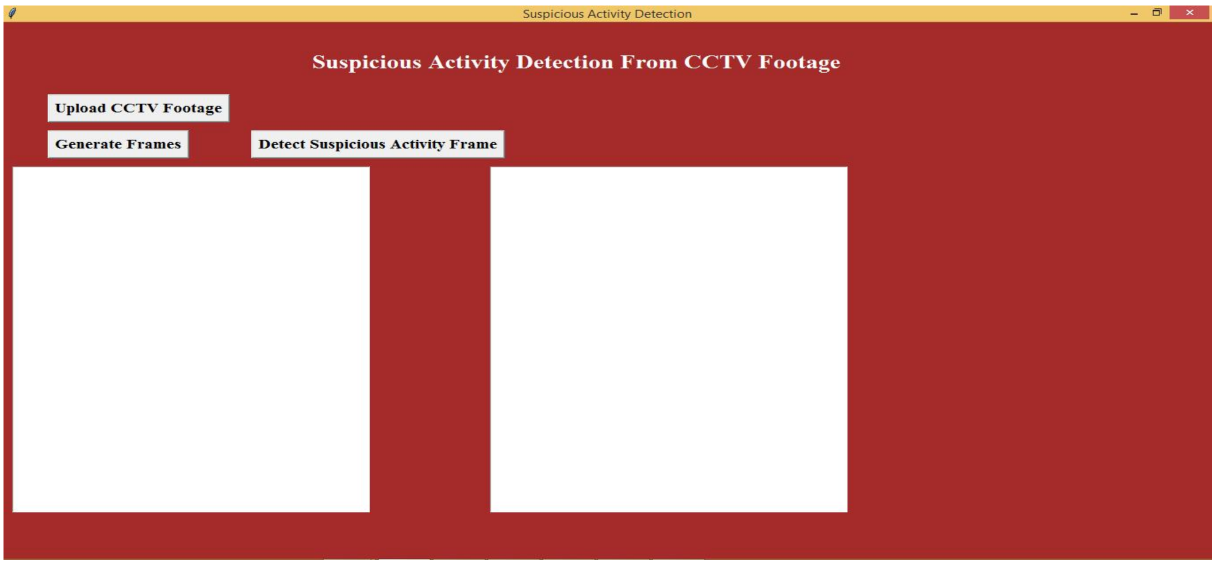
```
 text.insert(END,"frames/frame."+str(count)+" saved\n")
        print("frames/frame."+str(count)+" saved")
        #pathlabel.config(text="frames/frame."+str(count)+" saved")
      else:
        break
      count += 1
  pathlabel.config(text="Frame generation process completed. All frames saved inside
frame folder")

def detectActivity():
  imagePaths = sorted(list(paths.list_images("frames")))
  count = 0
  option = 0;
  text1.delete('1.0', END)
  for imagePath in imagePaths:
    predictions, probabilities = prediction.predictImage(imagePath, result_count=1)
    for eachPrediction, eachProbability in zip(predictions, probabilities):
      if float(eachProbability) > 80:
        count = count + 1;
      if float(eachProbability) < 80:
        count = 0
      if count > 10:
        option = 1
        print(imagePath+" is predicted as "+eachPrediction+" with probability : "
+str(eachProbability))
        text1.insert(END,imagePath+" is predicted as "+eachPrediction+" with probability :
" +str(eachProbability)+"\n\n")
        count = 0;
    print(imagePath+" processed")
  if option == 0:
    text1.insert(END,"No suspicious activity found in given footage")

font = ('times', 20, 'bold')
title = Label(main, text='Suspicious Activity Detection From CCTV Footage')
title.config(bg='brown', fg='white')
title.config(font=font)
title.config(height=3, width=80)
title.place(x=5,y=5)

font1 = ('times', 14, 'bold')
upload = Button(main, text="Upload CCTV Footage", command=upload)
upload.place(x=50,y=100)
upload.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='brown', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=300,y=100)
```

```
depthbutton = Button(main, text="Generate Frames", command=generateFrame)
depthbutton.place(x=50,y=150)
depthbutton.config(font=font1)

userinterest = Button(main, text="Detect Suspicious Activity Frame",
command=detectActivity)
userinterest.place(x=280,y=150)
userinterest.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=25,width=50)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=200)
text.config(font=font1)

text1=Text(main,height=25,width=50)
scroll=Scrollbar(text1)
text1.configure(yscrollcommand=scroll.set)
text1.place(x=550,y=200)
text1.config(font=font1)

main.config(bg='brown')
main.mainloop()
```

