

PLANT DISEASE IDENTIFICATION USING CNN

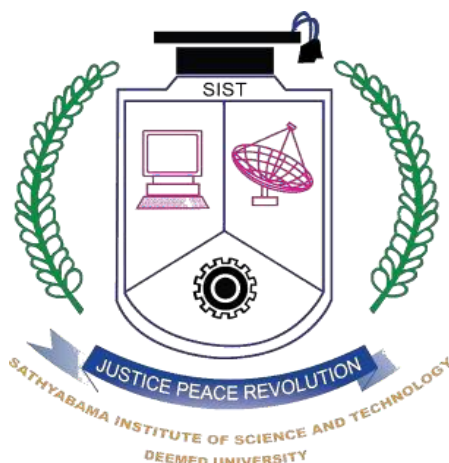
Submitted in partial fulfilment of the required for the award of

Masters of Computer Science

By

AISHWARYA D

REGISTER NO (42734001)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

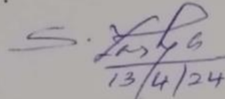
Accredited "A++" Grade by NAAC | 12B Status by UGC I Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

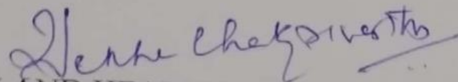
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AISHWARYA D (42734001)** who carried out the project entitled **"PLANT DISEASE IDENTIFICATION USING CNN"** under our supervision from JANUARY 2024 TO APRIL 2024.


13/4/24

Internal Guide

Ms. YUVA SNEHA M.Sc., Assistant Professor

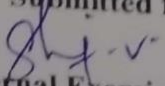


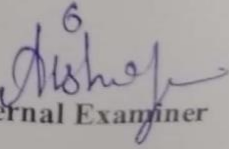
DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on

20/4/24


Internal Examiner

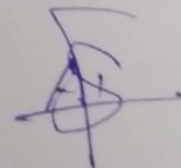

External Examiner

DECLARATION

I am, AISHWARYA D (42734001) hereby declare that the project report entitled " PLANT DISEASE IDENTIFICATION USING CNN " done by me under the guidance of MS. YUVA SNEHA S M.Sc., Assistant Professor, Department of Computer Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfilment of the requirements for the award of Master of Science degree in Computer Science .

DATE: 20/4/24

PLACE: Chennai



AISHWARYA D

Signature of the Candidate

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Ms. YUVA SNEHA, M.SC.CS., Assistant Professor** Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project

AISHWARYA D

ABSTRACT

Since the implementation of Tobacco Hazards Prevention Act, the prohibition against smoking in public places tends to be stricter in India. Its emphasis level can be inferred from the frequent law amendments in recent years. In order to effectively improve the prevention effect and bring down the costs of human surveillance, this project proposes a smoking detection system in public area with the human face analysis, which can quickly and accurately detect whether the person in the image is smoking or not. First, the system makes use of the existing facial detection scheme to locate the mouth, and then conducts white balance processing on the detected face image so as to reduce the misjudgment caused by the light. It then implements color segmentation and noise elimination in HSV color system in order to determine the location of the cigarette. The range between the location and the mouth is used to determine whether the person is smoking or not. This study hopes to combine with the indoor monitoring system to automatically detect the illegal behavior and send out alarms to notify the responsible units for reference. The system combined object detection with Artificial Neural Network with Haar Cascades on face recognition. Thus the implementation reduces to solve the problem of passive smoke affectors.

Front End: Python

Back End: MySQL

IDE: Pycharm Community Edition

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	1
1.1	OVERVIEW OF THE PROJECT	1
1.2	OBJECTIVE OF THE SYSTEM	1
1.3	EXISTING SYSTEM	2
1.4	PROPOSED SYSTEM	3
2	LITERATURE SURVEY	4
3	SYSTEM DESIGN	7
3.1	UML DAIGRAM	7
3.2	USE CASE DAIGRAM	7
3.3	SEQUENCE DIAGRAM	9
3.4	ACTIVITY DIAGRAM	10
4	SYSTEM ARCHITECTURE	13
5	SYSTEM IMPLEMENTAION	15
5.1	DATA FLOW DIAGRAM	15
5.2	DATA DICTIONARY	17
5.3	ER DIAGRAM	18
5.4	MODULE DESCRIPTION	18
5.5	CONVARIANCE EXTRACTION	21
5.6	FINE CHARGE	24
6	CODING AND SCREENSHOT	25
6.1	SAMPLE CODING	25
6.2	SAMPLE SCREENSHOT	36
7	CONCLUSION	42

LIST OF FIGURES

FIG NO	FIG NAME	PAGE NO
3.2	USE CASE DIAGRAM	8
3.3	SEQUENCE DIAGRAM	10
3.4	ACTIVITY DIAGRAM	12
4.1	SYSTEM ARCHITECTURE	14
5.1	LEVEL 0 DFD	16
5.2	LEVEL 1 DFD	17
5.3	ER DIAGRAM	18

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Smoking detection in public places is an important means to protect people's health and safety of life and property, but small target detection will have problems such as low detection accuracy and missed detection. In response to this problem, from the perspective of single-stage detection, a smoking detection model suitable for real-time monitoring is proposed. Based on the custom attention mechanism module and the improved residual network, design a backbone network that reuses underlying features while fusing features at different stages to enhance the ability to extract small target features and improve the accuracy of target detection; Smoking in public places not only causes harm to the health of oneself and others, but also has a great safety risk. Many fire incidents are caused by smoking in sensitive areas. Therefore, more and more public places begin to detect and control smoking behavior. Airports, high-speed trains, gas stations, flammable and explosive warehouses and other smoke-free areas need to be equipped with equipment that can accurately and efficiently monitor smoking behavior to ensure that firefighters and site management personnel can detect fire hazards in a timely manner. With the development of science and technology, the detection of smoking has been improved. Traditional detection methods mostly use various smoke detectors, but in open areas such as airports, gas stations and shopping malls, the smoke concentration will decrease rapidly, and the smoke sensing equipment cannot be triggered, so it is difficult to achieve the effect of monitoring and warning. Some researchers have also designed wearable detection devices, but they need to be worn by everyone. The production cost is high and the service life of the devices is short.

1.2 OBJECTIVE OF THE SYSTEM

The objective of a smoking detection and fine alert system is to effectively enforce smoking regulations in designated non-smoking areas and promote public health. This system utilizes advanced sensors like cameras or smoke detectors to continuously monitor areas where smoking is prohibited, such as public buildings, parks, and indoor facilities.

The primary goal is to automatically detect instances of smoking in these restricted zones. When smoking activity is detected, the system triggers an immediate response, such as issuing fines or notifying authorities. By doing so, the system aims to deter individuals from smoking in prohibited areas, reducing exposure to secondhand smoke and ensuring compliance with smoking policies.

Additionally, the system serves an educational purpose by raising awareness of smoking restrictions and the importance of maintaining smoke-free environments. It contributes to creating healthier and more respectful public spaces for everyone, particularly non-smokers who may be negatively impacted by exposure to tobacco smoke.

In summary, the smoking detection and fine alert system is designed to improve public health outcomes by enforcing smoking regulations, reducing smoking-related nuisances, and fostering a culture of compliance with smoking policies in designated non-smoking areas.

1.3 EXISTING SYSTEM

In existing system, personal identification number (PIN) is given for each individual to secure their amount .But this seems vulnerable to the risk of exposure and it can easily be forgotten. Fake cards can also be made for stealing the amount from the ATM. Recently many hackers have hacked people's account. For example, palm prints and palm prints are usually frayed; voice, signatures, hand shapes and iris images are easily forged. So we use biometric authentication process. The biometric systems which we come across frequently nowadays are palmprints, voice, hand geometry, palm print, iris or pupil, palm vein, palm vein and many more. Palmprint is an easy technique which we come across more often than the others even at public places not just for security purpose but also for attendance or as a substitute for signatures. Although the other systems have also been used quite regularly for security purpose and none of these are full proof and insusceptible to errors. All these biometric techniques have their pros and cons.

