Deep Convolutional Neural Network for Detection of Cigarette Smokers in Public Places

A PROJECT STAGE-I PROJECT REPORT

Submitted to
Jawaharlal Nehru Technological University Hyderabad
In partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY IN

INFORMATION TECHNOLOGY

Submitted by

HARSH NAGDA	**20E11A1221**
T . CHANDRAKANTH KUMAR GUPTA	**20E11A1223**
MEDABOINA MALLIKARJUNA	**20E11A1204**
MALKAPURE KISHAN	**20E11A1210**
PATLA SIDDHARTHA	**20E11A1230**

Under the Supervision of
MS. FALAK NAAZ, M.Tech.
Assistant Professor

Department of Information Technology
BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY
Accredited by NAAC, Accredited by NBA (UG Programmes: CSE & ECE)
Approved by AICTE, Affiliated to JNTUH Hyderabad
Ibrahimpatnam-501 510, Hyderabad, Telangana.

DECEMBER 2023

DEPARTMENT OF INFORMATION TECHNOLOGY

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Accredited by NAAC, Accredited by NBA (UG Programmes: CSE & ECE)
Approved by AICTE, Affiliated to JNTUH Hyderabad
Ibrahimpatnam -501 510, Hyderabad, Telangana

Certificate

*This is to certify that the Project Stage-I project work entitled "**Deep Convolutional Neural Network for Detection of Cigarette Smokers in Public Places**"is the bona fide work done*

By

HARSH NAGDA (20E11A1221)
T. CHANDRAKANTH KUMAR GUPTHA (20E11A1223)
MEDABOINA MALLIKARJUNA (20E11A1204)
MALKAPURE KISHAN (20E11A1210)
PATLA SIDDHARTHA (20E11A1230)

in the Department of Information Technology, BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY, Ibrahimpatnam is submitted to Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirements for the award of B.Tech degree in Information Technology during 2020- 2024.

Guide:
Ms. Falak Naaz
Assistant Professor
Dept of CSE
Bharat Institute of Engineering and Technology
Ibrahimpatnam – 501510, Hyderabad.

Head of the Department:
Dr. S. Karpaga Selvi
Associate Professor
Dept of IT
Bharat Institute of Engineering and Technology
Ibrahimpatnam – 501510, Hyderabad.

Viva-Voce held on…………………………………………

Internal Examiner **External Examiner**

DEPARTMENT OF INFORMATION TECHNOLOGY

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Accredited by NAAC, Accredited by NBA (UG Programmes: CSE & ECE)
Approved by AICTE, Affiliated to JNTUH Hyderabad
Ibrahimpatnam -501 510, Hyderabad, Telangana

Vision of the Institution

To achieve the autonomous & university status and spread universal education by inculcating discipline, character and knowledge into the young minds and mould them into enlightened citizens.

Mission of the Institution

Our mission is to impart education, in a conducive ambience, as comprehensive as possible, with the support of all the modern technologies and make the students acquire the ability and passion to work wisely, creatively and effectively for the betterment of our society.

Vision of IT Department

Serving the high quality educational needs of local and rural students within the core areas of Computer Science and Engineering and Information Technology through a rigorous curriculum of theory, research and collaboration with other disciplines that is distinguished by its impact on academia, industry and society.

Mission of IT Department

The Mission of the department of Information Technology is

- ➢ To work closely with industry and research organizations to provide high quality Computer education in both the theoretical and applications of Computer Science and Engineering.
- ➢ The department encourages original thinking, fosters research and development, evolve innovative applications of technology.

DEPARTMENT OF INFORMATION TECHNOLOGY

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Accredited by NAAC, Accredited by NBA (UG Programmes: CSE & ECE)
Approved by AICTE, Affiliated to JNTUH Hyderabad
Ibrahimpatnam -501 510, Hyderabad, Telangana

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

The Information Technology program provides students with an in depth education in the conceptual foundations of Information Technology and Computer Science and in complex hardware and software systems. It allows them to explore the connections between Information Technology and a variety of other disciplines in engineering and outside. Combined with a strong education in mathematics, science, and the liberal arts it prepares students to be leaders in Information Technology practice, applications to other disciplines and research.

Program Educational Objective 1: (PEO1)

The graduates of Information Technology will have successful career in technology or managerial functions.

Program Educational Objective 2: (PEO2)

The graduates of the program will have solid technical and professional foundation to continue higher studies.

Program Educational Objective 3: (PEO3)

The graduates of the program will have skills to develop products, offer services and create new knowledge.

Program Educational Objective 4: (PEO4)

The graduates of the program will have fundamental awareness of Industry processes, tools and technologies.

DEPARTMENT OF INFORMATION TECHNOLOGY

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Accredited by NAAC, Accredited by NBA (UG Programmes: CSE & ECE)
Approved by AICTE, Affiliated to JNTUH Hyderabad
Ibrahimpatnam -501 510, Hyderabad, Telangana

PROGRAM OUTCOMES (POs)

PO1:	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2:	**Problem analysis**: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3:	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4:	**Conduct investigations of complex problems**: Use research−based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5:	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6:	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7:	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8:	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9:	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10:	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11:	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work,as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12:	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life−long learning in the broadest context of technological change.

DEPARTMENT OF INFORMATION TECHNOLOGY

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Accredited by NAAC, Accredited by NBA (UG Programmes: CSE & ECE)
Approved by AICTE, Affiliated to JNTUH Hyderabad
Ibrahimpatnam -501 510, Hyderabad, Telangana

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1:	**Foundation of mathematical concepts:** To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.
PSO2:	**Foundation of Computer System:** The ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.
PSO3:	**Foundations of Software development:** The ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process.Familiarity and practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research.

DEPARTMENT OF INFORMATION TECHNOLOGY

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Accredited by NAAC, Accredited by NBA (UG Programmes: CSE & ECE)
Approved by AICTE, Affiliated to JNTUH Hyderabad
Ibrahimpatnam -501 510, Hyderabad, Telangana

QUALITY OF THE PROJECT

I. Consideration to Factors

Factors (Environment, Safety, Ethics, Cost)	Type of Project (Application, Product, Research, Review, etc.)	Standards
This project has impact on the WHO based on public information.	This is a research based project which analyses the efficient results using the algorithms of machine learning	

II. POs and PSOs addressed through the project with justification

S.No.	POs and PSOs Addressed	Justification
1.	PO1	**Engineering knowledge:** Machine Learning algorithms were used for the comparison of results.
2.	PO2	**Problem analysis:** The main drawbacks were observed, and led to implementation of three machine−learning algorithms.
3.	PO3	**Design/Development of solutions:** We had designed the solution that gives the accurate values.
4.	PO5	**Modern Tool Usage:** We had implemented all the algorithms using modern engineering and IT tools (i.e., Python language and Anaconda navigator).
5.	PSO1	**Foundation of mathematical concepts:** Calculating the accuracy and precision are done based on these three algorithms.

6.	PSO3	**Foundation of Software Development:** This project has the proper usage of Software Development Life Cycle.

DECLARATION

We hereby declare that this Project Work titled “Deep Convolutional Neural Network for Detection of Cigarette Smokers in Public Places” is a genuine project work carried out by us, in B.Tech (Information Technology) degree course of Jawaharlal Nehru Technology University Hyderabad, Hyderabad and has not been submitted to any other course or university for the award of my degree by us.

	Candidate Name(s)	Roll Number	Signature
1.	HARSH NAGDA	20E11A1221	
2.	T. CHANDRAKANTH KUMAR GUPTHA	20E11A1223	
3.	MEDABOINA MALLIKARJUNA	20E11A1204	
4.	MALKAPURE KISHAN	20E11A1210	
5.	PATLA SIDDHARTHA	20E11A1230	

Date:

ABSTRACT

With the development of Internet technology and the improvement of network quality, online videos have become increasingly popular. In particular, online live broadcast has become a hotspot in recent years, and smoking behaviour in these broadcasts is harmful to smokers and the surrounding environment. Therefore, it is necessary to detect and thereby effectively control smoking behaviours in video content. Traditionally, smoking images are detected based on the detection algorithms of cigarette smoke. Given the limited resolution of live broadcast videos, cigarette smoke is not visually apparent in the video content. This paper proposes a smoking image detection model based on a convolutional neural network, referred to as SmokingNet, which automatically detects smoking behaviours in video content through images. This method can detect smoking images by utilizing only the information of human smoking gestures and cigarette image characteristics without requiring the detection of cigarette smoke, showing high accuracy and superior performance for real-time monitoring.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to **Sri CH. Venugopal Reddy**, Chairman & Secretary of BIET, for providing congenial atmosphere and encouragement.

We would like to thank Prof. **G. Kumaraswamy Rao**, Former Director & O.S. of DLRL Ministry of Defense, Sr. Director R&D, BIET, **Dr. V. Kannan**, College Admin Incharge of BIET and **Dr. T.V.Rama Krishna**, College Academic Incharge of BIET for having provided all the facilities and support.

We would like to thank **Dr. Karpaga Selvi S**, Admin Incharge of IT/DS/CS and **Dr. N. Srihari Rao**, Academic Incharge of IT for their expert guidance and encouragement at various levels of our project.

We are thankful to our Project Coordinator **Dr. N. Srihari Rao**, Professor in IT/DS/CS Dept. for his support and cooperation throughout the process of this project.

We are thankful to our guide **Ms. Falak Naaz**, Assistant Professor in CS Dept. for his sustained inspiring guidance and cooperation throughout the process of this project. His wise counsel and suggestions were invaluable.

We express our deep sense of gratitude and thanks to all the Teaching and Non-Teaching Staff of our college who stood with us during the project and helped us to make it a successful venture.