7. SCREENSHOTS

7. SCREENSHOTS

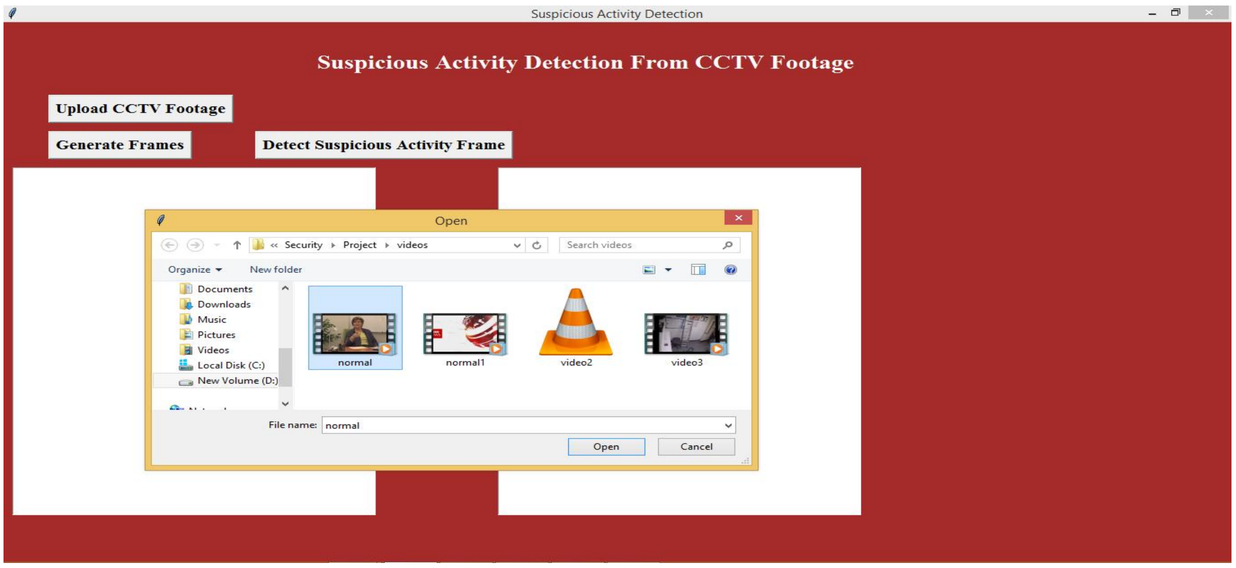
7.1 HOME PAGE:


Suspicious Activity Detection
Suspicious Activity Detection From CCTV Footage
Upload CCTV Footage
Generate Frames
Detect Suspicious Activity Frame

Screenshot 7.1: Home page

in above screen click on 'upload CCTV Footage" button to upload dataset and to get below screen.