- The system will be the one where the concern smoking might affect
- The smoking problem will be the major cause of pollution

1.4 PROPOSED SYSTEM

It has been noticed that in many countries citizens are suffering because of second-hand smoking and this problem is increasing rapidly. This project will provide a solution to the problem of smoking in public areas by using the concept of Image Processing. Initially the moving objects are detected in the captured video frame by using background estimation method. Then person's face in the captured frame is detected using Haar classifier and the presence of cigar is determined using template matching. By using the features of smoke and Gaussian blur is applied on the frame for noise removal and for smoothening, the presence of smoke in a frame is identified. The more layers the neural network has, the richer the features can be extracted and the stronger the representation ability is. However, with the increase of network depth, there will be problems of gradient explosion and dissipation. Another significant problem is model degradation, and the performance of network will get worse and worse

- The proposed system identifies the concern face and make fine
- The problem is that the one where the user smoking can be avoided

CHAPTER 2

LITERATURE SURVEY

2.1 SMOKE 100K: A DATABASE FOR SMOKE DETECTION

Hsiang-Yin Cheng; Jia-Li Yin; Bo-Hao Chen; Zhi-Min Yu

Due to the complex scenarios and the limited feature information in a single image, a precise smoke detection is much more challenging in practice. Most of previous smoke detection methods either extract textural and spatiotemporal characteristics of smoke or separate the smoke and background components of the image. However, those methods often fail in detecting smoke positions because of the limited feature information within a single image. Moreover, the task of smoke detection can be better achieved if the extra information from collected training dataset is available. One key issue is how to build a training dataset of paired smoke images and ground-truth bounding box positions for end-to-end learning. This paper proposes a large-scale benchmark image dataset to train a smoke detector. With the built dataset, experimental results demonstrate that the discriminative models can be effectively trained as the smoke detector to detect the smoldering fires precisely.

2.2 A NOVEL FUZZY-BASED SMOKE DETECTION SYSTEM USING DYNAMIC AND STATIC SMOKE FEATURES

Yashar Deldjoo; Fatemeh Nazary; Ali M. Fotouhi

Automatic fire surveillance is an important task for providing emergency response in the event of unexpected fire hazards. Early detection of fire can substantially mitigate the ecological or economical costs associated with a fire disaster. In this regard, as smoke usually always precedes fire, an intelligent smoke detection system is proposed that exploits a Fuzzy Inference System (FIS) in order to aggregate the features of smoke. In addition, robust smoke feature detection algorithms are implemented that take into account both dynamic and static characteristics of smoke. The smoke features include motion, motion orientation (estimated by using the accumulation of motion) for the former and texture for the latter. Experimental results on different video frames show that the proposed smoke detection system has robust performance on detecting the existence of smoke which shows the effectiveness of the proposed smoke detection system.

2.3 AN ADAPTIVE DETECTION METHOD FOR EARLY SMOKE OF COAL MINE FIRE BASED ON LOCAL FEATURES

Na Li; Jiameng Xue; Hongan Li

Aiming at the particularity of coal mine conditions, and in order to improve the effectiveness of smoke detection method in actual early detection, an early fire image smoke detection method based on adaptive histogram algorithm is proposed in this paper. Firstly, the principle of adaptive histogram algorithm for smoke image is explained, and the calculation method of line by line sliding window histogram is introduced. Then, the smoke detection process is given based on image adaptive histogram feature to realize coal mine smoke detection. Finally, the method is used to realize target detection in different smoke scenes. In the experiment, the traditional edge detection operator, smoke detection based on global histogram and the adaptive histogram smoke detection in this paper are used to detect smoke image and flame smoke respectively. The results show that this method can better reflect the motion characteristics of smoke at the edge, and contains large information entropy, which shows that this method shows good performance in smoke detection.

2.4 SMOKING BEHAVIOR DETECTION BASED ON TF-YOLOV5

Runqing Li; Bo Song; Xianglan Liu; Xuguang Pan; Lisu Han; Yan Xu

In this paper, the determination of smoking behavior is considered as a target localization problem by building a neural network, and the advanced YOLOv5 is chosen as the baseline network. For the problem of low accuracy of YOLOv5 in detecting small targets, an improved TF-YOLOv5 is proposed by adding Transformer modules to the feature extraction part and feature fusion part, respectively, so that the network has the prediction potential of self-attention. In addition, we also integrate Coordinate Attention to find attentional regions in dense scenes, further improving the detection accuracy of our proposed TF-YOLOv5 for small targets in dense scenes and enabling us to achieve better results in the detection of smoking behavior. Experiments on our homemade smoking behavior dataset. show that our proposed TF-YOLOv5 improves 1.8% over the baseline model (YOLOv5).

2.5 YOLO-CIGARETTE: AN EFFECTIVE YOLO NETWORK FOR OUTDOOR SMOKING REAL-TIME OBJECT DETECTION

Yankai Ma; Jun Yang; Zhendong Li; Ziqiang Ma

Smoking behavior is the most important factor that causes fire, most of the traditional smoking detection is to detect the smoke produced by the burning of cigarettes, but it is not applicable in public places with strong air circulation. This paper designs a smoking image detection model: YOLO-Cigarette, which is based on deep learning, it solves the problems of low detection accuracy caused by small cigarette targets and complex background environment in the existing cigarette detection algorithms. The original YOLOv5 network structure is improved by fusing the new fine-grained spatial pyramid pooling module (FSPP) and multi spatial attention mechanism (MSAM), effectively improving the accuracy of small target detection. At the same time, the parameters in the model are quantized to greatly reduce the calculation of the model and improve the detection speed. YOLO-Cigarette model is 4.8% better than the original model in map, and achieves 95% accuracy in the test set.

CHAPTER 3

SYSTEM DESIGN

3.1 UML DIAGRAMS

A UML (Unified Modeling Language) diagram is a graphical representation of a system or process using standardized symbols and notation. UML diagrams are widely used in software engineering and other fields to visually depict various aspects of a system, such as its structure, behavior, interactions, and architecture.

There are several types of UML diagrams, each serving a different purpose and focusing on different aspects of the system being modeled. Some common types of UML diagrams include:

Unified Modeling Language (UML) diagram for the proposed automated college timetable scheduling system. Here's an example of a high-level UML class diagram for the system:

3.2 USE CASE DIAGRAM

A use case diagram is a graphical representation of the interactions between actors (users or external systems) and a system under consideration. It illustrates the various ways users interact with the system to accomplish specific tasks or achieve certain goals.

Key components of a use case diagram include:

Actors: Actors represent the users or external systems interacting with the system. They are typically depicted as stick figures or icons outside the system boundary. Actors can be individuals, other systems, or even time-based triggers.

Use Cases: Use cases represent specific functionalities or tasks that the system performs to fulfill user goals. Each use case describes a particular interaction between the user and the system. Use cases are depicted as ovals within the system boundary, often labeled with brief action-oriented descriptions.

Relationships: Relationships illustrate the connections between actors and use cases. The primary relationship in a use case diagram is the association between an actor and a use case, indicating

which actors are involved in which use cases. Relationships can also show dependencies, generalizations, or extensions between use cases.

System Boundary: The system boundary defines the scope of the system being modeled. It encloses all the internal components and use cases of the system, distinguishing them from external actors and systems.

Use case diagrams are valuable tools for understanding and communicating system requirements because they provide a high-level overview of system functionality from a user's perspective. They help stakeholders visualize how users will interact with the system and the tasks they will perform, facilitating discussions about system behavior and requirements..

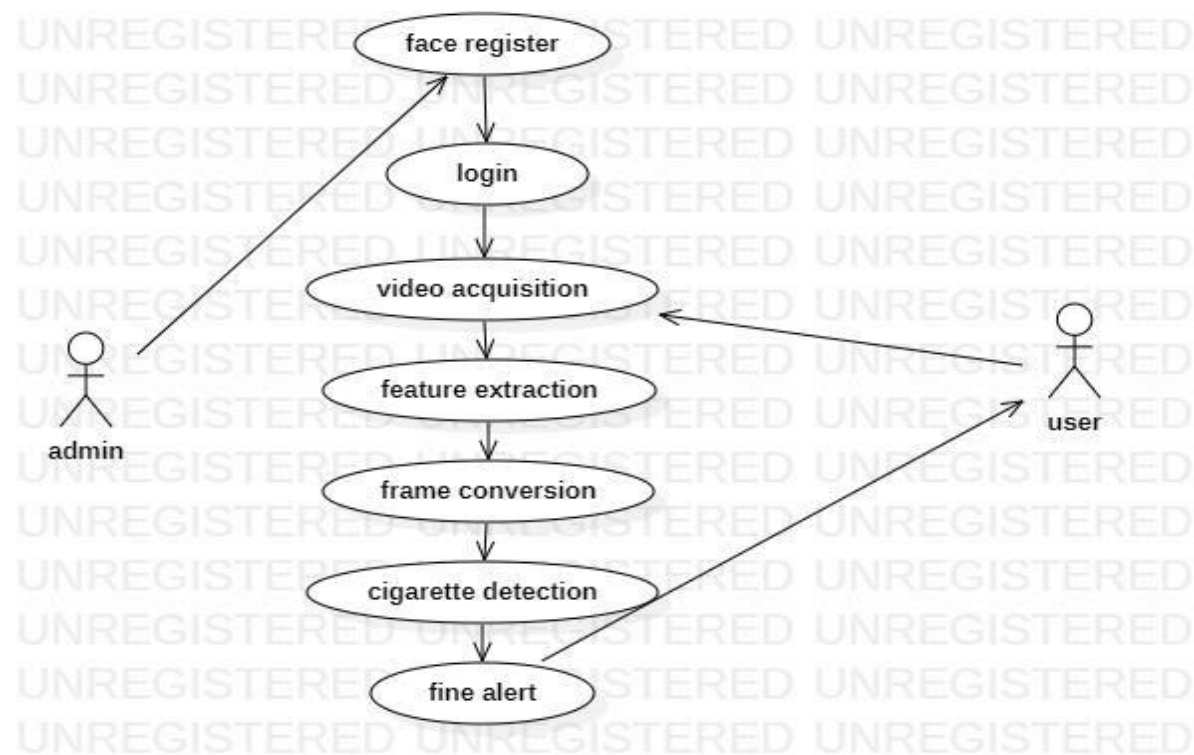


Fig 3.2 Use case diagram

3.3 SEQUENCE DIAGRAM

A Sequence Diagram is a type of UML (Unified Modeling Language) diagram that illustrates the interactions between objects or components within a system in a sequential manner. It focuses on the flow of messages or method calls exchanged between these objects over time to accomplish a specific behavior or scenario.