We place highest regards to our Parents, our Friends and Well-wishers who helped a lot in making the report of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SYMBOL	ABBREVIATION
MPL	Mobile Personal Live Cast
CNN	Convolutional Neural Network
DNN	Deep Neural Networks
RGB	Red Green Blue
NLP	Natural language processing
SRS	Software Requirement Specification
UML	Unified Modelling Language

1. INTRODUCTION

With the development of Internet technology and the improvement of network quality, online videos have become increasingly popular. In particular, online live broadcast has become a hotspot in recent years. Social apps such as Twitter and Facebook and mobile personal live cast (MPL) services have emerged and received much attention. With such social apps as Periscope and Facebook Live in the U.S. and Inke1 in China, numerous geo-distributed amateur broadcasters can broadcast their video content live to viewers around the world. It is well known that smoking behaviors are harmful to both smokers and the surrounding environment. Therefore, it is necessary to use images to automatically detect whether there are smoking behaviors in video content. Convolutional neural networks (CNNs) are a deep learning model. Here, "deep" indicates that, compared with shallow learning models, deep learning models involve neural networks with more hidden layers, and thus, the neural networks used for deep learning are called deep neural networks (DNNs). With the deepening of the research and the progress of computer hardware conditions, the number of layers of the deep learning models has increased from the initial value of to more than 100 nowadays. In this study, we design a CNN-based model called SmokingNet, which can automatically detect smoking behaviors in video content through images. Based on GoogleNet, the model optimizes the characteristics of smoking images. With non square convolution kernels, the model enhances the ability of feature extraction of the target images. Before model training, a super-large data set similar to the target image is used for pre-training the model. When the trained model is used to detect smoking images in the system, the full connection layers in the model are converted into convolution layers, which improves the detection ability of the model for local small targets while maintaining considerable detection efficiency.

2. LITERATURE SURVEY

In recent years, some researchers have proposed smoking image detection methods based on image recognition technology. Inoue et al. assigned eigen vectors to low dimensional spaces using subspace theory, and thereafter used feature clustering to classify cigarette smoke. Although this method can achieve smoking identification through smoke classification, the threshold in the algorithm is empirically set and its value will change with the background, leading to high false detection rates and poor applicability. Wu et al. proposed a method for detecting smoking images by first subtracting the current frame from the background obtained from a Gaussian mixture model to generate the foreground of the motion, subsequently using the shape and colour features to identify human hands, faces, and cigarettes in the foreground, combining the colour features to detect smoke, and finally conducting a comprehensive assessment of the relative positions of the human hands, faces, cigarettes, and smoke to identify whether smoking behaviours appear.

In the method proposed by Iwamoto et al., each frame of the video was first divided into several image blocks; subsequently, the blocks were subjected to comprehensive analysis based on image features such as gradient, variance, kurtosis, and skewness, and the blocks conforming to smoke characteristics were labelled; finally, the morphological changes and area changes of the labelled blocks of each frame in the entire video time domain were subjected to statistics to determine whether the marked areas contain smoke so as to detect smoking images. Odetallah et al. focused on the analysis of colour characteristics of cigarette smoke by performing background differentiation in the three channels of the RGB colour space so as to extract the foreground images conforming to colour characteristics of cigarette smoke, and subsequently conducted comprehensive analysis of the area changes and distance changes of human faces versus smoke-like images so as to detect smoking in the video. Bien et al. analysed indoor smoking images based on the recognition of human gestures, in which human hands and heads were first identified through skin colour along with a detection of their movements, and subsequently, a support vector machine was employed as a classifier for the probability characteristics of smoking behaviours, with the accuracy rate of classification reaching 83.33%.

Image recognition is the first breakthrough in deep learning. In 1995, LeCun et al., for the first time, used CNNs of deep learning models to successfully recognize handwritten digits. As

LeCun et al. incorporated image convolution operation into the learning models, the networks used by these models are called CNNs. In 2012, Krizhevsky et al. used an improved CNN (named AlexNet) to secure the first place in the ImageNet image recognition competition with an error rate ten percent lower than that of the model that secured the second place. AlexNet not only inherits the advantages of CNNs, but also overcomes their side effects. At present, the ReLU activation function, LRN operation, and dropout technology together with the convolutional layers and the pooling layers constitute the basic structure of CNNs. In 2013, Zeiler et al. won the ImageNet competition by adjusting the network structure through a visualization technology of CNNs. In 2014, Google introduced the GoogLeNet model and won the first place in the ImageNet competition that year. Since GoogLeNet, it has been accepted in the academic community that a further increase in the number of CNN layers can improve the recognition accuracy, but the increase would make sample training more difficult. In 2015, Srivastava et al. proposed highway networks , which allowed undecayed information flow across several layers through the gate mechanism so as to solve the problem of difficulty in training deep networks. He et al. proposed a residual neural network based on highway networks and increased the number of network layers to 152, winning first place in the 2015 ImageNet competition..

3. MOTIVATION

The escalating popularity of online videos, particularly in the dynamic landscape of live broadcasts, underscores the critical imperative to tackle detrimental behaviours like smoking within this burgeoning digital sphere. As social apps and mobile personal live cast services gain traction, the potential dissemination of content featuring smoking behaviours becomes a growing concern. This prevalence not only poses risks to individual health but also contributes to an adverse environmental impact. Recognizing the limitations of conventional detection methods, the motivation behind this research is to pioneer an innovative approach.

The core motivation lies in the aspiration to harness the power of deep learning, specifically through the implementation of convolutional neural networks (CNNs), to develop a cutting-edge model named SmokingNet. This model is meticulously crafted to address the unique challenges presented by the real-time and often low-resolution nature of live broadcast videos. Traditional algorithms reliant on detecting visible cigarette smoke may falter in this context, necessitating a paradigm shift toward a more nuanced and efficient solution.

The overarching goal is to create a tool that not only detects smoking behaviours within the multifaceted landscape of online live broadcasts but does so with a level of precision and adaptability that goes beyond the capabilities of existing methods. As online platforms continue to diversify globally, SmokingNet is envisioned as a universal solution capable of seamlessly integrating with various social apps and live broadcasting services. By aligning with the global variations in digital content consumption habits, this model seeks to provide a comprehensive and versatile solution to the pervasive issue of smoking in online videos.

In summary, the motivation encapsulates the urgent need to respond to the evolving dynamics of digital content consumption, where harmful behaviours like smoking can be magnified. By leveraging the advancements in deep learning and CNNs, the goal is to pioneer SmokingNet as a transformative tool, ultimately contributing to a safer and more responsible online viewing environment.

3.1 Objectives:

i. **Efficient Smoking Detection in Online Videos:** The primary objective is to design a robust smoking detection model that can operate effectively in the context of online live broadcasts, where video resolution may be limited. SmokingNet aims to overcome the challenges posed by traditional methods by focusing on human smoking gestures and cigarette image characteristics for detection.

ii. **Real-time Monitoring and Control:** The goal is to provide a solution for real-time monitoring of smoking behaviours within online video content. SmokingNet, leveraging CNNs, is designed to achieve high accuracy and superior performance, enabling timely interventions or alerts to prevent the propagation of harmful smoking behaviours.

iii. **Adaptability to Diverse Video Platforms:** The proposed model should be adaptable to various online video platforms, considering the emergence of different social apps and live broadcasting services globally. SmokingNet aims to provide a universal solution for detecting smoking behaviours, ensuring its applicability across different platforms and services.

iv. **Optimized Feature Extraction:** SmokingNet optimizes feature extraction by enhancing the capabilities of CNNs through nonsquare convolution kernels. This ensures the model's sensitivity to subtle smoking gestures, even in scenarios where traditional smoke visibility is limited.

v. **Reducing False Positives:** Addressing the limitations of existing smoking detection methods, SmokingNet seeks to minimize false positives by relying on human smoking gestures and precise cigarette image characteristics. This objective aims to enhance the reliability and applicability of the proposed model.

vi. **Pre-training for Enhanced Performance:** The use of a super-large dataset for pre-training is intended to improve the model's performance. SmokingNet aims to leverage a diverse dataset that mirrors the characteristics of target images, contributing to the model's ability to generalize and adapt to different scenarios

4. PROBLEM STATEMENT

4.1 Existing System:

Online live broadcast has become a hotspot in recent years. Social apps such as Twitter and Facebook and mobile personal livecast (MPL) services have emerged and received much attention. With such social apps as Periscope and Facebook Live in the U.S. and Inke1 in China, numerous geo-distributed amateur broadcasters can broadcast their video content live to viewers around the world. It is well known that smoking behaviours are harmful to both smokers and the surrounding environment. Therefore, it is necessary to use images to automatically detect whether there are smoking behaviours in video content.

In recent years, some researchers have proposed smoking image detection methods based on image recognition technology. Inoue et al. assigned eigenvectors to low dimensional spaces using subspace theory, and thereafter used feature clustering to classify cigarette smoke. Although this method can achieve smoking identification through smoke classification, the threshold in the algorithm is empirically set and its value will change with the background, leading to high false detection rates and poor applicability

4.2 Proposed System:

The CNNs in deep learning have been widely used in image detection. The features to be extracted through CNNs for image recognition no longer need to be defined manually, and the feature extraction is achieved via automatic fitting through training. Each convolution operation can be regarded as a process of feature extraction, in which the weights of the convolution kernels are not preset but are continuously updated through training until the model converges, when the weights constitute the optimal feature extraction scheme. SmokingNet, a detection model based on CNNs, optimizes the characteristics of smoking images based on GoogLeNet and enhances the ability of feature extraction of the target images using non-square convolution kernels. This model is pre-trained with a super-large data set similar to target images prior to model training, and the trained model is used to detect smoking images.

Convolutional neural networks (CNNs) are a deep learning model. Here, “deep” indicates that, compared with shallow learning models, deep learning models involve neural networks with more hidden layers, and thus, the neural networks used for deep learning are called deep neural networks (DNNs). With the deepening of the research and the progress of computer hardware conditions, the number of layers of the deep learning models has increased from the initial value of 6 to more than 100 nowadays In this study, we design a CNN-based model called SmokingNet, which can automatically detect smoking behaviours in video content through images. Based on GoogLeNet, the model optimizes the characteristics of smoking images. With nonsquare convolution kernels, the model enhances the ability of feature extraction of the target images. Before model training, a super-large data set similar to the target image is used for pre-training the model. When the trained model is used to detect smoking images in the system, the full connection layers in the model are converted into convolution layers, which improves the detection ability of the model for local small targets while maintaining considerable detection efficiency.