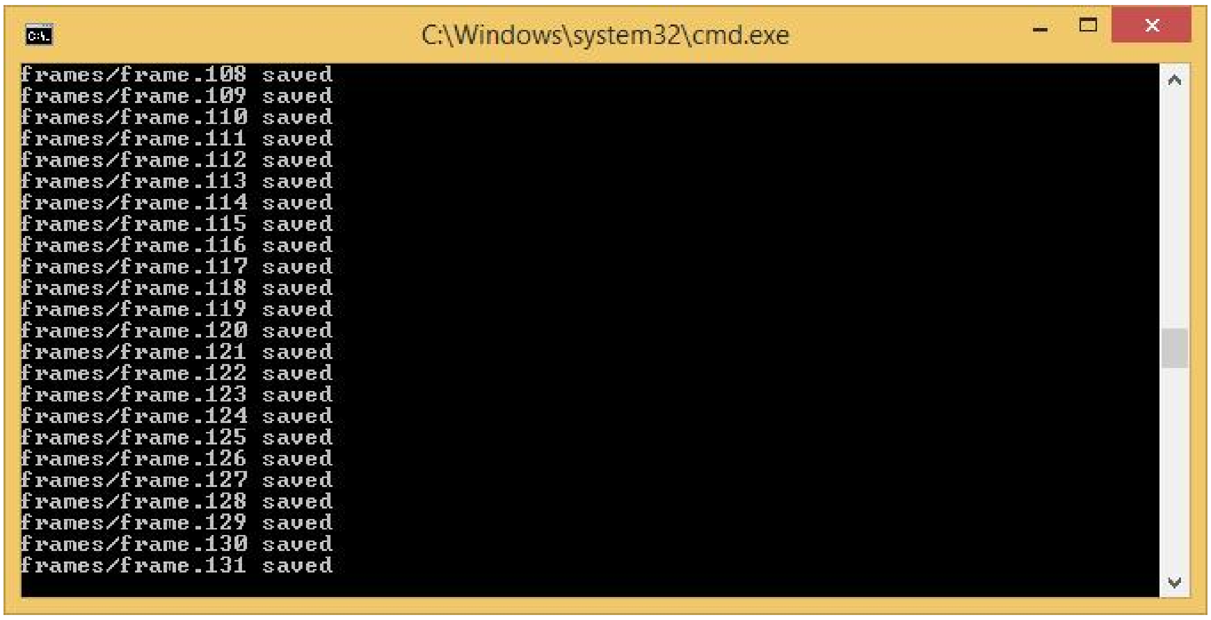
7.2 UPLOAD CCTV Footage:



Screenshot 7.2: UPLOAD CCTV Footage

In above screen uploading 'video' click on 'open' button to load video and after uploading video click on 'Generate Frames' button to generate frame

7.3 Saving the frames from Video:

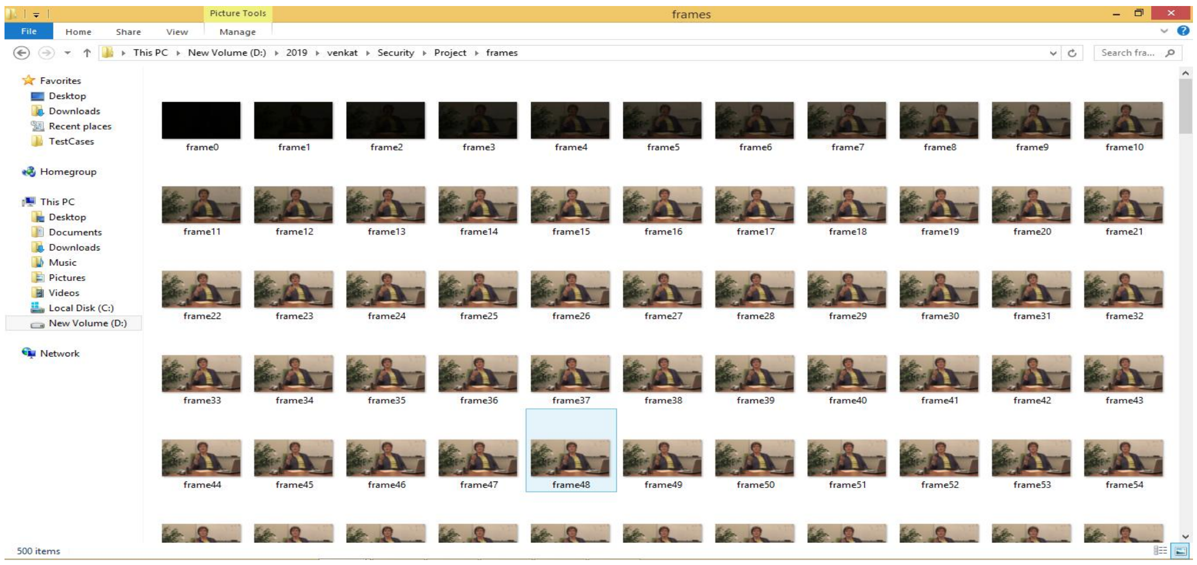


```
frames/frame.108 saved
frames/frame.109 saved
frames/frame.110 saved
frames/frame.111 saved
frames/frame.112 saved
frames/frame.113 saved
frames/frame.114 saved
frames/frame.115 saved
frames/frame.116 saved
frames/frame.117 saved
frames/frame.118 saved
frames/frame.119 saved
frames/frame.120 saved
frames/frame.121 saved
frames/frame.122 saved
frames/frame.123 saved
frames/frame.124 saved
frames/frame.125 saved
frames/frame.126 saved
frames/frame.127 saved
frames/frame.128 saved
frames/frame.129 saved
frames/frame.130 saved
frames/frame.131 saved
```