Key components of a sequence diagram include:

Objects/Participants: Represent the entities (objects, components, or actors) involved in the interaction. Each object is typically represented by a box or lifeline vertically aligned along the diagram's left side.

Lifelines: Vertical lines extending downwards from the objects, representing the existence of the objects over time during the interaction.

Messages: Arrows drawn between lifelines, indicating the flow of communication between objects. Messages can be synchronous (e.g., method calls), asynchronous (e.g., signals or events), or replies.

Activation/Execution Occurrences: Represented by narrow rectangles on lifelines, showing the duration during which an object is actively processing a message or performing an operation.

Optional Elements: Other optional elements may include conditionals (such as alternative or loop constructs), iterations, and parallel execution to capture more complex scenarios. This sequence diagram illustrates the sequential flow of interactions between the Administrator, the College Timetable System, and the Staff during the process of generating and approving a timetable.

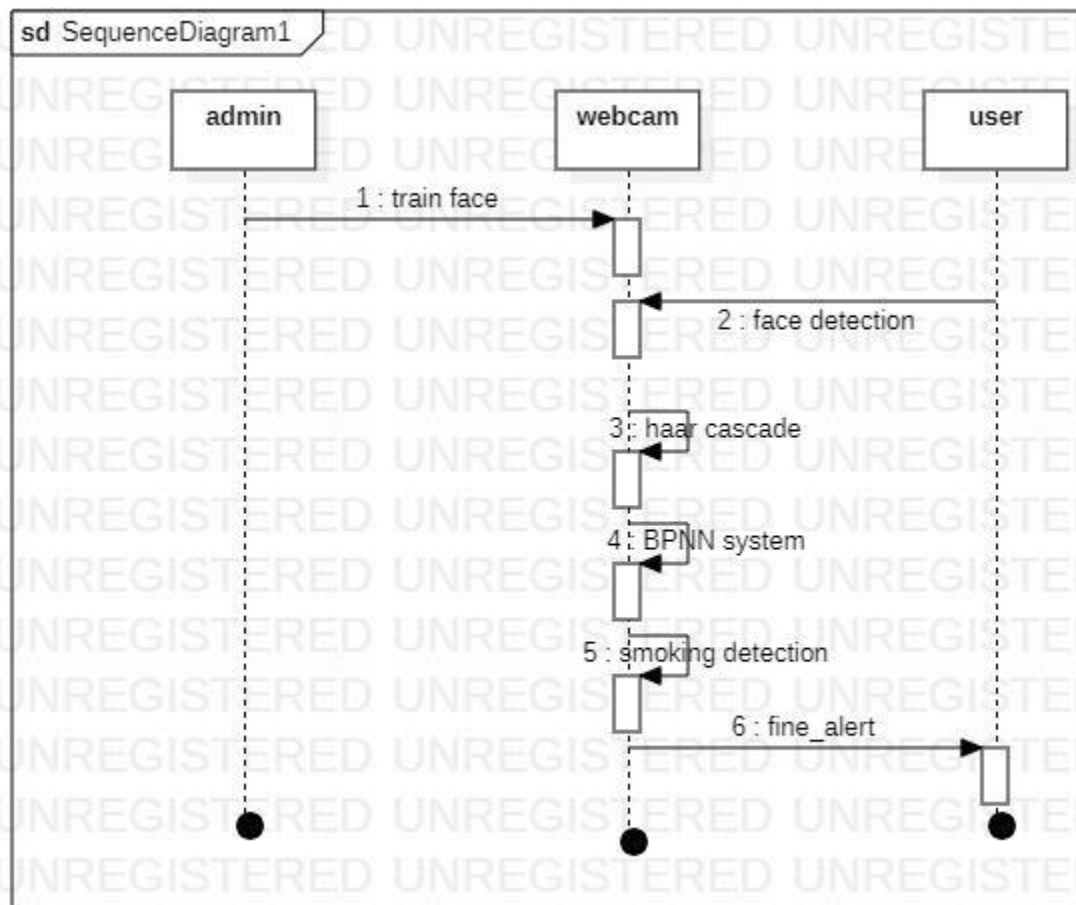


Fig 3.3 Sequence Diagram

3.4 ACTIVITY DIAGRAM

An Activity Diagram is a type of UML (Unified Modeling Language) diagram that illustrates the flow of activities or actions within a system or a specific business process. It visually represents the sequence of actions, decisions, and transitions from one activity to another, allowing stakeholders to understand the workflow or behavior of a system or process.

Key components of an activity diagram include:

Activities: Represented by rounded rectangles, activities depict specific actions or steps within the system or process. Each activity represents a task that needs to be performed.

Transitions: Arrows connecting activities, indicating the flow of control from one activity to another. Transitions represent the order in which activities are executed, typically triggered by the completion of a preceding activity or by the satisfaction of a condition.

Decision Nodes: Diamond-shaped symbols used to represent decision points or branches in the workflow. Depending on the outcome of a decision, the flow may proceed along different paths.

Merge Nodes: Used to consolidate multiple flow paths back into a single path after a decision or branching point.

Initial Node: Indicates the starting point of the activity diagram. It represents the initiation of the workflow or process

Final Node: Represents the endpoint of the activity diagram, indicating the completion of the workflow or process.

Fork and Join Nodes: Fork nodes split the flow into multiple parallel paths, while join nodes merge parallel paths back into a single flow.

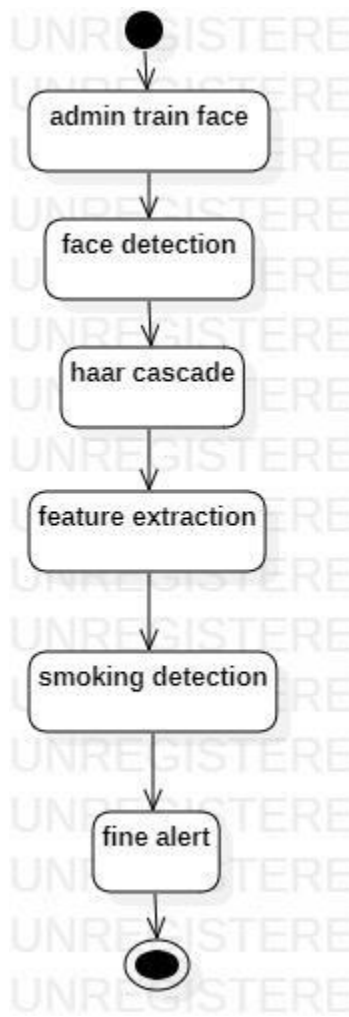


Fig 3.4 Activity Diagram

CHAPTER 4

SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architects or solution architects are the people who know what components to choose for the specific use case, make right trade-offs with awareness of the bottlenecks in the overall system. Usually, solution architects who have more years of experience tend to be good at system architecting because system design is an open-ended problem, there is no one correct solution. Mostly it's trial and error experiments done with right trade-offs. So, experience teaches you what components to choose based on the problem at hand but in order to gain that experience you need to start somewhere. That's why I am writing this article for the developers who have never done system designs.

The system architecture is mainly based on the fuzzy nature analytics where the learner behavior while learning a course and his style of learning is efficiently predicted using the intuitionistic fuzzy logic.

The system architecture for smoking detection in a public place involves a network of sensors, data processing systems, and alert mechanisms. This architecture typically includes deploying smoke detectors and cameras throughout the area to monitor for smoking activity. These sensors collect data continuously and transmit it to a centralized server or cloud platform. In the data processing phase, machine learning algorithms analyze the sensor data in real-time. This involves image processing for video feeds to detect smoke and behavioral pattern recognition for identifying smoking activities. Once smoking is detected, the system triggers an alert through an integrated notification system to notify relevant authorities or personnel for immediate response. The architecture also integrates with existing surveillance systems or public safety networks to ensure coordinated action. Privacy and compliance are key considerations, requiring anonymization of data and secure communication protocols to adhere to privacy regulations. This comprehensive architecture enables effective monitoring and response to smoking incidents in public places.

