Industry Oriented Mini Project Report
on

Youtube Spam Comment Detection

Submitted in partial fulfillment of the requirements
for the award of degree of

BACHELOR OF TECHNOLOGY
in
Information Technology
by

Vardha Pranathi (20WH1A1217)
P.Poornika (20WH1A1236)
K.Charishma (20WH1A1259)

Under the esteemed guidance of

Mr. Ch.Anil Kumar
Assistant Professor

Department of Information Technology
BVRIT HYDERABAD College of Engineering for Women
Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyd–500090
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
(NAAC ‘A’ Grade & NBA Accredited- ECE, EEE, CSE & IT)

November 25, 2023

BVRIT HYDERABAD

College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE & IT)

CERTIFICATE

This is to certify that the Project report on **"YOUTUBE SPAM COMMENT DETECTION"** is a bonafide work carried out by **Vardha Pranathi (20WH1A1217), P. Poornika (20WH1A1236) and K. Charishma (20WH1A1259)** in the partial fulfillment for the award of B.Tech degree in **Information Technology , BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad** affiliated to Jawaharlal Nehru Technological University, Hyderabad, under my guidance and supervision. The results embodied in the project work have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide
Ch. Anil Kumar
Assistant Professor
Department of IT

Head of the Department
Dr. Aruna Rao S L
Professor & HoD
Department of IT

External Examiner

DECLARATION

We hereby declare that the work presented in this project entitled **"Youtube Spam Comment Detection"** submitted towards completion of in IV year I sem of B.Tech IT at "BVRIT HYDERABAD College of Engineering for Women",Hyderabd is an authentic record of our original work carried out under the esteemed guidance of **ch. Anil Kumar, Assistant Professor**, Department of Information Technology.

Vardha Pranathi (20WH1A1217)

K. Charishma (20WH1A1259)

P. Poornika (20WH1A1236)

ACKNOWLEDGMENTS

We would like to express our profound gratitude and thanks to **Dr. K. V. N. Sunitha, Principal, BVRIT HYDERABAD College of Engineering for Women** for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. Aruna Rao S L, Professor & Head, Department of IT, BVRIT HYDERABAD College of Engineering for Women** for all the timely support, constant guidance and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Ch. Anil Kumar, Assistant Professor, Department of IT, BVRIT HYDERABAD College of Engineering for Women** for her constant guidance, encouragement, and moral support throughout the project.

Finally, we would also like to thank our Project Coordinators **K.S Niraja, Assistant Professor and A. Aruna Jyothi, Assistant Professor**, all the faculty and staff of Department of IT who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

Vardha Pranathi (20WH1A1217)

K. Charishma (20WH1A1259)

P. Poornika (20WH1A1236)

ABSTRACT

YouTube is one of the most popular video-sharing platforms in the world, with millions of videos uploaded and billions of views generated every day. Unfortunately, this popularity has also attracted spammers, who use the platform to post irrelevant or promotional comments. These spam comments can significantly impact the quality and credibility of the content and the platform as a whole .This paper presents a novel approach to detect spam comments on YouTube through a multifaceted methodology. The proposed method transforms comments into numerical featuresusing TF-IDF. This numerical representation facilitates the utilization of two machine learning models Multinomial Naive Bayes and Support Vector Machine (SVM), both trained on the preprocessed data. By leveraging these machine learning models, the approach aims to enhance the platform's ability to discern and combat spam effectively. The integration of TF-IDF text vectorization and machine learning models enhances the platform's capability to discern and filter out spam effectively.

Keywords: YouTube, spam comments, video-sharing platform, TF-IDF, text vectorization, machine learning, Multinomial Naive Bayes, Support Vector Machine (SVM), content quality, credibility, spam detection, user experience.

Contents

List of Figures

Chapter 1

Introduction

In recent years, informal online communities such as Facebook and YouTube have become increasingly prevalent platforms in individuals' daily lives. People utilize social media not only to stay connected with friends and family but also to share thoughts and ideas through blogs. This growing trend has attracted a significant number of users to these platforms, making them susceptible targets for spammers. YouTube, in particular, has emerged as the most popular social network among the younger demographic. Beauty influencers, for instance, have initiated numerous makeup tutorials, drawing a substantial audience, primarily composed of teenage girls. Currently, 200 million users contribute to the creation of 400 million new YouTube videos daily.