Screenshot 7.3: Saving the frames from Video

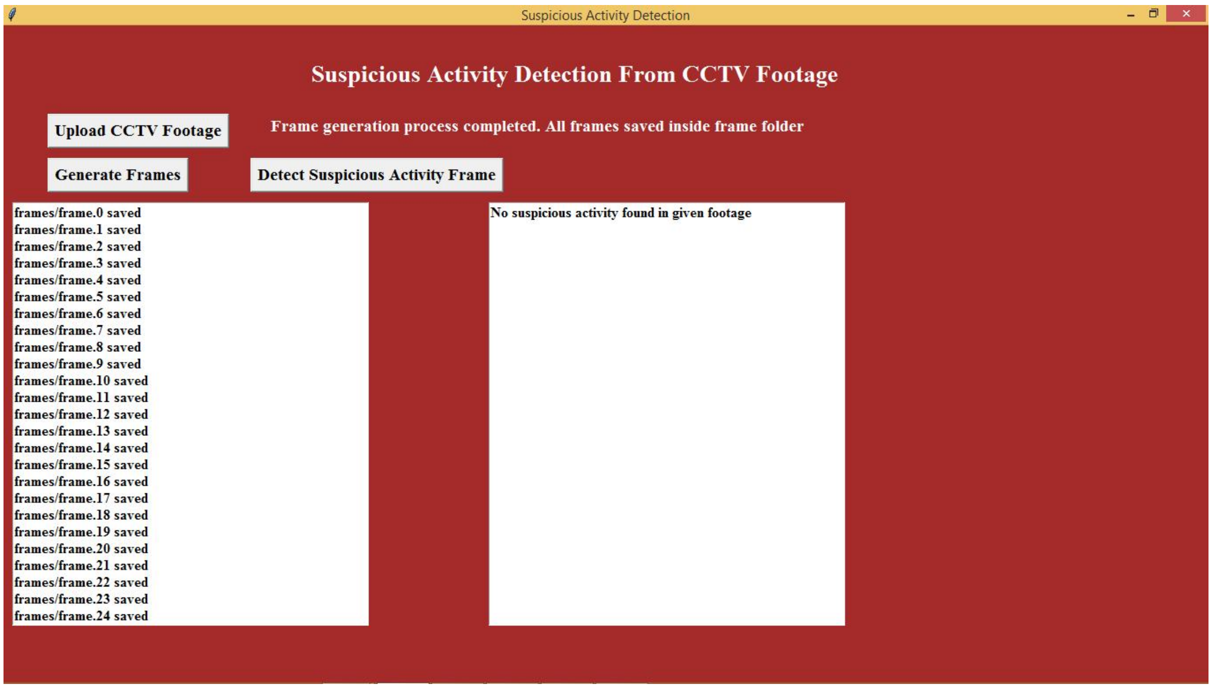
In above figure we can see extracted frames are saving inside 'frames' folder frame no. Now we see frames folder below which has images from video