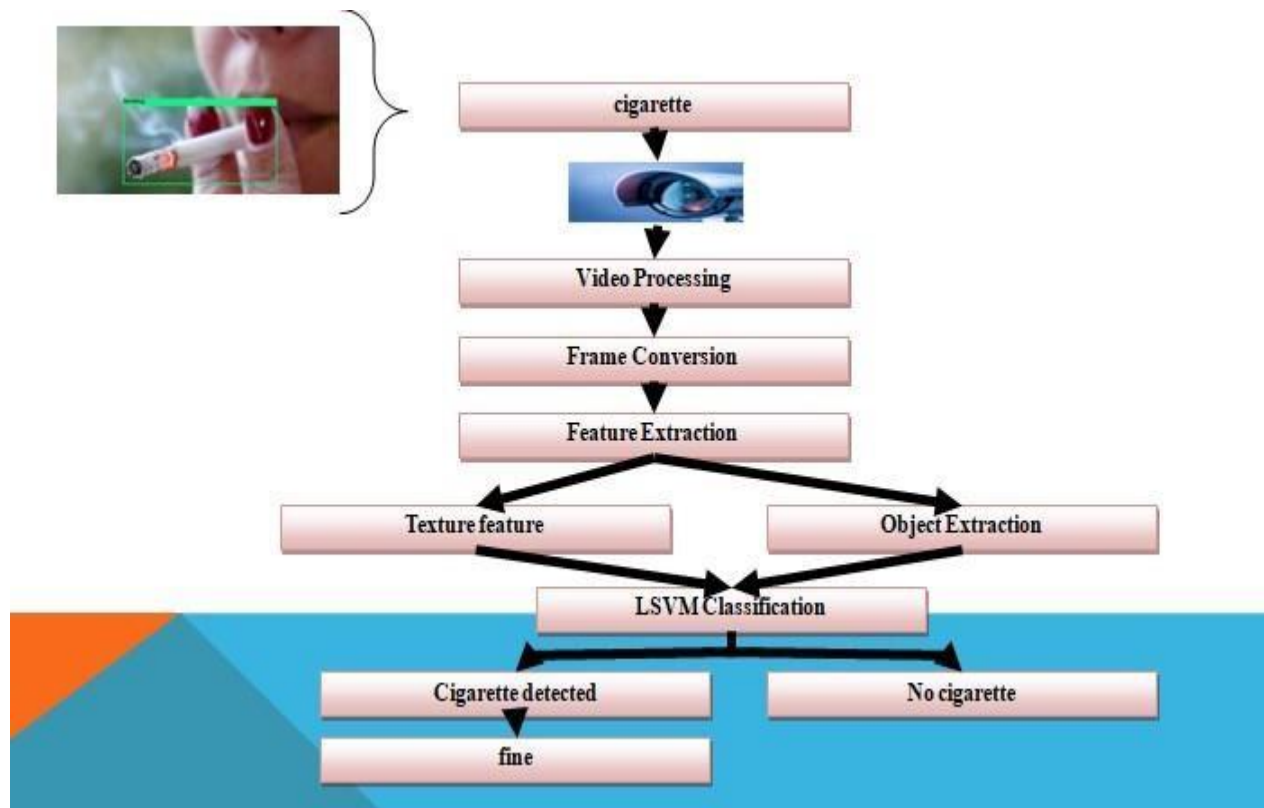


Fig 4.1 System Architecture

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols:

Symbol	Description
	An entity . A source of data or a destination for data.
	A process or task that is performed by the system.
	A data store , a place where data is held between processes.
	A data flow .

LEVEL 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

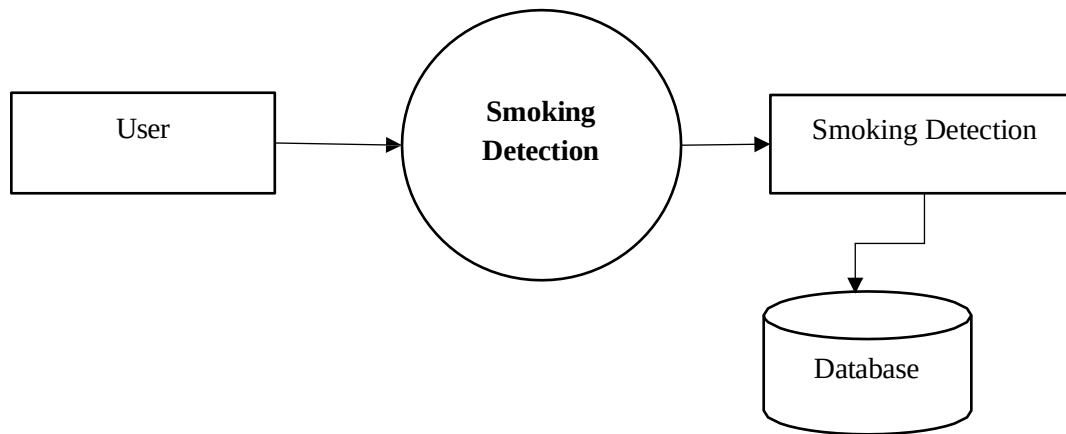


Fig 5.1. level 0 DFD

LEVEL 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes. A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into sub processes that can then be analyzed and improved on a more intimate level.

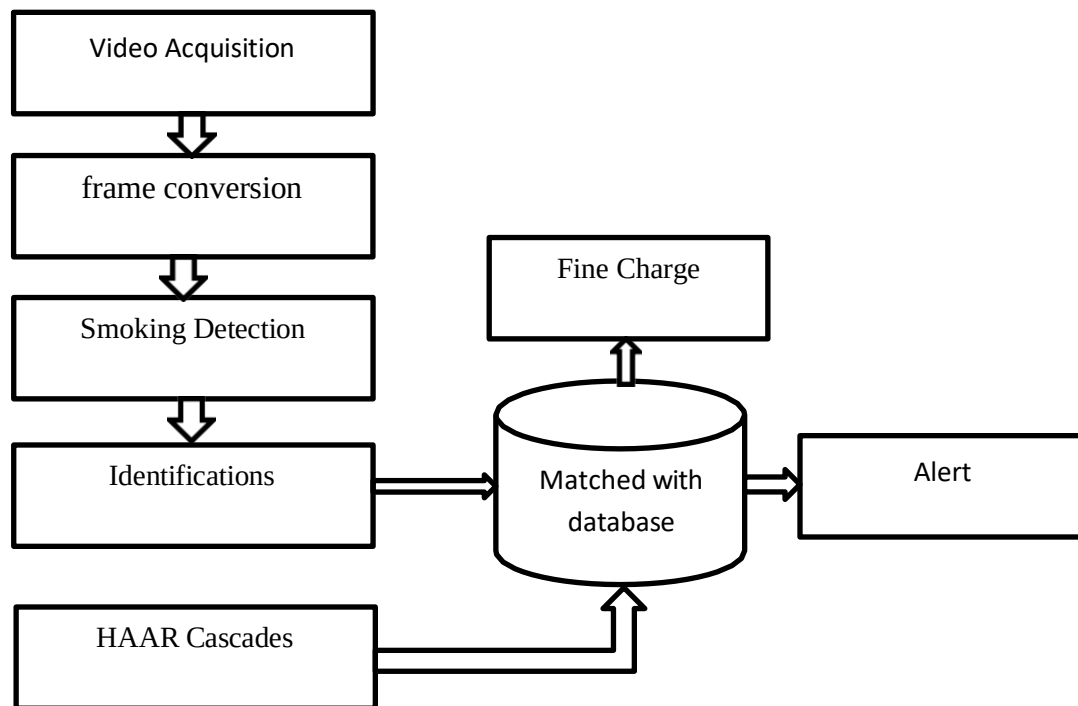


Fig 5.2 Level 1 DFD

5.2 DATA DICTIONARY

User Register table

Field	Type	Null	Default
id	int(100)	Yes	NULL
name	varchar(100)	Yes	NULL
contact	varchar(100)	Yes	NULL
email	varchar(100)	Yes	NULL
address	varchar(100)	Yes	NULL
aadhar number	varchar(100)	Yes	NULL
password	varchar(100)	Yes	NULL