5. DESIGN AND METHODOLOGY

5.1 SYSTEM REQUIREMENTS SPECIFICATIONS:

The production of the requirements stage of the software development process is **Software Requirements Specifications (SRS)** (also called a **requirements document**). This report lays a foundation for software engineering activities and is constructing when entire requirements are elicited and analyzed. **SRS** is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectation of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

This is a communication between the clients and software designers/programmers. The specific goals are:

- Facilitating their views
- Information regarding the scope of work
- Providing a format/structure
- Providing frameworks for primary and secondary testing
- Platform for on-going refinement

5.1.1 HARDWARE REQUIREMENTS:

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum

5.1.2 SOFTWARE REQUIREMENTS:

- Programming Language / Platform : Python
- IDE : PyCharm/Jupiter

5.1.3 FUNCTIONAL REQUIREMENT:

Functional requirements are detailed specifications that outline the functionalities, features, and capabilities that a software system must possess to meet the needs of its users and fulfill the objectives of the business. In the context of the outlined data processing and machine learning workflow, functional requirements specify what the system should do at each step. These requirements provide a clear and comprehensive description of the expected behavior of the system, focusing on tasks, operations, and interactions within the software.

- Beginning with data loading, the system is expected to support the import of datasets in various formats, accommodating datasets of varying sizes.
- Data preprocessing entails handling null values through options such as record deletion or replacement with zero, column mean, mode, or median.
- Duplicate records are identified and deleted, and categorical values are processed using label encoding and one-hot encoding methods.
- Feature engineering involves outlier detection and removal, feature selection, feature scaling, and strategies for handling imbalanced datasets. Additionally, the system should provide tools or libraries for data visualization and define the mechanism for splitting datasets into training and testing sets.
- The machine learning model workflow involves creating, training, and evaluating models, with specified types of models and metrics for performance evaluation. Optimization techniques include hyperparameter selection and cross-validation strategies to enhance model robustness.

- The system is expected to support model deployment by providing methods for saving models, deploying them to web servers, and testing their functionality.
- Overall, these functional requirements lay the foundation for a comprehensive and robust data processing and machine learning system, addressing key aspects of dataset handling, model training, and deployment.

5.1.4 NON-FUNCTIONAL REQUIREMENTS:

Describe user-visible aspects of the system that are not directly related with the functional behaviour of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy (.e. how precise are the systems numerical answers.).

- Portability
- Reliability
- Usability
- Time Constraints
- Error messages
- Performance
- Standards
- Interoperability
- Scalability

5.2 PYTHON & ITS LIBRARIES:

5.2.1 Python Installation:

To install python the P.C the following steps are required.

Step 1: Initially, go to the website https://www.python.org/

Step 2: Click on the Python download for Windows

Step 3: Choose the python release that is suitable for PC and start downloading.

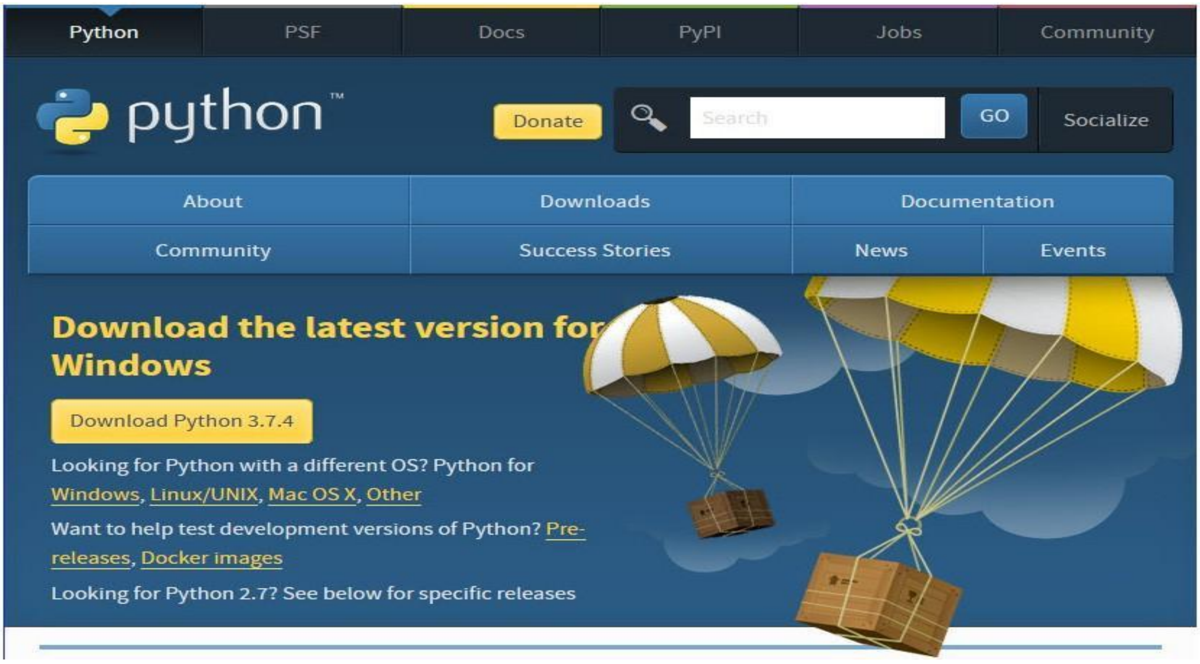

Fig 5.1 Python Download

Step 4: Install the python by double click the icon named python-3.7.4-amd64.exe

Fig 5.2 Installation

Step 5: Choose any one of the checkbox and then click on "Install Now", the following screens appears.

Fig 5.3 Setup

5.2.2 Introduction to Python:

Python, founded by Guido van Rossum in 1991, has evolved into a powerful and versatile programming language with distinct characteristics that set it apart in the software development landscape. Its syntax, designed for readability and simplicity, uses indentation to structure code blocks, enhancing the overall clarity of programs. As an interpreted language, Python offers agility in development, enabling programmers to test and modify code swiftly without the need for a separate compilation step. The language embraces dynamic typing, allowing variables to change types during runtime, promoting flexibility in coding practices.

Python is deeply rooted in object-oriented programming principles, providing robust support for encapsulation, inheritance, and polymorphism. This object-oriented approach fosters the creation of modular and maintainable code, contributing to the language's widespread adoption in diverse development scenarios.

One of Python's standout features is its extensive standard library, encompassing a wide range of modules and packages for various purposes. This comprehensive library includes tools for file I/O, regular expressions, networking, databases, and more. Additionally, Python's thriving community actively contributes to its ecosystem through the Python Package Index (PyPI), hosting a vast repository of third-party libraries and frameworks. This collaborative spirit ensures that Python remains at the forefront of technological advancements, catering to diverse needs such as web development, data science, machine learning, and artificial intelligence.

Python's cross-platform compatibility further enhances its appeal, allowing developers to write code on one platform and seamlessly execute it on others, including Windows, macOS, and Linux. The language's simplicity and readability make it an ideal choice for beginners, supported by comprehensive documentation, tutorials, and a robust community that readily shares knowledge and expertise.

Furthermore, Python's community-driven development model is exemplified by the Python Enhancement Proposal (PEP) process, through which users can propose and discuss improvements to the language. This collaborative governance ensures that Python continues to evolve, incorporating new features and enhancements while maintaining backward compatibility.

In summary, Python's multifaceted strengths, including its readability, versatility, extensive standard library, community support, and active development model, have propelled it to the forefront of programming languages. Its impact spans a wide spectrum of applications, from small-scale scripting to large-scale software development and complex scientific research, establishing Python as a cornerstone in the contemporary software development landscape.

A. Statements:

- For assigning a value to a variable, there is no need for specifying the data-type. The successive assignments of a common value to multiple variables can result in allocating the storage to names and object.
- The if statement will execute along with the else block and elif (i.e., else if) block.
- “import” statements are generally used for import modules whose variables/functions can be used in the present program. It can be specified by using anyone of 3ways:

  i. import <module> as [<formal name>]

  ii. from <module> import*

  iii. from <module> import <function1> [<formal name1>], <function2> [<formalname2>]

B. Expressions:

- The arithmetic operations perform same as like as other programming language but the only change is occurred during the division.

- In python there are 2 types of divisions:
  1 Floor division (//)
  2 Floating point division (/)

- From python 3.5 version, an infix operator that is denoted by @ is being used by the libraries of the NumPy for matrix multiplication.
- In python, + is used for concatenation of tuples and % for string format.

Note:

i. The operator that is being used for exponentiation is**.
ii. Tuples are immutable, can be denoted by ( ) and hence these can be used as keys of dictionaries.
iii. Lists are mutable, can be denoted by [ ] and hence these cannot be used as keys of dictionaries.

- Slice(:) will return a copy of entire tuple, list and each element is referred as a shallow copy. Slice will take elements from start index but does not include the stop index.

5.2.3. Libraries:

The most popular libraries which are being used by the developers for the implementation in the existing applications are

i. **NumPy :** The abbreviation is numerical python, it contains the basic linear algebra, Fourier transformations, etc. The main feature is Array interface, which can be utilized for expressing the images and binary row streams as an array containing real numbers.

ii. **Keras** : It is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

iii. **Pandas:** This library mainly provides the data structures of high level and variety of tools for the analysing . It performs the structured data operations and manipulations. The operations that are performed: Slicing, Data Munging, Concatenation, Changing the index, Changing the column headers.

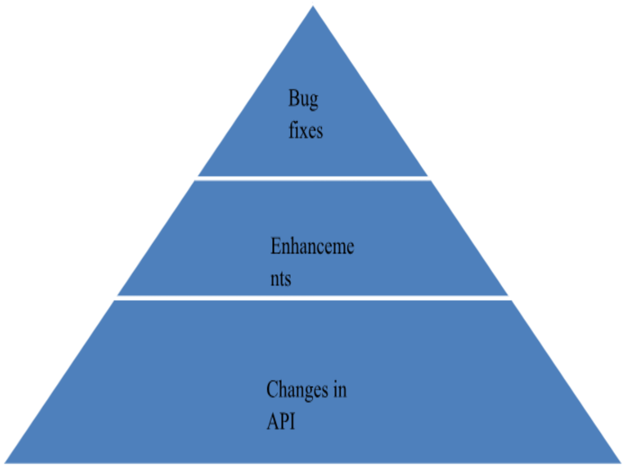

Fig 5.4 Pandas Features

iv. **matplotlib:** This is a plotting library that is used for 2D graphics in python programming language. It is even used in python scripts, shell, web application servers and other GUI toolkits. The main disadvantages are this module is heavily reliant on other packages and only works for python but not any other high level programming languages.