While this expansive environment on YouTube provides a platform for creative expression, it also presents an opportunity for spammers to disseminate irrelevant content. These unsolicited messages often aim to entice users into clicking on links that lead to malicious sites containing malware, phishing, and scams. Notably, the comments section below each video is a prominent feature of YouTube, enabling users to share opinions and ideas.

This project focuses on predicting spam comments within YouTube video comments using machine learning, a subset of artificial intelligence. The chosen supervised learning approach relies on a large number of labeled datasets..

Spam comments, typically generated by automated bots posing as users, are often entirely irrelevant to the video content. The comments section becomes a prime target for spammers to post irrelevant messages, links, and ideas. Artificial intelligence serves as the methodology for extracting, transforming, loading, and predicting significant data from vast datasets, allowing for pattern recognition and converting data into an understandable format for further use. Classification and prediction are two fundamental types of data analysis that describe the categorization of data and the forecasting of trends in future data.

1.1 Motivation

This project is motivated by the pervasive influence of YouTube, a central hub for information and entertainment. The escalating threat of spam comments jeopardizes the platform's integrity, impacting user experience and the credibility of content creators. As YouTube evolves into a space for influencers and community engagement, the disruption caused by spam becomes more detrimental. Leveraging advancements in machine learning, the project seeks to deploy a proactive solution for the automated detection and mitigation of spam. By addressing this issue, the aim is to fortify YouTube's user experience, foster a positive online environment, and safeguard the platform's standing in the digital landscape.

1.2 Objective

The primary objective of this project is to develop an efficient and automated system for the detection and elimination of spam comments on YouTube. Utilizing machine learning algorithms, the goal is to enhance the platform's ability to discern irrelevant or promotional content within the comments section. This project aims to contribute to the preservation of content quality and user experience on YouTube by providing a robust solution to mitigate the impact of spam. Additionally, the objective includes showcasing the effectiveness of machine learning techniques in addressing contemporary challenges in online community management, thereby advancing knowledge in the intersection of technology and social platforms.

1.3 Problem Statement

The escalating prevalence of spam comments on YouTube poses a critical challenge, adversely affecting the platform's content quality, user experience, and overall credibility. Despite YouTube's widespread popularity, its susceptibility to spammers jeopardizes the integrity of the online community. Existing spam filtering systems, while in place, are not entirely effective, allowing irrelevant and potentially harmful content to persist. This project addresses the problem by proposing a novel approach to accurately detect spam comments. The goal is to enhance YouTube's content moderation capabilities, creating a safer and more enjoyable environment for users and content creators.

1.4 Aim

Detection of spam comments on YouTube is a common problem in content moderation and online community management and The primary goal for this project is to develop a machine learning model that can accurately detect comments on YouTube videos as spam or legitimate.Leveraging Multinomial Naive Bayes and Support Vector Machine classifiers, the project aims to enhance the platform's content moderation capabilities. Through the utilization of TF-IDF text vectorization, the models are trained to accurately identify and classify spam comments, contributing to an improved user experience and content quality. The project also includes the creation of a user-friendly GUI application, allowing users to interact with the trained models for real-time spam comment classification. The ultimate goal is to provide an effective and accessible solution to mitigate the impact of spam on the YouTube platform.