5.3 ER DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

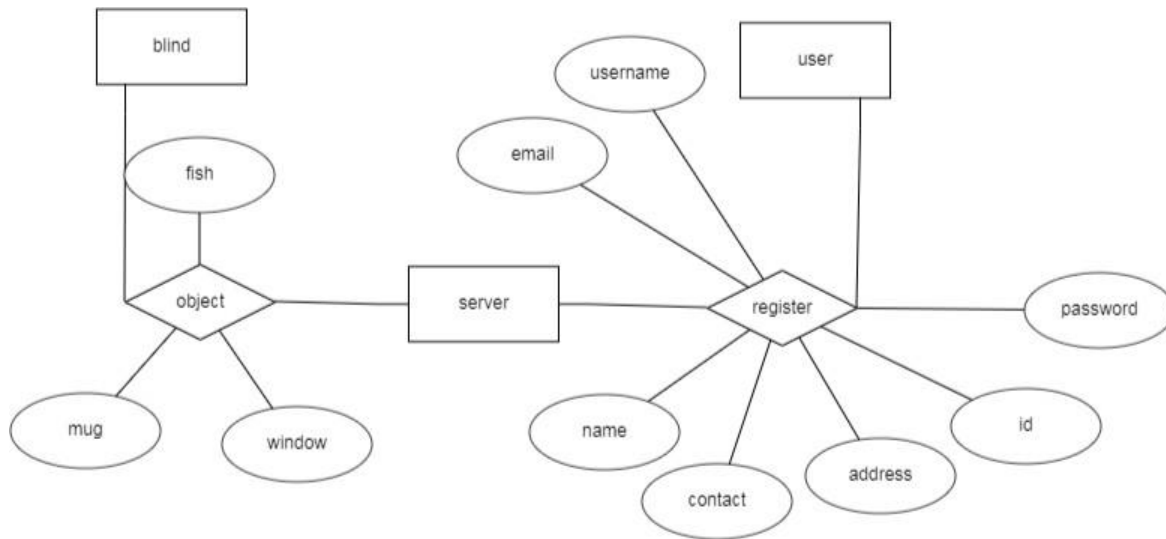


Fig 5.3 ER Diagram

5.4 MODULE DESCRIPTION

VIDEO STREAMING

Smoking recognition is a classical problem in computer vision: the task of determining if the image data contains a specific smoking and it is noted that general smoking recognition approaches exploit features extraction. Features that have received the most attention in the recent years are the local features. The main idea is to focus on the areas containing the most discriminative information. The detection of an smoking cannot be achieved with high accuracy in normal cases, as most of the application provides the detection of smoking captured as an image rather than detection in a live streaming video.

The proposed system captures the image through the continuous video stream rather than taking the picture of each smoking every time. The smoking detection and recognition have become easier when it is detecting from the image but in the case of the video stream, the processing speed should be high to detect the entire smoking in a frame. The functional process is to capture the images through the camera and process it through image processing algorithms.

KEY FRAME EXTRACTION

The frame extraction play the vital role in many video processing applications like content based video retrieval, shot detection, segmentation, CC cameras , etc. The frame conversion can be get with the seconds of the video which are get with the video. Each frame will be analyzed to know the smoking in the scene. The frame conversion is the process of extracting the images from the video where the sequences of images can be delivered as frames with the given video. In this project, a new technique for key frame extraction is presented. The scheme uses an aggregation mechanism to combine the visual features extracted from the correlation of RGB color channels, color histogram, and moments of inertia to extract key frames from the video. An adaptive formula is then used to combine the results of the current iteration with those from the previous. The use of the adaptive formula generates a smooth output function and also reduces redundancy. The results are compared to some of the other techniques based on smokingive criteria. The experimental results show that the proposed technique generates summaries that are closer to the summaries created by humans.

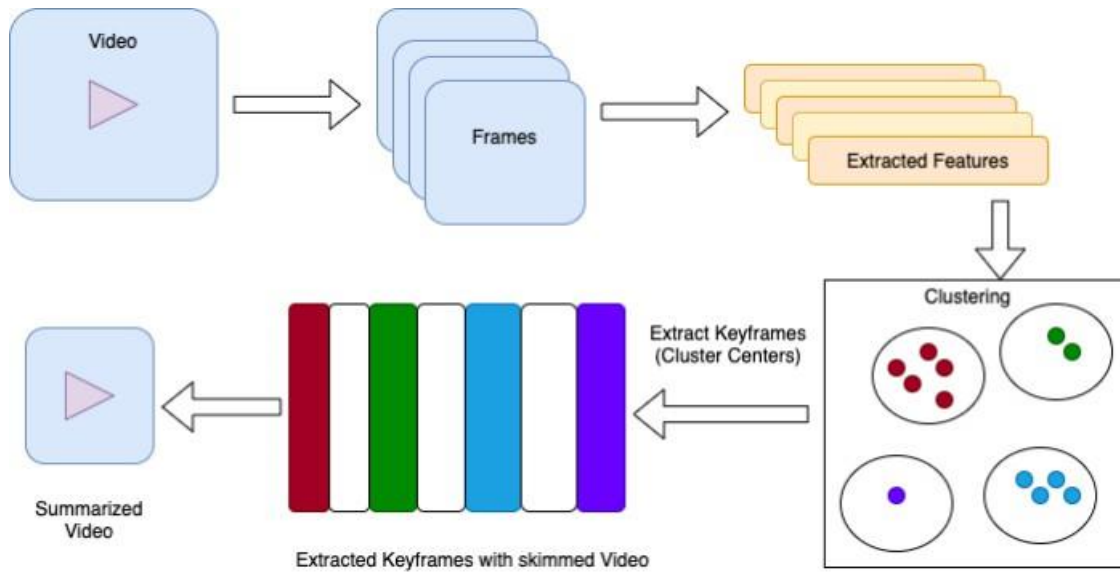


Fig 5.4 Summarized key frame extraction from video streaming

1) Uniform Sampling: Uniform sampling is one of the most common methods for keyframe extraction. The idea is to select every k th frame from the video where the value of k is dictated by the length of the video. A usual choice of length for a summarized video is 5% to 15% of the original video, which means every 20th frame in case of 5% or every 7th frame in case of 15% length of the summarized video is chosen. For our experiment, we have chosen to use every 7th frame to summarize the video. This is a very simple concept which does not maintain semantic relevance. Uniform sampling is often considered as a generalized baseline for video summarization.

2) Image histogram: Image histograms represent the total distribution of an image. It gives us the number of pixels for a specific brightness values rated from 0 to 256. Image histograms contain important information about images and they can be utilized to extract keyframes. We extract the histogram from all frames. Based on the difference between histograms of two frames, we decide whether the frames have significant dissimilarities among them. We infer that, a significant inter-frame image histogram dissimilarity indicates a rapid change of scene in the video which might contain interesting components. For our experiments, if histograms of two consecutive frames are 50% or more dissimilar, we extract that frame as a keyframe.

3) Scale Invariant Feature Transform: Scale Invariant Feature Transform (COVARIANCE) , has been one of the most prominent local features used in computer vision is applications en for each channel resulting in a 16X16X16 tensor. Due to computational reasons, a simplified version of this histogram was computed, where each channel was treated separately, resulting in feature vectors for each frame belonging to $\mathbb{R}^{48}$. The next step suggested for clustering is slightly different. But, the simplified color histograms give comparable performance to the true color histograms. Therefore, we modified VSUMM a bit by using the features extracted from VGG16 at the 2nd fully connected layer, and clustered it using kmeans.

5) ResNet16 on ImageNet: While reading about approach of VSUMM, we decided to test a different approach. We chose ResNet16 trained on image net, with different range of filters, and chopped of last loss layer, so as to obtain the embeddings of each image (512 dimension). We extracted frames out of the videos, and forward pass them through ResNet16, and after obtaining the embeddings for each frame in video, we clustered them using 2 algorithms: K-means and Gaussian Mixture Models. The number of cluster has been take as 15% of the video frame numbers. We later chose the frames closest to the center of clusters as the keyframes.

5.5 COVARIANCE EXTRACTION

The main purpose of using features instead of raw pixel values as the input to a learning algorithm is to reduce/increase the in-class/out-of class variability compared to the raw input data, and thus making classification easier.

Types of features that can be extracted from image depend on the type of image, the level of granularity desired, and the context of the application. Once the features have been extracted, theirs representation depends on the technique used. The features extraction process should be precise, so that the same features are extracted on two images showing the same smoking.

Global image descriptor: features overall image are usually based on color indices and the most famous global color descriptor is the color histogram.

Local image descriptors: Local features are ones that have received the most attention in recent years. The main idea is to focus on the areas containing the most discriminated information.

Semi-local image descriptor: most shape descriptors fall into this category. This descriptor is based on extracting accurate contours of shapes in the image or in the region of interest. In this case, image segmentation is generally useful as a preprocessing step.

COVOLUTIONAL NEURAL NETWORK FOR CLASSIFICATION

The artificial neural network classification system is used to classify the co-variance descriptor points. Feature matching using invariant features has obtained significant importance due to its application in various recognition problems. Such techniques have enabled us to match images irrespective of various geometric and photometric transformations between images.