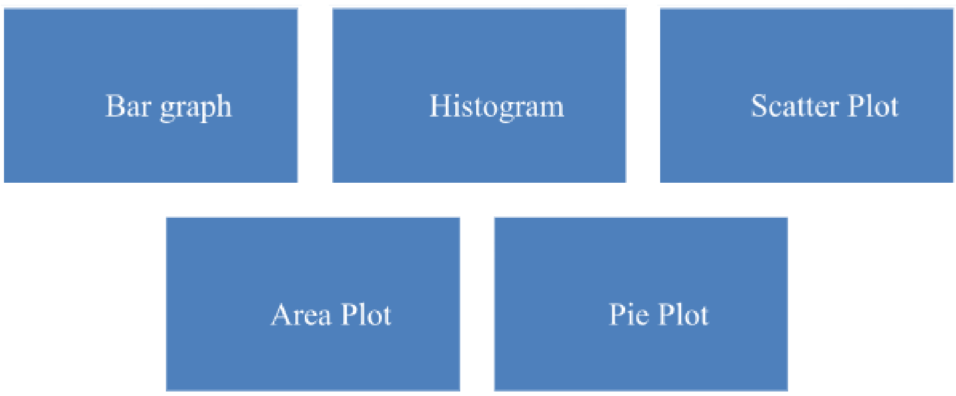

Fig 5.5. Types of Matplotlib

v. **TensorFlow**: It is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

vi. **PyTorch**: is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports on Computer Vision, Natural Language Processing (NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

vii. **Fastai**: It is a deep learning library that provides high-level components which can quickly and easily provide state-of-the-art results in standard deep learning domains. It also provides researchers with low-level components that can be mixed and matched to build new approaches. fastai includes various features, such as a GPU-optimized computer vision library which can be extended in pure Python, a new type dispatch system for Python.

viii. **Imutils** : It is a computer vision package that includes a series of OpenCV + convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, among others quite easy.

ix. **OpenCV** is a popular and open-source computer vision library that is focused on real-time applications. The library has a modular structure and includes several hundreds of computer vision algorithms. OpenCV includes a number of modules including image processing, video analysis, 2D feature framework, object detection, camera calibration, 3D reconstruction and more.

x. **scikit-learn:** Generally, it is a python library associated with both Numpy and SciPy that works with complex data. This tool is mainly used for classification, regression, clustering and dimensionality reduction. The name is given as "sklearn" that is used for importing the modules from it.

5.3 ANACONDA (A PYTHON DISTRIBUTION):

5.3.1 Installation of Anaconda:

Step1: Go to anaconda navigator website for downloading.

https://www.anaconda.com/distribution/

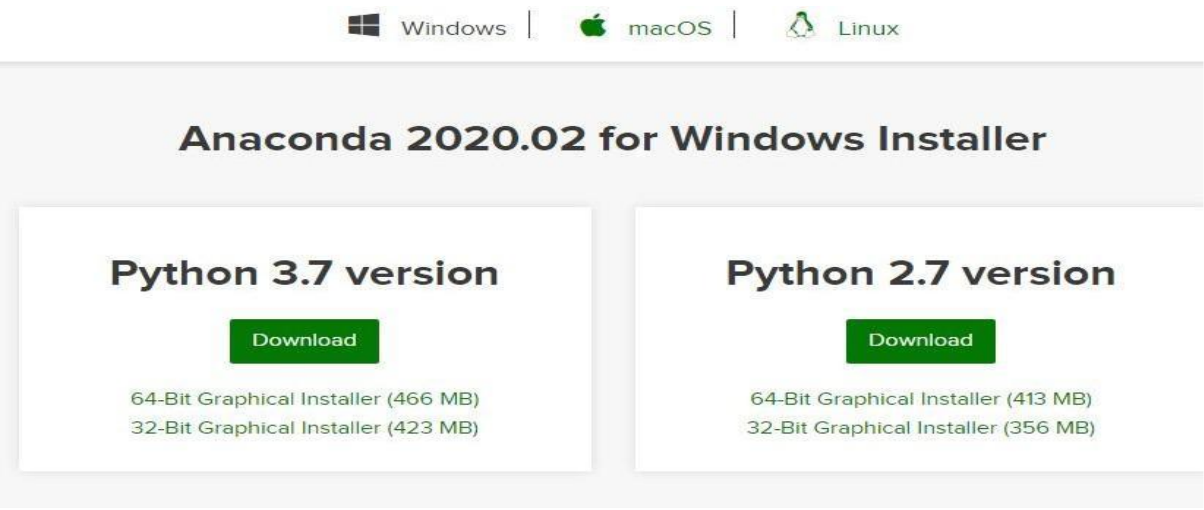

Fig 5.6 Anaconda Navigator Installer

Step 2: Click on Anaconda3 that is available in Downloads.

Step 3: Click on Next and then accept the terms and conditions. After which a popup will come to select the path and the options will be displayed for installation.

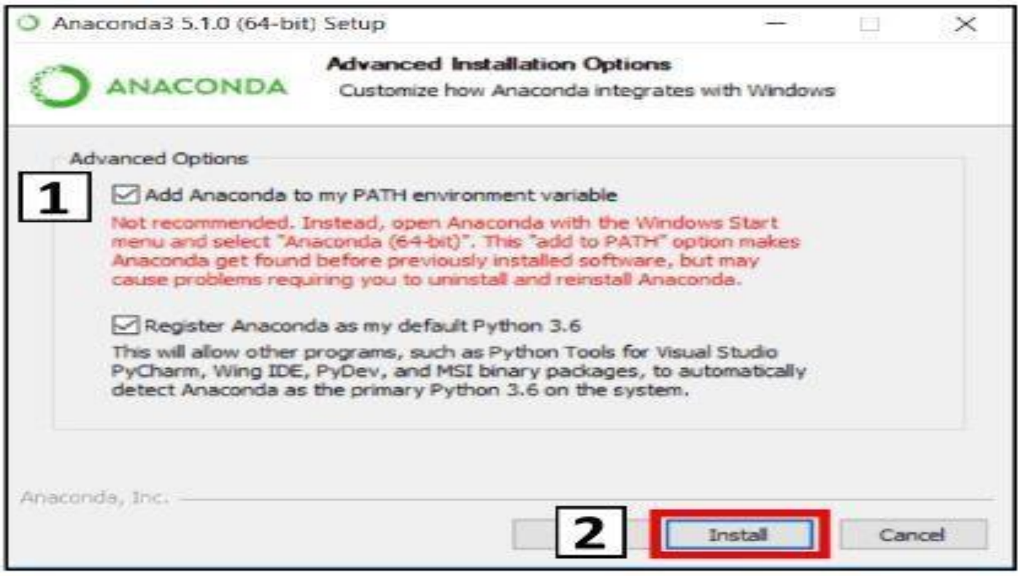

Fig 5.7 Installation options

Step 4: Click on install button and hence the following screens will be displayed.

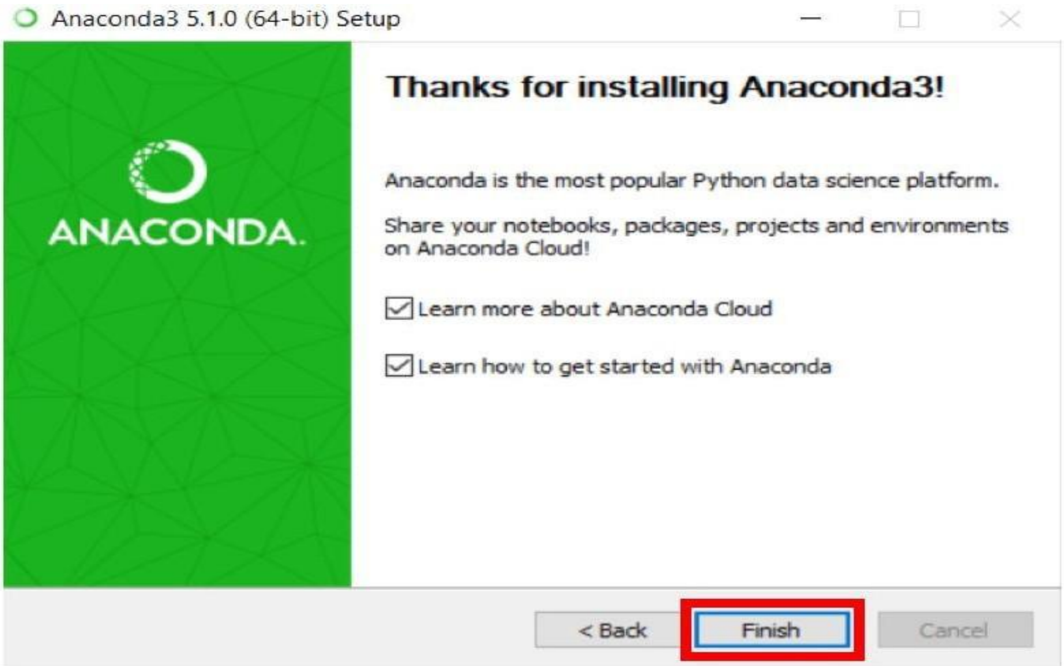

Fig 5.8 Installation Successful

5.3.2 Introduction:

This is a free and open-source distribution for R and Python programming languages for computing the scientific calculations that aims for the simplification of package management and deployment. The "conda" which is a package management system is responsible for the managing the versions. This distribution includes the data science packages, which are suitable for MacOS, Linux and Windows, and hence it comes up with 1500 packages that were selected from PyPI, virtual environment manager and conda. Basically it has 2 types of managers:

i. **pip (preferred installer program):**

When a package is being installed with pip, then it automatically installs any dependent python package without noticing the previous installed packages.

ii. **conda:**

This manager will carefully analyse the present environment that includes the current and previously installed packages, then installs the missing one. The following command is used for installation: **conda install.**

He available packages on PyPI can be installed into the environment of conda using "pip" and the conda will maintain a track on what is being installed. The default installation will include python.

A Desktop GUI (graphical user interface) which allows to launch the applications and also maintains the packages, environments without any command line commands is Anaconda navigator.