7.4 Frame Mining:



Screenshot 7.4: above figure shows all images from video extracted. After frame extraction will get below screen

7.5 Frame saving:


Suspicious Activity Detection
Suspicious Activity Detection From CCTV Footage
Upload CCTV Footage
Generate Frames
Detect Suspicious Activity Frame

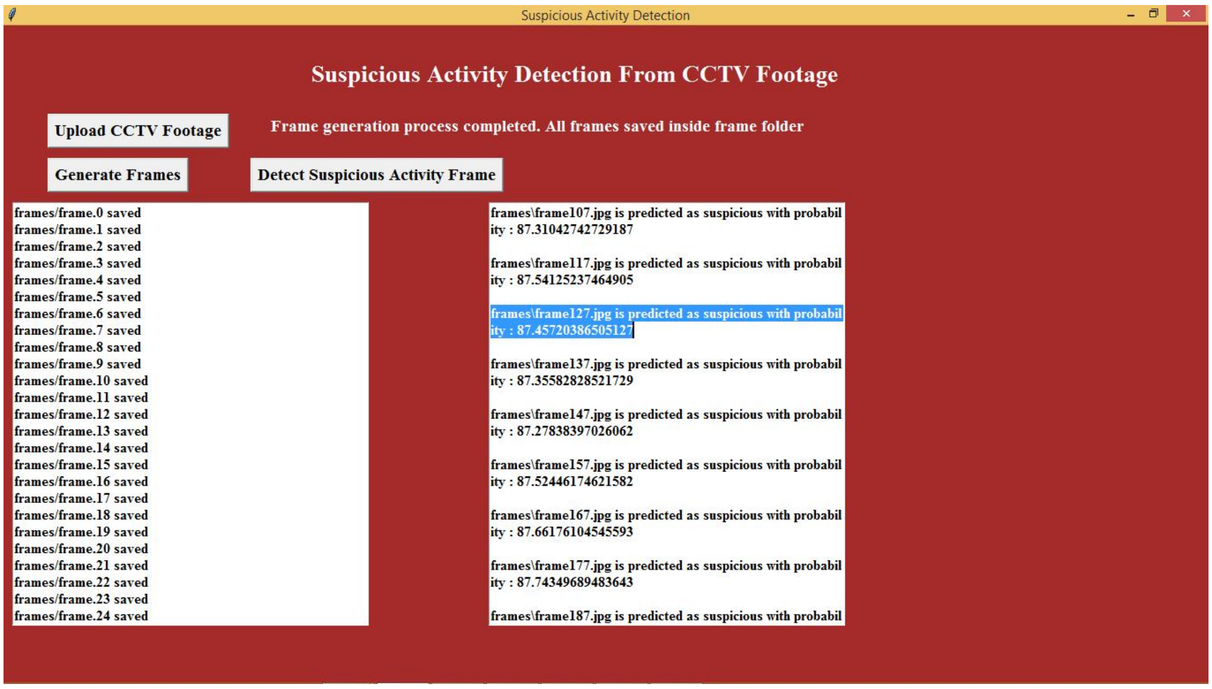
Frame generation process completed. All frames saved inside frame folder

```
frames/frame.0 saved
frames/frame.1 saved
frames/frame.2 saved
frames/frame.3 saved
frames/frame.4 saved
frames/frame.5 saved
frames/frame.6 saved
frames/frame.7 saved
frames/frame.8 saved
frames/frame.9 saved
frames/frame.10 saved
frames/frame.11 saved
frames/frame.12 saved
frames/frame.13 saved
frames/frame.14 saved
frames/frame.15 saved
frames/frame.16 saved
frames/frame.17 saved
frames/frame.18 saved
frames/frame.19 saved
frames/frame.20 saved
frames/frame.21 saved
frames/frame.22 saved
frames/frame.23 saved
frames/frame.24 saved
```

```
No suspicious activity found in given footage
```