For the smoking detection and labeling the machine learning method is used. The Artificial Neural Network is used for training and testing purpose. The region based segmentation is used and then they are segmented to identify the smoking. The smoking detection and recognition have become easier when it is detecting from the image but in the case of the video stream. We first take a pre-trained Artificial Neural Network. Then, this model is retrained. We train the last layer of the network based on the number of classes that need to be detected. The third step is to get the Region of Interest for each image. We then reshape all these regions so that they can match the ANN input size. After getting the regions, we train ANN to classify smokings and background. For each class, we train one binary ANN. Finally, we train a linear regression model to generate tighter bounding boxes for each identified smoking in the image.

Step (1) Obtain the set of key-points of smokings

- a. Select a large set of images of daily smokings.
- b. Extract the CO-VARIANCE feature points of all the images within the set and obtain the CO-VARIANCE descriptor for each feature point extracted from each image.

Step (2) Obtain the keypoints descriptor for the first video frame.

- a. Extract CO-VARIANCE feature points of the given image.
- b. Acquire CO-VARIANCE descriptor for each feature point.
- c. Match the frame key-points with those of the smokings and identify detected smokings.

Step (3) For the next frame:

- a. If it contains the same smokings they will not be detected
- b. New smokings will be detected and identified.
- c. Another method will be used in this step to identify similar and dissimilar frames for further treatments.

Step (4) For each smoking detected in video a video file is launched to notify the blind about the identity of smokings.

SMOKING DETECTION AND RECOGNITION

The smoking detection method will be done in the testing process. With the region of interest the smoking in the scene are detected. The detected smokings will be further labeled with the name of the smoking. Thus the smoking recognition is made here.

Smoking detection is the process of finding instances of real-world smokings such as faces, bicycles, and buildings in images or videos. Smoking detection algorithms typically use extracted features and learning algorithms to recognize instances of an smoking category. It is commonly used in applications such as image retrieval, security, surveillance, and automated vehicle parking systems. Detecting a reference smoking (left) in a cluttered scene (right) using feature extraction and matching. RANSAC is used to estimate the location of the smoking in the test image. Local features and their descriptors are the building blocks of many computer vision algorithms. Their applications include image registration, smoking detection and classification, tracking, and motion estimation. These algorithms use local features to better handle scale changes, rotation, and occlusion.



Fig 5.5 Shows the smoking features extraction at each key frame

The features we find are described in a way which makes them invariant to size changes, rotation and position. These are quite powerful features and are used in a variety of tasks. We use a standard Java implementation of SIFT. The SIFT descriptor is a 128 dimensional description of a patch of pixels around the key point.

5.6 FINE CHARGE

The fine charge is the one where the person without smoking will be placed with a fine amount where the concern fine placed using HAAR cascades.

CHAPTER 6

CODING AND SCREENSHOT

6.1 SAMPLE CODING

Ar_master

```
import pymysql

class master_flask_code:

    def __init__(self):

        self.user = 'root'

        self.password = ""

        self.host = 'localhost'

        self.database = 'python_smoke_detection'

    def find_max_id(self,table):

        conn = pymysql.connect(user=self.user, password=self.password, host=self.host,
        database=self.database)

        cursor = conn.cursor()

        cursor.execute("SELECT id FROM "+table)

        data = cursor.fetchall()

        maxin = len(data)

        if maxin == 0:

            maxin = 1

        else:

            maxin += 1

        return maxin

    def insert_query(self,qry):

        conn = pymysql.connect(user=self.user, password=self.password, host=self.host,
        database=self.database)
```

```

        cursor = conn.cursor()

        result=cursor.execute(qry)

        conn.commit()

        conn.close()

        return result

    def select_login(self,qry):

        conn = pymysql.connect(user=self.user, password=self.password, host=self.host,
        database=self.database)

        cursor = conn.cursor()

        cursor.execute(qry)

        data = cursor.fetchall()

        check = len(data)

        if check == 0:

            return 'no'

        else:

            return 'yes'

    def select_single_colum(self, table, colum):

        conn = pymysql.connect(user=self.user, password=self.password, host=self.host,
        database=self.database)

        qry1 = ("select " + colum + " from " + table)

        cursor = conn.cursor()

        cursor.execute(qry1)

        data = cursor.fetchall()

        return data

    def select_entire_colum(self,table,colum):

        conn = pymysql.connect(user=self.user, password=self.password, host=self.host,
        database=self.database)

```

```

qry1=("select * from "+table+" where symptoms LIKE '%" +column+"%")

cursor = conn.cursor()

cursor.execute(qry1)

data = cursor.fetchall()

return data

def select_direct_query(self,qry):

    conn = pymysql.connect(user=self.user, password=self.password, host=self.host,
database=self.database)

    cursor = conn.cursor()

    cursor.execute(qry)

    data = cursor.fetchall()

    return data

```

Image Hash

''''''

Image hashing library

=====

Example:

```

>>> from PIL import Image

>>> import imagehash

>>> hash = imagehash.average_hash(Image.open('test.png'))

>>> print(hash)

d879f8f89b1bbf

>>> otherhash = imagehash.average_hash(Image.open('other.bmp'))

>>> print(otherhash)

```

```
ffff3720200ffff
```

```
>>> print(hash == otherhash)
```

```
False
```

```
>>> print(hash - otherhash)
```

```
36
```

```
>>> for r in range(1, 30, 5):
```

```
...     rothash = imagehash.average_hash(Image.open('test.png').rotate(r))
```

```
...     print('Rotation by %d: %d Hamming difference' % (r, hash - rothash))
```

```
...
```

```
Rotation by 1: 2 Hamming difference
```

```
Rotation by 6: 11 Hamming difference
```

```
Rotation by 11: 13 Hamming difference
```

```
Rotation by 16: 17 Hamming difference
```

```
Rotation by 21: 19 Hamming difference
```

```
Rotation by 26: 21 Hamming difference
```

```
>>>
```

```
.....
```

```
from __future__ import (absolute_import, division, print_function)
```

```
from PIL import Image, ImageFilter
```

```
import numpy
```

```
font=('times', 16, 'bold')).pack()
```

```
Button(sc1, text='OK', command=del_sc1, fg="black", bg="#42bcf5", width=9,  
height=1,
```

```
activebackground="Red", font=('times', 15, 'bold')).place(x=90, y=50)
```

```

def clear():

    txt.delete(first=0, last=22)


def training():

    a = txt.get()

    if (a == ""):

        messagebox.showinfo(title="Alert", message="Enter Your Name")

    else:

        d = 0

        s1 = student;

        s1.name = a

        tt = tk_master()

        tt.test()

def testing():

    import test_1


w = 780

h = 500

ws = add_product.winfo_screenwidth()

hs = add_product.winfo_screenheight()

x = (ws / 2) - (w / 2)

y = (hs / 2) - (h / 2)

add_product.geometry('%dx%d+%d+%d' % (w, h, x, y))

add_product.title("SMOKE DETECTION")

```

```
message = Label(add_product, text="SMOKE DETECTION & FACE RECOGNITION",  
fg="#004080", width=45, height=3,
```

```
font=('times', 20, 'italic bold '))
```

```
message.place(x=12, y=20)
```

```
lbl = Label(add_product, text="USERNAME", width=20, height=2, fg="black",  
font=('times', 15, ' bold '))
```

```
lbl.place(x=30, y=200)
```

```
txt = Entry(add_product, validate="key", width=20, font=('times', 25, ' bold '))
```

```
txt.place(x=300, y=200)
```

```
compare_dataset = Button(add_product, text="TRAINING", width=15, height=1,  
fg="#FFF", bg="#004080",
```

```
command=training, activebackground="orange",  
activeforeground="white",
```

```
font=('times', 15, ' bold '))
```

```
compare_dataset.place(x=250, y=350)
```

```
resust_dataset = Button(add_product, text="TESTING", width=15, height=1, fg="#FFF",  
bg="#004080", command=testing,
```

```
activebackground="orange", activeforeground="white", font=('times', 15, '  
bold '))
```

```
resust_dataset.place(x=450, y=350)
```

```
add_product.mainloop()
```

```
b1 = Button(canvas1, text="Add Student", command=add_product, font=('times', 15, ' bold  
'))
```

```
canvas1.create_window(400, 150, window=b1)
```

```

def view_user():

    tt = tk_master()

    tt.view_user()

b1 = Button(canvas1, text="View User", command=view_user, font=('times', 15, 'bold '))

canvas1.create_window(400, 200, window=b1)

def logout1():

    tt = tk_master()

    tt.admin_fine()

b1 = Button(canvas1, text="Fine", command=logout1, font=('times', 15, 'bold '))

canvas1.create_window(400, 250, window=b1)


def logout():

    admin_home_root.destroy()


b1 = Button(canvas1, text="Logout", command=logout, font=('times', 15, 'bold '))

canvas1.create_window(400, 300, window=b1)


admin_home_root.mainloop()

def test(self):

    import cv2

    import sys

    import os


from sample_data import student


casePath = "data/haarcascades/haarcascade_frontalface_default.xml"