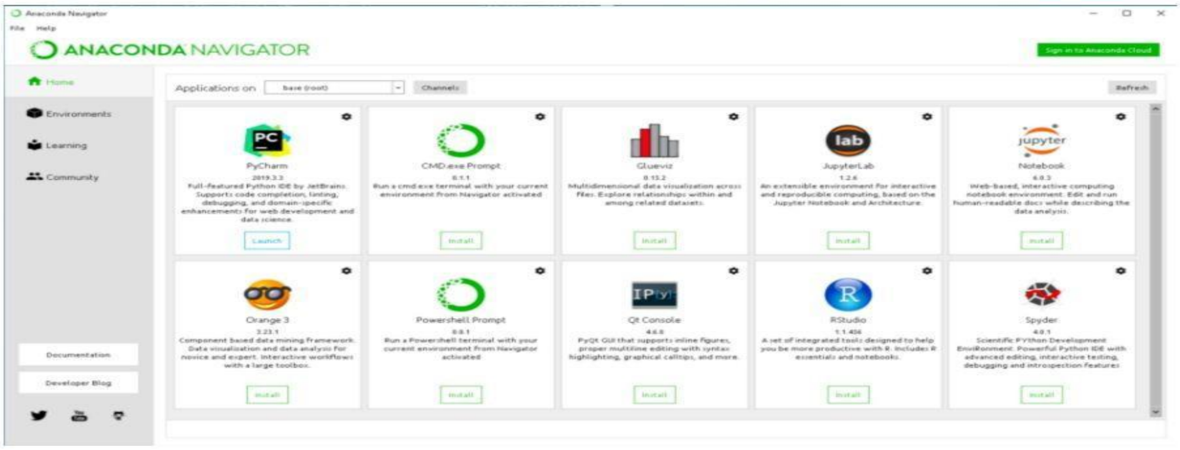

Fig 5.9 Anaconda Home Page

5.4 System Design:

i. Dataflow Diagram:

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

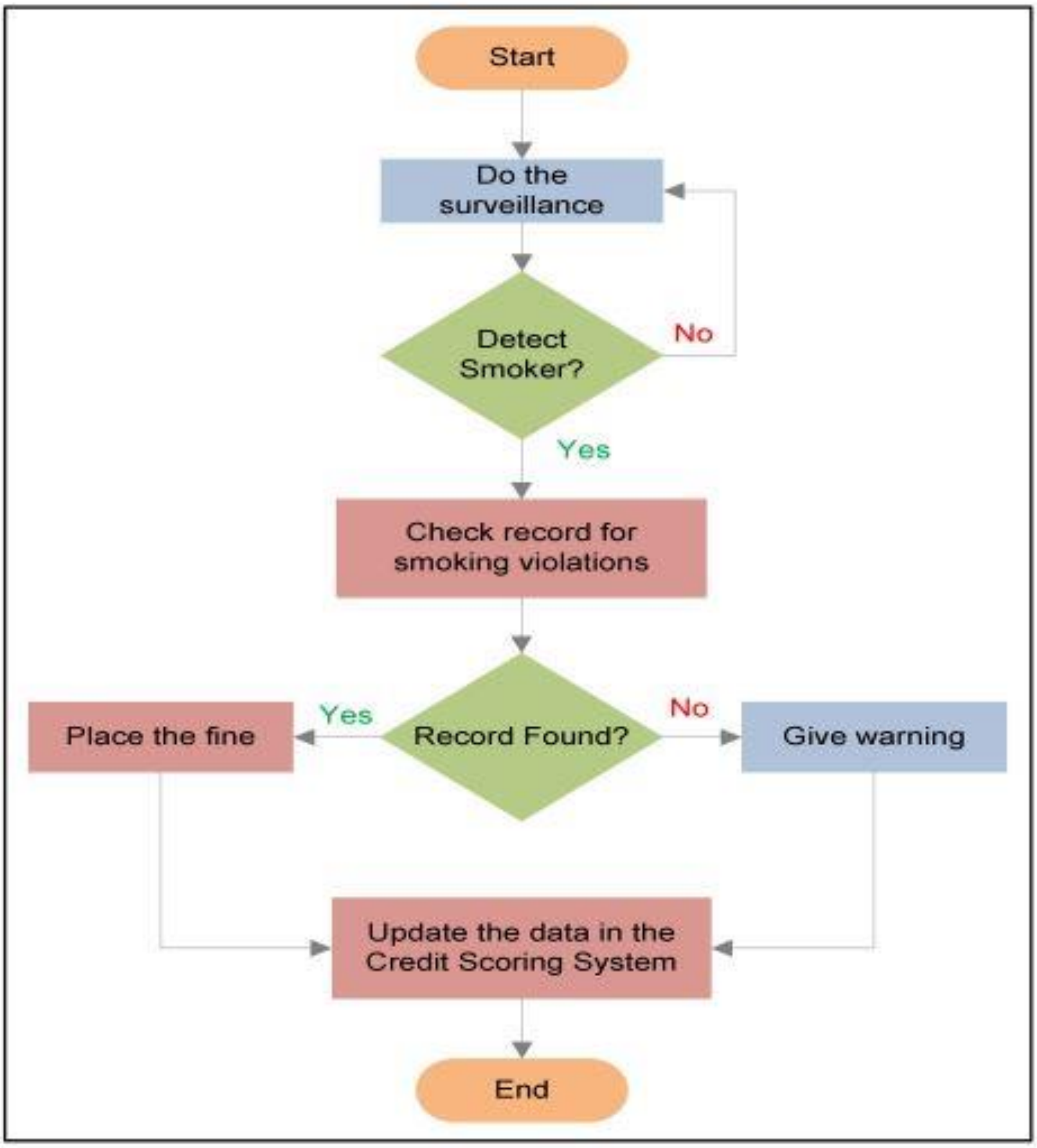

Fig 5.10 Dataflow Diagram

ii. System Architecture:

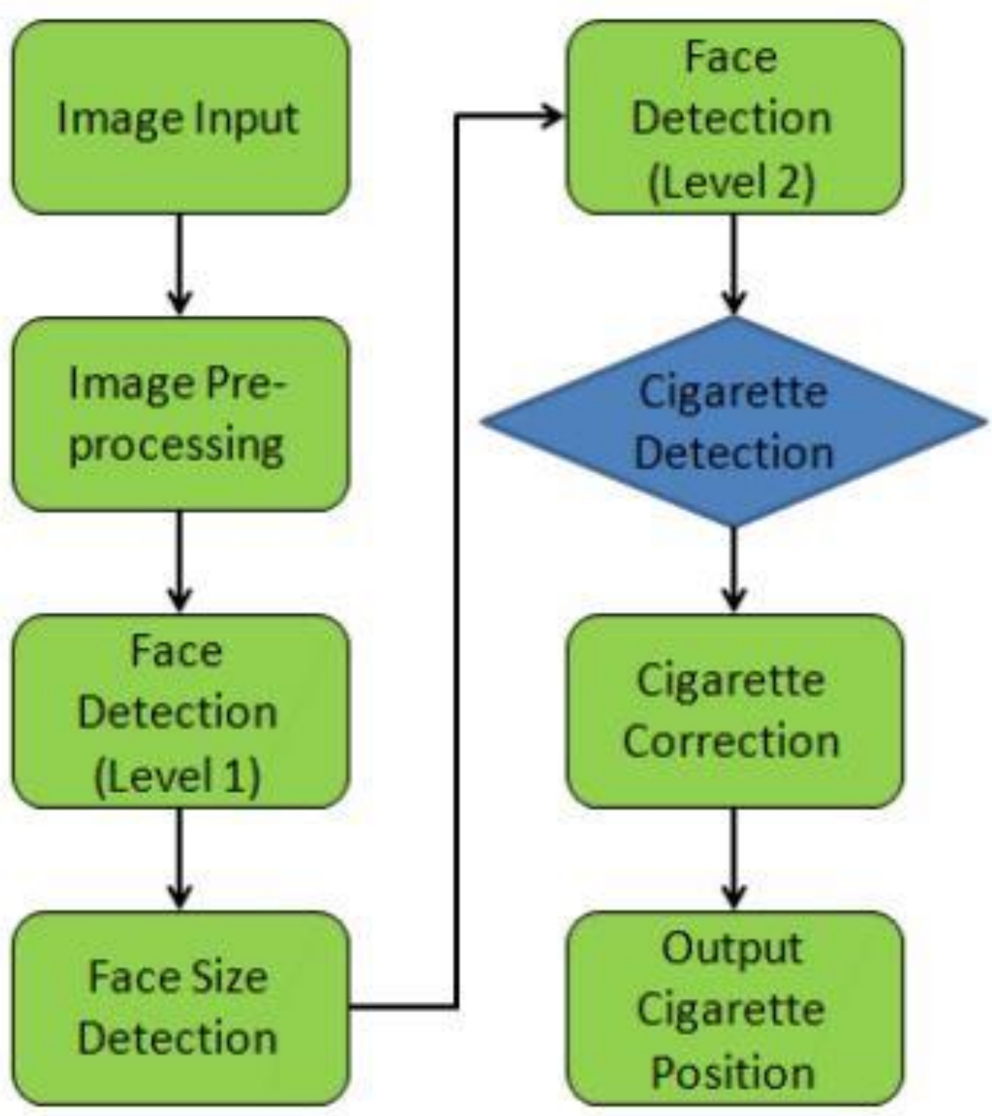

Fig 5.11 System Architecture

iii. Technical Architecture:

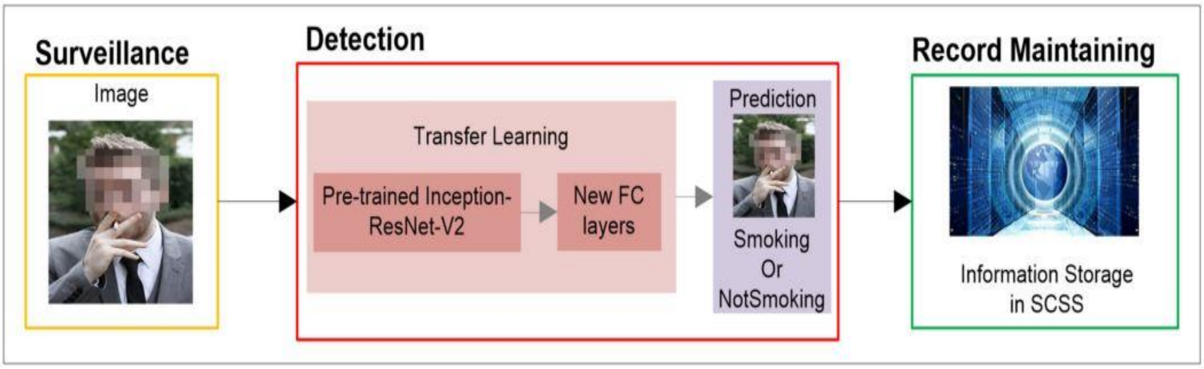

Fig 5.12 Technical Architecture

iv. UML DIAGRAMS

Unified Modelling Language (UML) is a modelling language. The main purpose of UML is to visualize the way a system has been designed. It is a visual language to sketch the behavior and structure of the system. This was adopted by Object Management Group (OMG) as a standard in 1997.