Screenshot 7.5: Now click on 'Detect Suspicious Activity Frame' button to start monitoring frames for suspicious activity

7.5 Frames which are having Suspicious Activity:


Suspicious Activity Detection
Suspicious Activity Detection From CCTV Footage
Upload CCTV Footage
Generate Frames
Detect Suspicious Activity Frame

Frame generation process completed. All frames saved inside frame folder

```
frames/frame.0 saved
frames/frame.1 saved
frames/frame.2 saved
frames/frame.3 saved
frames/frame.4 saved
frames/frame.5 saved
frames/frame.6 saved
frames/frame.7 saved
frames/frame.8 saved
frames/frame.9 saved
frames/frame.10 saved
frames/frame.11 saved
frames/frame.12 saved
frames/frame.13 saved
frames/frame.14 saved
frames/frame.15 saved
frames/frame.16 saved
frames/frame.17 saved
frames/frame.18 saved
frames/frame.19 saved
frames/frame.20 saved
frames/frame.21 saved
frames/frame.22 saved
frames/frame.23 saved
frames/frame.24 saved
```

```
frames\frame107.jpg is predicted as suspicious with probabil
ity : 87.31042742729187

frames\frame117.jpg is predicted as suspicious with probabil
ity : 87.54125237464905

frames\frame127.jpg is predicted as suspicious with probabil
ity : 87.45720386505127

frames\frame137.jpg is predicted as suspicious with probabil
ity : 87.35582828521729

frames\frame147.jpg is predicted as suspicious with probabil
ity : 87.27838397026062

frames\frame157.jpg is predicted as suspicious with probabil
ity : 87.52446174621582

frames\frame167.jpg is predicted as suspicious with probabil
ity : 87.66176104545593

frames\frame177.jpg is predicted as suspicious with probabil
ity : 87.74349689483643

frames\frame187.jpg is predicted as suspicious with probabil
```

Screenshot 7.6: Above figure in right text area we can see details of all frames which has such activities.

8. TESTING

8. TESTING

8.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/ora finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail unacceptably. There are various types of tests. Each test type addresses a specific testing requirement.

8.2 TYPES OF TESTINGS

8.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.2.3 FUNCTIONALITY TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.2.5 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional

Test Results: All the test cases mentioned above passed successfully. No defects were encountered

8.3 TEST CASES

Test Case ID	Description	Objective	Test Steps	Expected Result	Negative Test
TC 1	Uploading CCTV Footage	Verify that the system correctly accepts and processes video	1. Navigate to the video upload page. 2. Upload a valid CCTV video	The video is uploaded successfully and the system	Upload an unsupported file format. The system should display an error message.

		files.	file (e.g., .mp4).	generates confirmation.	
TC 2	Frame Generation	Ensure that the system correctly extracts frames from the uploaded video.	After uploading the video, click the "Generate Frames" button.	Frames are generated and saved, displayed with timestamps.	If no video is uploaded, the system should prompt the user to upload a file.
TC 3	Real-Time Detection of Suspicious Activity	Test the system's ability to detect suspicious activities.	1. Upload a video containing suspicious activities. 2. Start the detection process.	The system flags suspicious frames accurately.	Upload a video without suspicious activity; the system should report no suspicious behavior.
TC 4	Accuracy of Detection	Verify the accuracy of the system in distinguishing between normal and suspicious activities.	1. Upload a mix of normal and suspicious activity videos. 2. Check detection accuracy for both.	The system accurately classifies the activities, minimizing false positives and negatives.	N/A
TC 5	User Interface for Monitoring	Ensure the user-friendly dashboard displays alerts and activity reports effectively.	1. Navigate through the dashboard. 2. Check the display of real-time detection alerts and video playback.	All alerts and reports are displayed clearly, and the dashboard responds smoothly.	N/A
TC 6	Data Security and Privacy	Validate that the system securely handles video and detection data.	Upload video and monitor the system for any data leakage or unauthorized access.	Data is securely stored, and only authorized users can access sensitive information.	N/A
TC 7	System Scalability	Test the system's performance when processing large datasets or multiple video streams.	1. Upload multiple videos simultaneously. 2. Start the detection process for all videos.	The system handles the load without performance degradation.	N/A
TC 8	Error Handling	Verify how the system	1. Upload a corrupted video	The system shows	N/A

		responds to errors, such as corrupted video files or network issues.	file. 2. Simulate network disconnection during the detection process.	appropriate error messages and provides options to retry or cancel the process.	