```

```

faceCascade = cv2.CascadeClassifier(cascPath)

train = True

video_capture = cv2.VideoCapture(0)

name = "train"

if os.path.exists(name):
    h = 0;
else:
    os.mkdir(name)


s1 = student;
print(s1.name)
name1 = "train\\" + s1.name
if os.path.exists(name1):
    j = 0;
else:
    os.mkdir(name1)

k = 0

while True:

    # Capture frame-by-frame
    ret, frame = video_capture.read()


    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #####

    if (frame is None):
        print("Can't open image file")

    face_cascade = cv2.CascadeClassifier(cascPath)

```

```

faces = face_cascade.detectMultiScale(frame, 1.1, 3, minSize=(100, 100))

if (faces is None):

    print('Failed to detect face')

if (True):

    for (x, y, w, h) in faces:

        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

        # lastimg = cv2.resize(frame, (32, 32))

        # cv2.imshow('img', frame)

facecnt = len(faces)

print("Detected faces: %d" % facecnt)

i = 0

height, width = frame.shape[:2]

for (x, y, w, h) in faces:

    r = max(w, h) / 2

    centerx = x + w / 2

    centery = y + h / 2

    nx = int(centerx - r)

    ny = int(centery - r)

    nr = int(r * 2)

    faceimg = frame[ny:ny + nr, nx:nx + nr]

    lastimg = cv2.resize(faceimg, (100, 100))

    i += 1

```

```

        k += 1

        if ((k < 20) & (k >= 5)):
            str = name1 + '\\%d.jpg' % k
            cv2.imwrite(str, lastimg)

cv2.imshow('Video', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# When everything is done, release the capture
video_capture.release()
cv2.destroyAllWindows()

def view_user(self):

    view_user = Toplevel()

def view_data():

    date_list = []

    TableMargin1 = Frame(view_user, width=700)
    TableMargin1.place(x=50, y=150, width=700, height=355)
    scrollbarx1 = Scrollbar(TableMargin1, orient=HORIZONTAL)
    scrollbarx1 = Scrollbar(TableMargin1, orient=VERTICAL)

```

```

tree1 = ttk.Treeview(TableMargin1, columns=("name", "contact", "email", "address"),
height=400,

                        selectmode="extended", yscrollcommand=scrollbary1.set,
xscrollcommand=scrollbarx1.set)

scrollbary1.config(command=tree1.yview)
scrollbary1.pack(side=RIGHT, fill=Y)
scrollbarx1.config(command=tree1.xview)
scrollbarx1.pack(side=BOTTOM, fill=X)
tree1.heading('name', text="name", anchor=W)
tree1.heading('contact', text="contact", anchor=W)
tree1.heading('email', text="email", anchor=W)
tree1.heading('address', text="address", anchor=W)
tree1.column('#0', stretch=NO, minwidth=0, width=0)
tree1.column('#1', stretch=NO, minwidth=0, width=200)
tree1.column('#2', stretch=NO, minwidth=0, width=200)
tree1.column('#3', stretch=NO, minwidth=0, width=200)
tree1.column('#4', stretch=NO, minwidth=0, width=200)
tree1.pack()

# tree1.bind('<Button-1>', selectItem)

if 1 == 2:

    messagebox.showinfo('Alert', "Enter User Id", parent=view_user)

else:

    data = mm.select_direct_query(

        "select * from user_details")

    # print(data)

    for x in data:

```

6.2 SAMPLE SCREENSHOT



Fig 6.2.1 Admin



Fig 6.2.2 Admin Login



Fig 6.2.3 Adding User Details

ADD USER DETAILS

Name

Contact

Email

Address

Username

Aadhar No

Password

ADD USER DETAILS

Name

Contact

Email

Address

Username

Aadhar No

Password

Alert

Registration Success

OK

Fig 6.2.4 User Detail Register

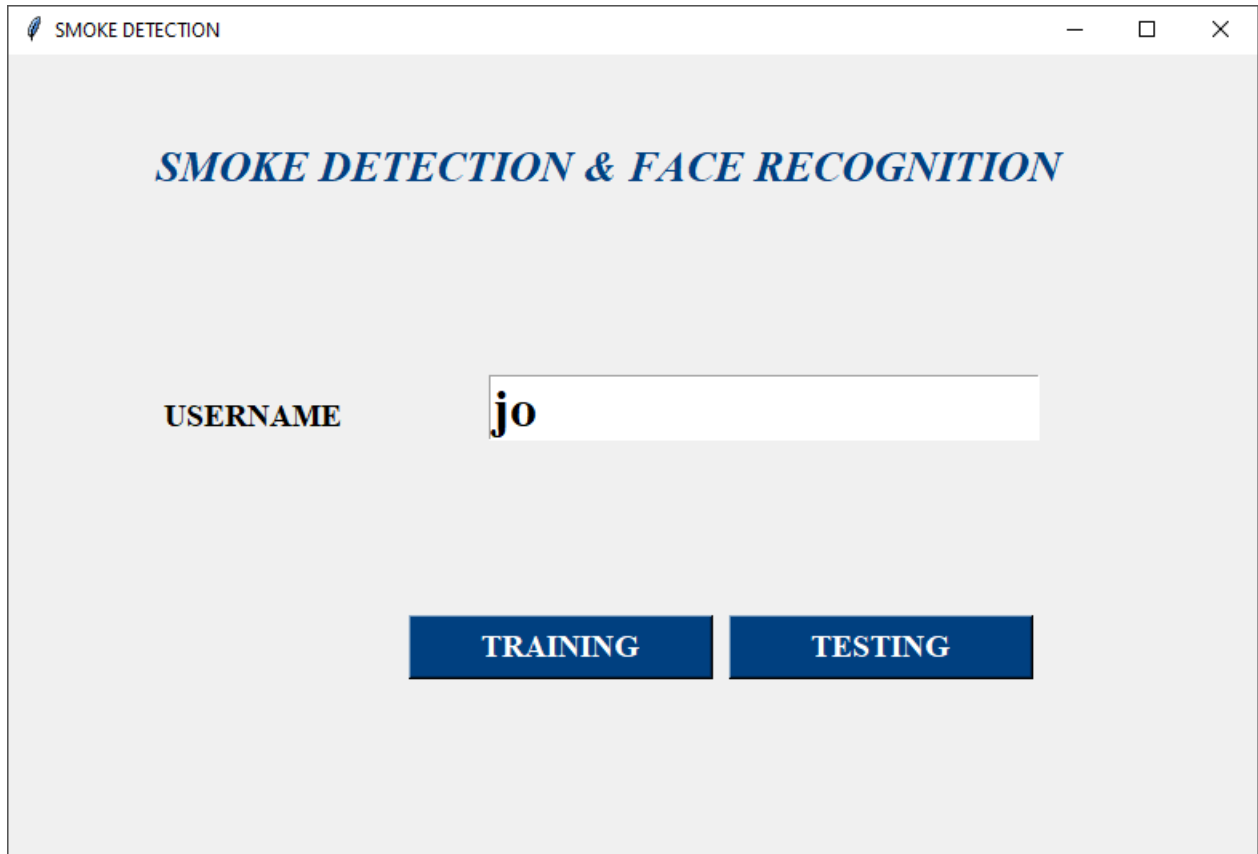


Fig 6.2.5 Face Recognition

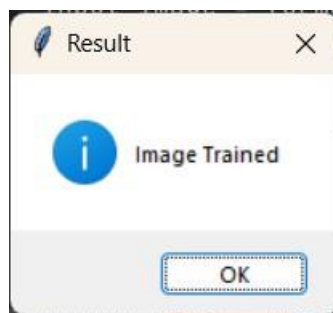
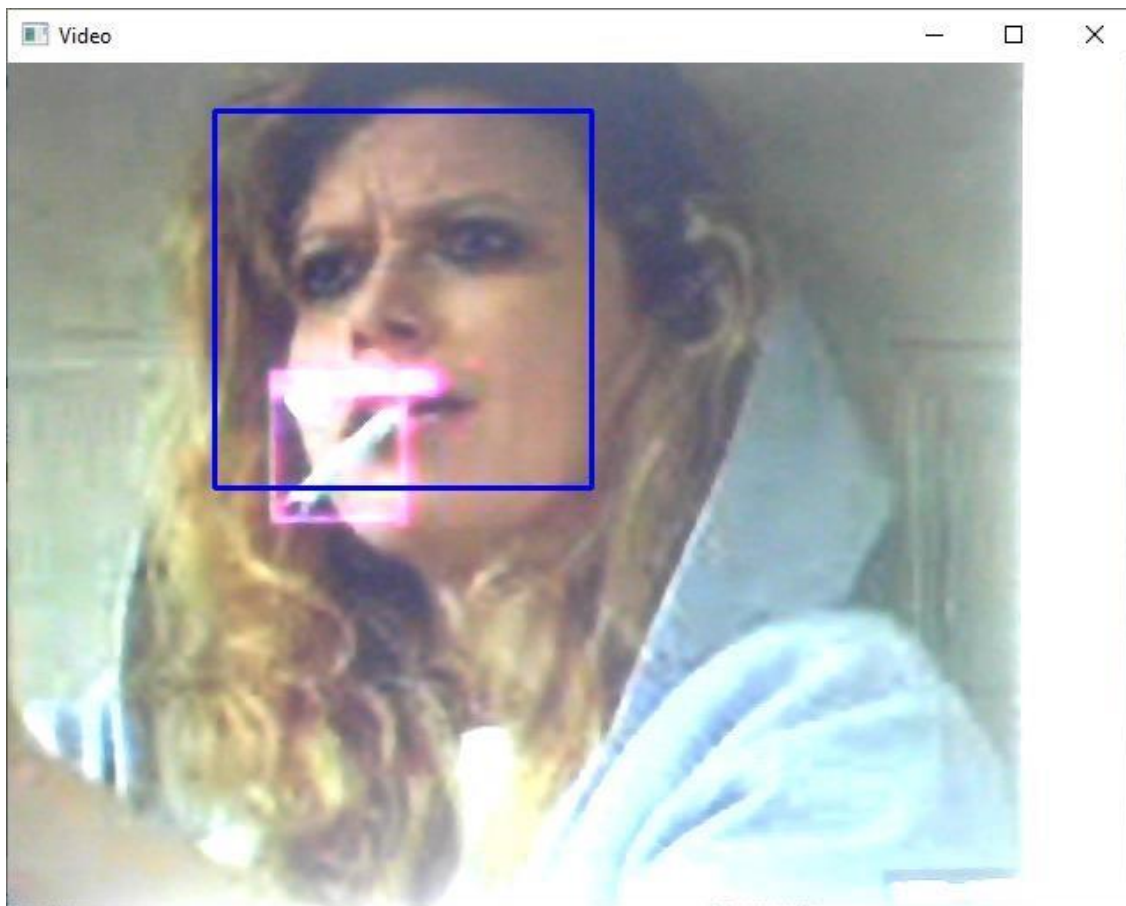


Fig 6.2.6 Image Training

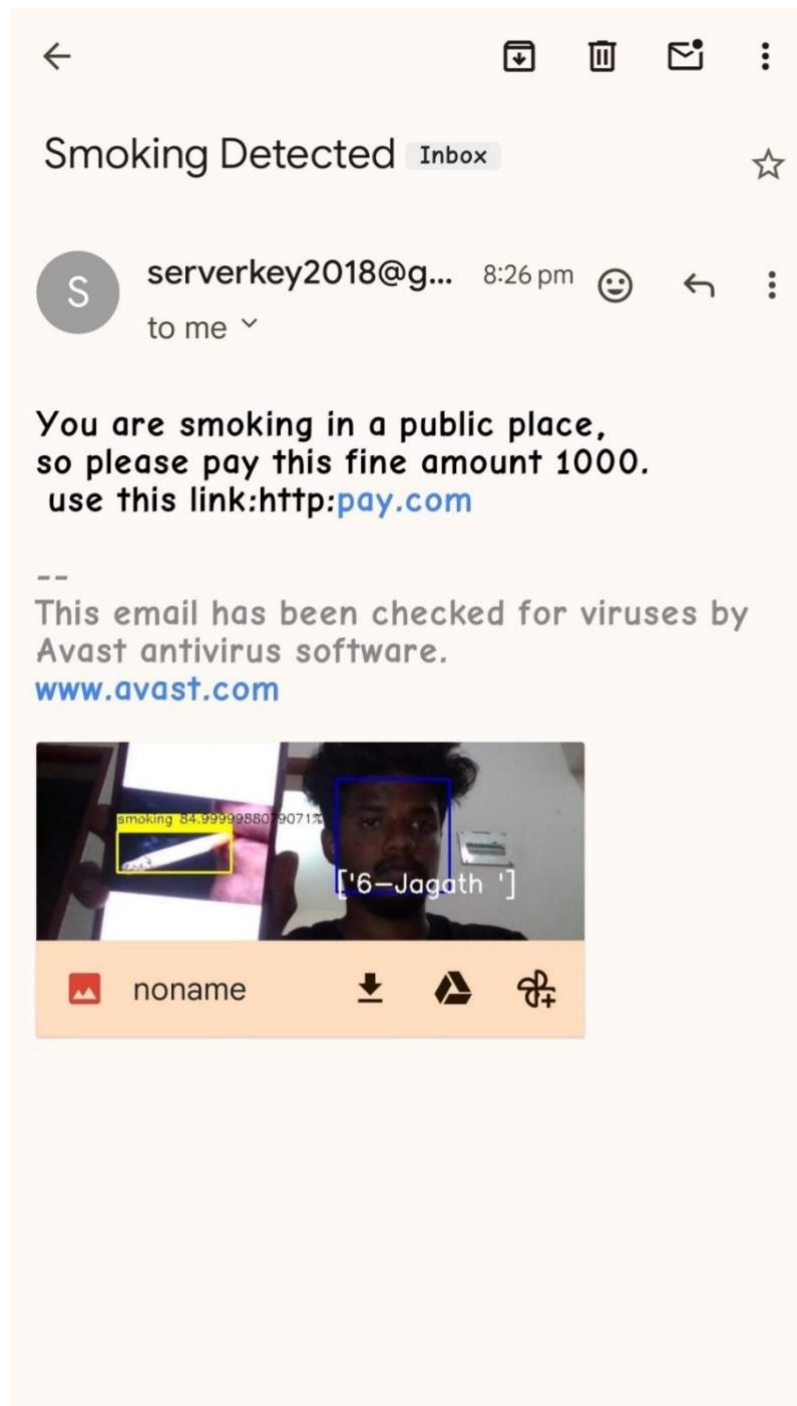


Fig 6.2.7 Fine Alert

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

In this project, the system has developed an automated system for smoking detection in public places which helps the people to avoid second-hand smoking. Also this system provides an easy way for the authorities to find the victims of smoking and to take required actions against the victim. According to the actual application scenarios of smoking detection in public places, this paper proposes a lightweight detection model, which has a good detection effect for cigarette detection. First of all, in order to improve the problem of low detection accuracy of small targets, draw lessons from the dense connection idea of the Dense Net network, improve the residual network, and design a feature extraction backbone network based on a custom attention mechanism module.

7.2 FUTURE ENHANCEMENT

- **Enhanced Sensor Technologies** : Develop more advanced sensor technologies capable of detecting smoking behaviors with higher accuracy and in various environmental conditions (e.g., outdoor settings, crowded spaces).
- **Integration with IoT** : Integrate smoking detection systems with Internet of Things (IoT) devices for seamless communication and data sharing. This could involve leveraging smart city infrastructure for improved coverage and response.
- **Real-time Analytics and Predictive Modeling** : Implement real-time analytics and predictive modeling using machine learning techniques to anticipate smoking hotspots based on historical data and current trends, enabling proactive intervention.
- **Multi-modal Detection** : Combine multiple detection modalities such as video analytics, audio recognition, and air quality sensors to improve the reliability and effectiveness of smoking detection, reducing false positives.
- **Privacy-preserving Solutions** : Develop privacy-preserving solutions that can detect smoking behavior without compromising individuals' privacy, utilizing techniques like edge computing to process data locally and only transmit relevant information.

REFERENCES

- Lopez-Meyer, P.; Tiffany, S.; Sazonov, E. Identification of cigarette smoke inhalations from wearable sensor data using a support vector machine classifier. In Proceedings of the Engineering in Medicine and Biology Society (EMBC), 2012
- Annual International Conference of the IEEE, San Diego, CA, USA, 28 August–1 September 2012; pp. 4050–4053.
- Patil, Y.; Tiffany, S.; Sazonov, E. Understanding smoking behavior using wearable sensors: Relative importance of various sensor modalities. In Proceedings of the Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, Chicago, IL, USA, 26–30 August 2014; pp. 6899–6902.
- Lopez-Meyer, P.; Patil, Y.; Tiffany, T.; Sazonov, E. Detection of hand-to-mouth gestures using a RF operated proximity sensor for monitoring cigarette smoking. *Open Biomed. Eng. J.* **2013**, 9, 41–49.
- Cole, C.A.; Anshari, D.; Lambert, V.; Thrasher, J.F.; Valafar, H. Detecting Smoking Events Using Accelerometer Data Collected Via Smartwatch Technology: Validation Study. *JMIR mHealth uHealth* **2017**, 5, e189.
- Cole, C.; Janos, B.; Anshari, D.; Thrasher, J.; Strayer, S.; Valafar, H. Recognition of smoking gesture using smart watch technology. In Proceedings of the International Conference on Health Informatics and Medical Systems, Las Vegas, NV, USA, 25 July 2016.