a. Use Case Diagram:

- The purpose of use case diagram is to capture the dynamic aspect of a system. This is used to gather the requirements of a system including internal and external influences.
- 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.
- The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

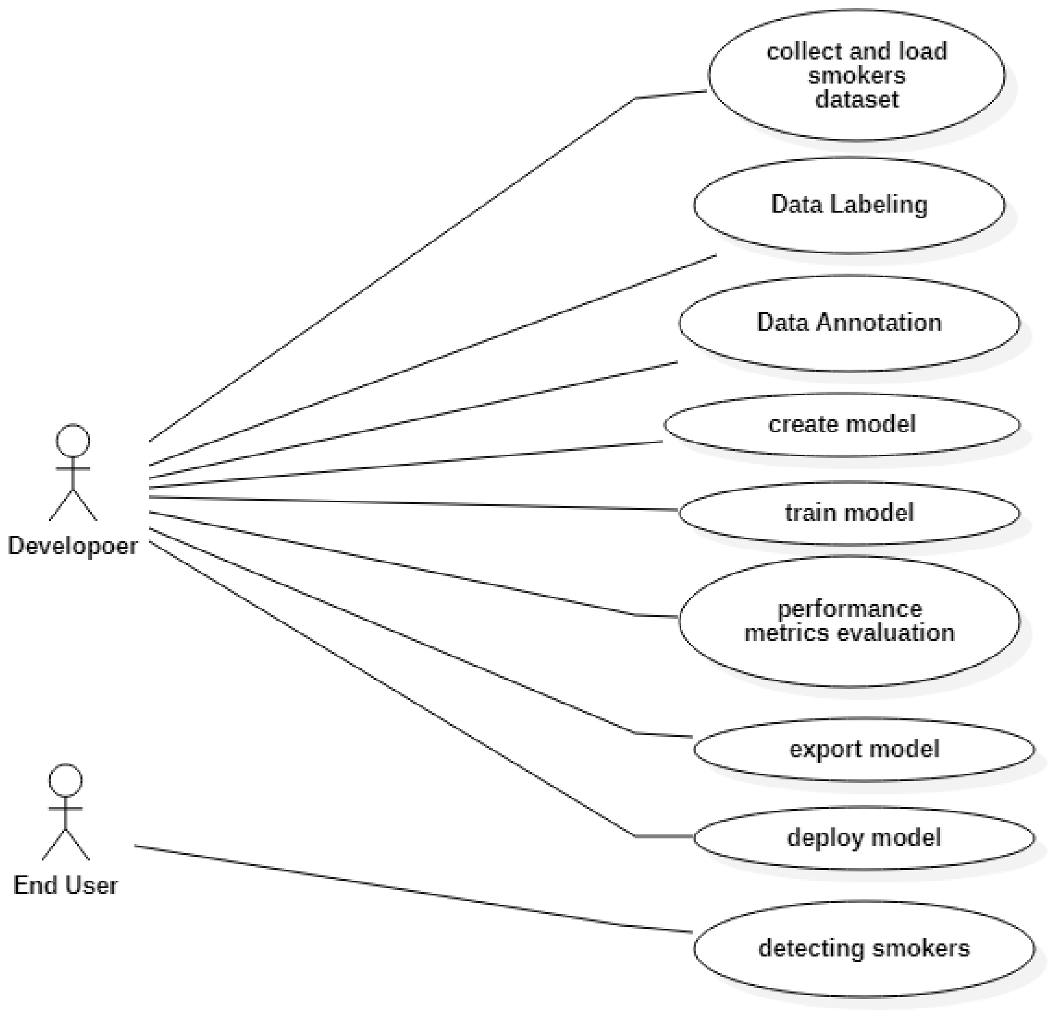

Fig 5.13 Use Case Diagram

b. Sequence Diagram:

- A sequence diagram details the interaction between objects in a sequential order i.e. the order in which these interactions take place.
- This diagrams sometimes known as event diagrams or event scenarios. This helps in understanding how the objects and component interacts to execute the process.
- This has two dimensions which represents time (Vertical) and different objects (Horizontal)

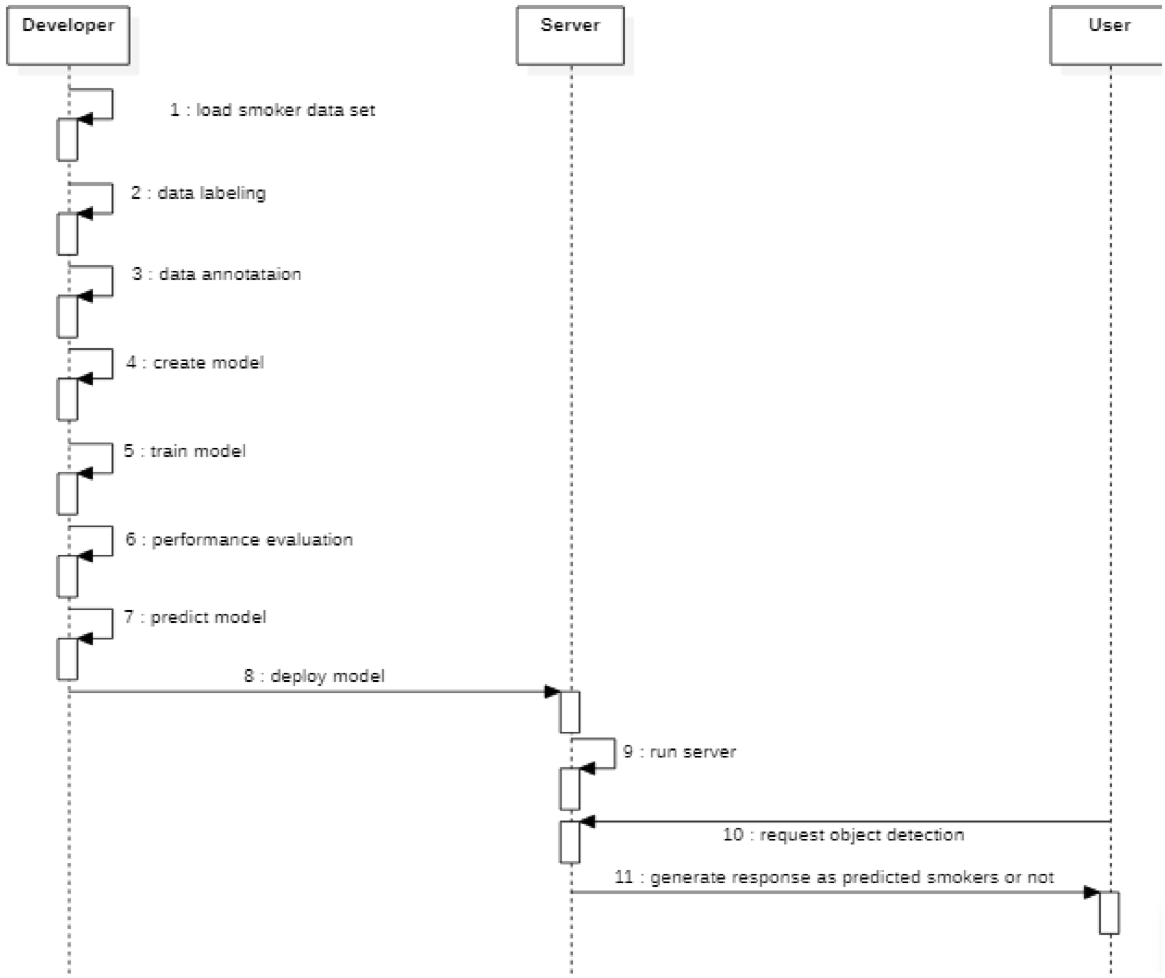

Fig 5.14 Sequence Diagram

c. Activity Diagram:

- It is behavioral diagram which reveals the behavior of a system. it sketches the control flow from initiation point to a finish point showing the several decision paths that exist while the activity is being executed.
- This doesn't show any message flow from one activity to another, it is sometimes treated as the flowchart. Despite they look like a flowchart, they are not.
- In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system..

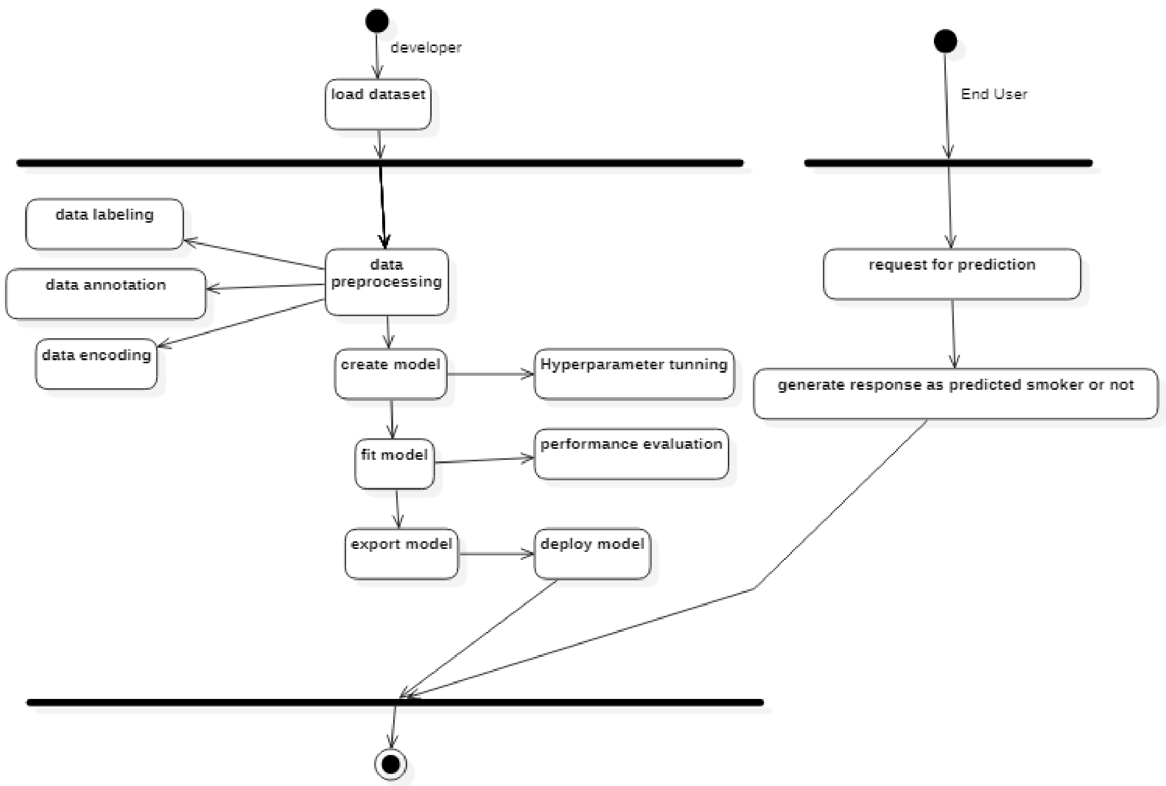

Fig 5.15 Activity Diagram

d. Class Diagram:

- The class diagram describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among the classes.
- It explains which class contains information and also describes responsibilities of the system. This is also known as structural diagram.