9. CONCLUSION&FUTURESCOPE

9. CONCLUSION & FUTURE SCOPE

9.1 PROJECT CONCLUSION

In conclusion, convolutional neural networks (CNNs) have demonstrated
significant promise for detection of suspicious activity in areas such as surveillance and security. Their ability to learn complex visual patterns allows them to differentiate between normal and suspicious behaviors without extensive manual feature engineering.

The study highlights the importance of having a large, diverse, and well-annotated dataset for effective model training. However, challenges such as high computational demands and potential biases in training data remain.

Future research could focus on improving model efficiency, enhancing interpretability, addressing bias and fairness, and conducting real-world deployments. Overall, CNNs are poised to enhance suspicious activity detection capabilities, with ongoing research critical to unlocking their full potential

9.2 FUTURE SCOPE

1. Future enhancements in suspicious activity detection can leverage advanced algorithms
for more accurate behavior analysis.
2. Real-time data processing will enable immediate insights into security threats.
3. Adapting models based on user behavior and environmental changes will improve detection accuracy.
4. An intuitive user interface will allow security personnel to easily access and interpret alerts.
5. This will result in more informed decision-making and stronger security measures.

10. BIBLIOGRAPHY

10. BIBLIOGRAPHY

10.1 REFERENCES

[1] Zhou, J. T., Du, J., Zhu, H., Peng, X., & Goh, R. S. M. (2019). “Anomaly Net: An Anomaly Detection Network for Video Surveillance.” *IEEE Transactions on Information Forensics and Security*, 1(1), 99-105.

[2] Rokade, M. D., & Bora, T. S. (2021). "Survey on Anomaly Detection for Video Surveillance." *International Research Journal of Engineering and Technology (IRJET)*.

[3] Medel, J. R., & Savakis, A. (2017). “Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks.” *International Symposium on Neural Networks*, Springer, 189–196.

[4] Luo, W., Liu, W., & Gao, S. (2017). “A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework.” In *IEEE International Conference on Computer Vision (ICCV)*.

[5] Chong, Y. S., & Tay, Y. H. (2017). “Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder.” In *International Symposium on Neural Networks*, Springer, 189–196.

[6] Medel, J. R., & Savakis, A. (2016). “Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks.” arXiv preprint arXiv:1612.00390.

[7] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., & Davis, L. S. (2016). “Learning Temporal Regularity in Video Sequences.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 733–742.

[8] Sabokrou, M., Fathy, M., Hoseini, M., & Klette, R. (2015). “Real-Time Anomaly Detection and Localization in Crowded Scenes.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[9] Lu, C., Shi, J., & Jia, J. (2013). “Abnormal Event Detection at 150 FPS in MATLAB.” In *Proceedings of the IEEE International Conference on Computer Vision*, 2720–2727.

[10] Mousavi, H., Nabi, M., Galoogahi, H. K., Perina, A., & Murino, V. (2015). “Abnormality Detection with Improved Histogram of Oriented Tracklets.” In

10.2 Websites

1. **IJNRD (International Journal of Novel Research and Development)**
https://www.alertlogic.com/blog/detecting-suspicious-and-malicious-activity-on-your-network/

2. **Alert Logic** : https://www.ijnrd.org/papers/IJNRD2305443.pdf

3.https://www.mdpi.com/2079-9292/11/24/4210

4. https://www.pnrjournal.com/index.php/home/article/view/1151/928