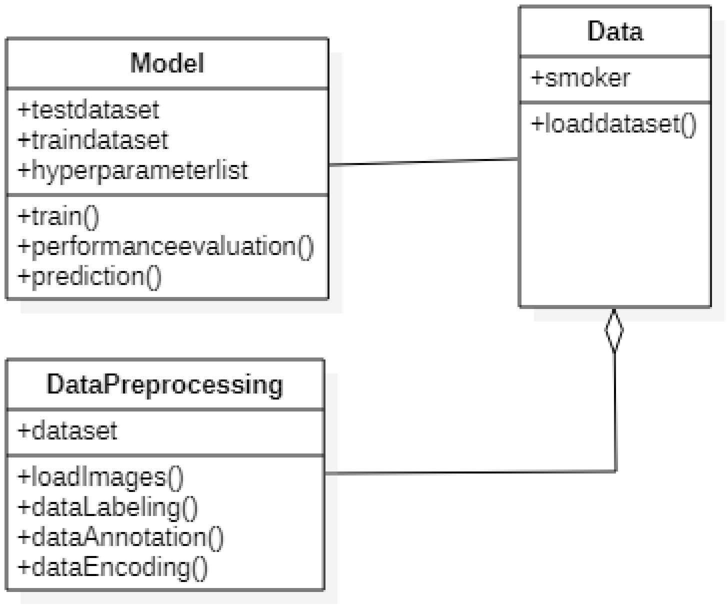

Fig 5.16 Class Diagram

e. Deployment diagram

There may be more steps involved, depending on what specific requirements you have, but below are some of the main steps:

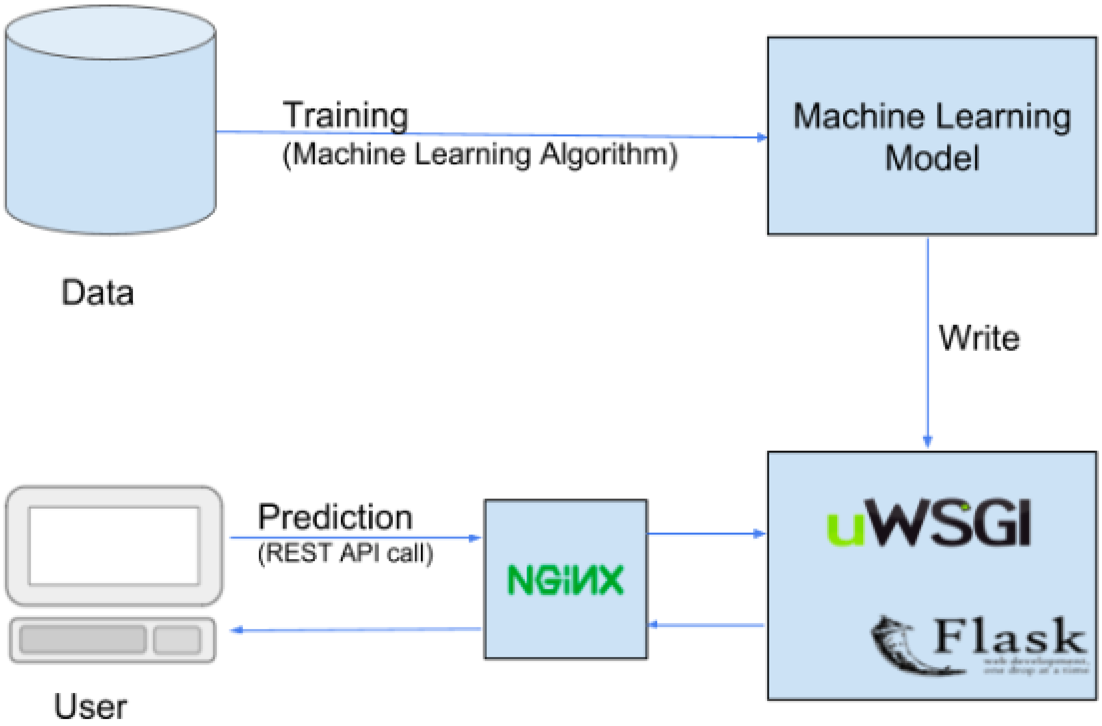

Fig 5.17 Deployment diagram

5.5 Algorithm:

SmokingNet Structures The convolution kernels of the CNN convolutional layers have been used to extract local features of a given image, and the features extracted by the first convolutional layer directly affect the feature fusion of the deep network. In most cases, the convolution kernels of a classical CNN model are squares[9,11,12], but, as cigarettes are strip-shaped, the square convolution kernels are not suitable for extracting the shape characteristics of cigarettes. Based on the shape characteristics of cigarettes, convolution kernels of four sizes are included in the first convolutional layer of SmokingNet, as shown in Figure 5.10. affect the feature fusion of the deep network. In most cases, the convolution kernels of a classical CNN model are squares[9,11,12], but, as cigarettes are strip shaped, the square convolution kernels are not suitable for extracting the shape characteristics of cigarettes. Based on the shape characteristics of cigarettes, convolution kernels of four sizes are included in the first convolutional layer of SmokingNet, as shown in Figure 5.10.As shown in Figure 5.10(a), if only a small convolution kernel of 3*3 pixels is used, the cigarette-like part of the positive sample background (as indicated by the black box) will interfere with feature extraction; if a large convolution kernel of 5*5 pixels as shown in Figure 5.10(b) is used, it is impossible to extract fine edge features; similarly, although the convolution kernels in Figure 5.10(c)(d) can well extract cigarette features in the horizontal and vertical directions, they cannot deal with other angles at which the cigarette may be positioned. Therefore, SmokingNet simultaneously uses these four convolution kernels in the first layer to enhance the capability of CNNs for feature extraction, and in the second layer, fuses the feature graphs generated by different convolution kernels.

SmokingNet Training Methods For CNNs, the training data used in this test task are still not sufficient in quantity. If the model is trained by directly using randomized network weights, over-fitting is likely to occur, and therefore, it is necessary to conduct fine-tuning of the model after it has been subjected to pre-training based on a super-large data set so as to improve network performance Generally, the closer the image content of the pre-training data set is to the detection target, the better the training effect, and hence, SmokingNet does not use the commonly-used ImageNet data sets for pre-training but selects 1mHand a trained model developed by Koller et al. to identify sign language as the initial weight of the network, owing to the following three major reasons: first, the training samples of 1mHand contain and only contain human hands, trunks, and faces, and are thus similar to the cigarette-free background images of the positive smoking samples in which the parts in contact with cigarettes are all hands or mouths; second, 1mHand uses a super-large data set of more than 1 million data, which can improve the generalization ability of the model ; third, 1mHand directly uses GoogLeNet for training, and moreover, given that the overall structure of SmokingNet is similar to that of GoogLeNet, significant pre-training time can be saved, and fine-tuning can be initiated by directly using the network weight data file under the open Caffe framework of GoogLeNet.

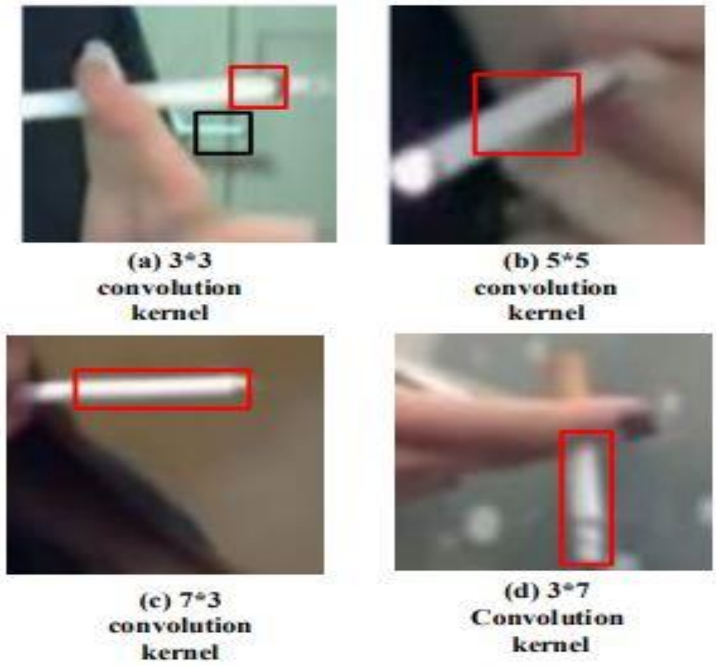

Fig 5.18 SmokingNet Structures

5.6 Modules

i. Image Acquisition

The first step of the Smoking Detection system is image acquisition. High-quality Human Smoking images need to collection from public places.

The entire sample set is divided into three parts: training samples and validation samples in the training phase and testing samples in the testing phase. Moreover, the sample set is divided into positive and negative samples—a positive sample is an image showing smoking behaviors, whereas a negative sample is a background image.

ii. Annotated Dataset Collection

A Knowledge-based dataset is created by proper labelling of the collected images with unique classes.

iii. Image Processing

The obtained images that will be engaged in a preprocessing step are further enhanced specifically for image features during processing. The segmentation process divides the images into several segments and utilized in the extraction of Smoking features from dataset.

iv. Feature-Extraction

This section involves the coevolutionary layers that obtain image features from the resize images and is also joined after each convolution with the ReLU. Max and average pooling of the feature extraction decreases the size. Ultimately, both the convolutional and the pooling layers act as purifiers to generate those image characteristics.

v. Classification

The final step is to classify images, to train deep learning models along with the labelled images to be trained on how to recognize and classify images according to learned visual patterns. The authors used an open-source implementation via the TensorFlow module, using Python and OpenCV including the VGG-16 CNN model.

5.7 Algorithm Implementation

The CNNs in deep learning have been widely used in image detection. The features to be extracted through CNNs for image recognition no longer need to be defined manually, and the feature extraction is achieved via automatic fitting through training. Each convolution operation can be regarded as a process of feature extraction, in which the weights of the convolution kernels are not preset but are continuously updated through training until the model converges, when the weights constitute the optimal feature extraction scheme. SmokingNet, a detection model based on CNNs, optimizes the characteristics of smoking images based on GoogLeNet and enhances the ability of feature extraction of the target images using non-square convolution kernels. This model is pre-trained with a super-large data set similar to target images prior to model training, and the trained model is used to detect smoking images. A. Training Samples and Testing Samples In the detection of smoking images based on CNNs, the entire sample set is divided into three parts: training samples and validation samples in the training phase and testing samples in the testing phase. Moreover, the sample set is divided into positive and negative samples—a positive sample is an image showing smoking behaviors, whereas a negative sample is a background image.

Positive samples Positive samples are collected from online smoking videos and the smoking videos made by our research group. Positive samples are acquired for training by playing these videos frame by frame. When a smoking frame appears, the image containing the complete cigarette is manually captured using a screenshot tool. The screenshot tool is specifically developed for this collection task based on the computer vision library OpenCV. The screenshot tool contains a candidate box which can be manually adjusted in size and moved to various positions with the mouse. When the left mouse button is pressed, the tool will save the local image in the candidate box to the local disk. To achieve sample expansion from a limited number of video images, the tool automatically performs three image transformations when capturing an image—horizontal mirror transformation, 45-degree clockwise rotation, and 45-degree counterclockwise rotation—generating local images at the same location of the original image and saving all of them to the disk, which indicates that a screenshot operation generates four images simultaneously.

In addition, as all image samples are to be scaled to squares of the same size during the training phase, the candidate frame of the screenshot tool is always square, i.e., the saved training samples are all images with an aspect ratio of 1:1, so that image stretching can be avoided during the training phase to ensure the quality of the training samples.

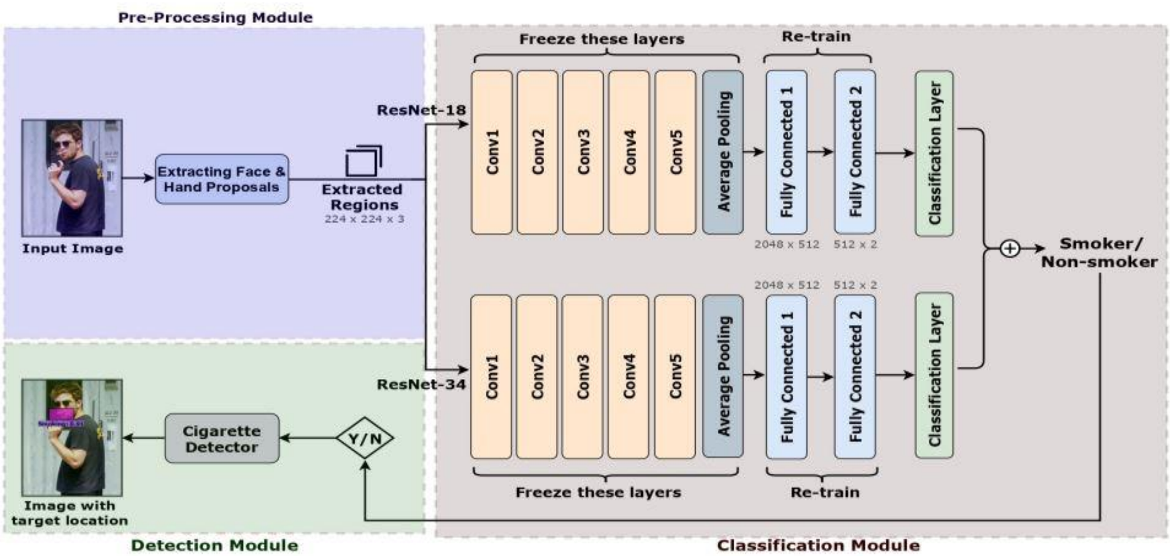

FIG 5.19 IMPLEMENTION

6. CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In this study, we design and implement a deep learning model, i.e., SmokingNet, which is specially optimized for smoking images. In addition, we conduct large-scale model training and testing sample collection with respect to smoking as a specific detection target. Based on GoogLeNet, the detection accuracy of smoking images is enhanced by improving the network structure and using special convolution layers to better extract the characteristics of smoking images. In addition, specific pre-training models are selected for SmokingNet based on detailed analysis of the characteristics of smoking images, and subsequently, the training parameters and training process are elaborated. Finally, the detection performance of SmokingNet is tested by comparison with those of other deep and shallow learning models. The experimental results show that, compared with the classical deep learning models, SmokingNet shows significantly improved detection performance with precision and recall over one percent higher than those of the second best model and with a detection efficiency as high as 80 FPS, indicating that SmokingNet is fully capable of achieving real-time detection of smoking images during live webcast.

6.2 FUTURE SCOPE

The future scope of a project like SmokingNet, which focuses on deep learning-based smoking detection, is broad and can extend into various directions. Here are several potential future scopes for the project:

a. Expanding to Other Harmful Activities:
Extend the model to detect and classify other harmful activities, such as drug use or violence. This would broaden the applicability of the system in ensuring safety and security.

b. Multimodal Detection:
Integrate additional sensory inputs, such as audio or sensor data, to create a multimodal

detection system. Combining multiple modalities can enhance accuracy and robustness in diverse environments.

c. Real-time Intervention Systems:

Develop real-time intervention systems that can automatically alert authorities or security personnel when smoking or other harmful activities are detected. This could be integrated into surveillance systems for rapid response.

d. Mobile and Edge Computing:

Optimize the model for deployment on edge devices or mobile platforms. This would enable on-device processing, making the system more versatile and capable of functioning in remote or resource-constrained environments.

e. Behavioural Analysis:

Move beyond static image analysis and incorporate dynamic behavioral analysis. Understand patterns of movement and interactions to improve the system's ability to recognize and respond to specific behaviours.

f. Privacy-Preserving Solutions:

Explore techniques for privacy-preserving deep learning. Implement methods to ensure that the system can detect harmful activities without compromising the privacy of individuals.

g. Integration with Smart Cities:

Collaborate with smart city initiatives to integrate SmokingNet into smart city infrastructure. This could contribute to the creation of safer public spaces and enhance overall city security.

h. Continuous Learning and Adaptation:

Implement systems for continuous learning and adaptation. The model could be designed to learn from new data continuously, allowing it to adapt to changing behaviors and maintain effectiveness over time.

i. Global Collaboration:

Collaborate with researchers, organizations, and government agencies globally to standardize and share datasets. This could lead to the development of a more universally applicable model and contribute to a global effort to enhance public safety.

j. Commercial and Industrial Applications:

Explore commercial and industrial applications, such as monitoring workplaces for safety compliance. The technology could be adapted for use in manufacturing plants or other settings where specific behaviors pose risks.

k. Humanitarian and Public Health Initiatives:

Consider the potential humanitarian applications, such as monitoring public spaces for compliance with anti-smoking regulations. This could contribute to public health initiatives aimed at reducing the prevalence of smoking.

l. Ethical AI Development:

Place a strong emphasis on ethical AI development. Ensure fairness, transparency, and accountability in the deployment of the system, addressing potential biases and social implications.

REFERENCES

[1]. Swapnil Dhanwal, Vishnu Bhaskar, and Tanya Agarwal, "Automated Censoring Of Cigarettes In Videos Using Deep Learning Techniques," in Book: Decision Analytics Applications in Industry, Springer, pp. 339–348. 2020.

[2]. Joseph P Stitt and Lynn T Kozlowski, "A System for Automatic Quantification of Cigarette Smoking Behaviour," in IEEE International Conference of Engineering in Medicine and Biology Society (EMBS), 2006, pp. 4771–4774.

[3]. Tzu-Chih Chien, Chieh-Chuan Lin, and Chih-Peng Fan, "Deep Learning Based Driver Smoking Behaviour Detection For Driving Safety," Journal of Image and Graphics, vol. 8, no. 1, 2020.

[4]. Volkan Y Senyurek, Masudul H Imtiaz, Prajakta Belsare, Stephen Tiffany, and Edward Sazonov, "A CNNLSTM Neural Network for Recognition of Puffing in Smoking Episodes using Wearable Sensors," Springer Biomedical Engineering Letters, pp. 1–9, 2020.

[5]. Peng Mao, Kunlun Zhang, and Da Liang, "Driver Distraction Behaviour Detection Method Based On Deep Learning," Journal of Management Science and Engineering (JMSE), vol. 782, no. 2, pp. 022012, 2020.

[6]. Masudul H Imtiaz, Volkan Y Senyurek, Prajakta Belsare, Stephen Tiffany, and Edward Sazonov, "Objective Detection Of Cigarette Smoking From Physiological Sensor Signals," in IEEE Annual International Conference of the Engineering in Medicine and Biology Society (EMBS), 2019, pp. 3563–3566.

[7]. Mingqi Lu, Yaocong Hu, and Xiaobo Lu, "Driver Action Recognition using Deformable and Dilated Faster R-CNN with Optimized Region Proposals," Springer Applied Intelligence Journal, vol. 50, no. 4, pp. 1100– 1111, 2020.

[8]. Ali Khan, "Dataset Containing Smoking and Not-Smoking Images (Smoker vs Non-Smoker)," https://data.mendeley.com/datasets/7b52hhzs3r/1 , 2020, Accessed on 06.01.2021.

[9]. Ivan Itzcovich, "Faced: Algorithm for Detecting Face Regions," https://github.com/iitzco/faced , 2018, Accessed on 06.01.2021.

[10]. Tzutalin Darrenl, "Labelimg: A Graphical Image Annotation Tool," https://github.com/tzutalin/labelImg , 2017, Accessed on 06.01.2021.