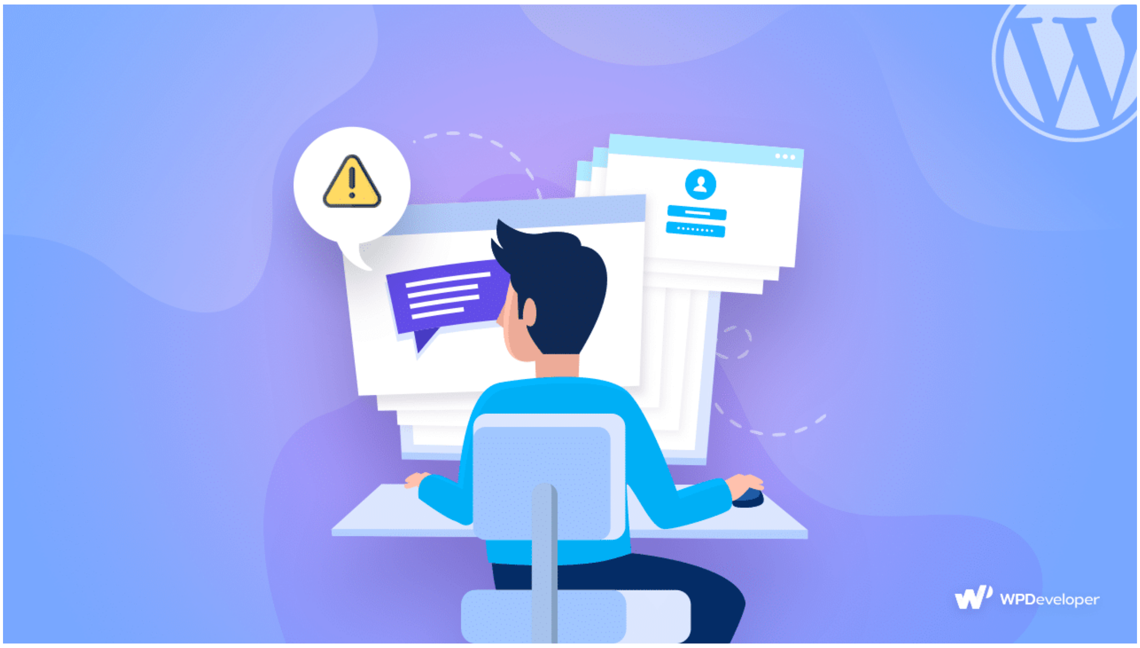

Figure 1.1: Detecting spam comments

Chapter 2

Literature Survey

The paper titled ”A Research on YouTube Spam Comments Detection and Deletion” [1] discusses the different techniques used to identify and remove spam comments. The authors propose a new method that uses natural language processing (NLP) to detect spam comments. They compare their method to existing methods and show that it is more accurate. They utilize the YouTube Data API to access and remove comments. Their method underwent evaluation on a dataset of YouTube comments, demonstrating its effectiveness. The method uses a machine learning algorithm to classify comments as spam or not spam. The algorithm is trained on a dataset of labeled YouTube comments.

The paper titled ”Detection of Spam in YouTube Comments Using Different Classifiers” [2]reviews extensive research on spam detection across various domains, highlighting the limitations of rule-based filtering, statistical methods, and machine learning algorithms. Rule-based filtering, reliant on manually crafted rules, is easily circumvented by spammers employing obfuscation or synonym substitution. Statistical methods, though more effective, can still be deceived by carefully crafted spam comments. Machine learning algorithms, particularly Random Forests, emerge as the most promising approach, capable of learning from labeled comments to identify distinguishing patterns between spam and legitimate comments.

The paper titled ”YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model” [3] involves a systematic approach to address the growing challenge of spam comments on YouTube.The focus is on leveraging machine learning techniques to enhance the performance of spam comment detection, acknowledging the limitations of YouTube’s existing spam blocking system. This model dissects word sequences, providing insights into potential spam. Additionally, a language modeling approach is applied to blog comments, where different language models are used for blog posts, comments, and external links. The core contribution of the research lies in the proposed Cascaded Ensemble Machine Learning Model tailored for YouTube spam comment detection. Two ensemble models, employing hard and soft voting strategies, respectively, are introduced. These ensemble models combine the outputs of six distinct machine learning techniques, including decision trees, logistic regression, Bernoulli Naïve Bayes, random forests, and support vector machines with linear and Gaussian kernels. The chosen techniques are carefully curated for their effectiveness in spam comment classification.

The paper titled ”N-Gram Assisted Youtube Spam Comment Detection” [4] presents a novel approach for detecting spam comments on YouTube utilizing n-grams, a technique that involves extracting sequences of n consecutive words from text. The authors employ the YouTube Data API to access and remove comments, evaluating their method against a dataset of YouTube comments to demonstrate its effectiveness in identifying spam comments with high accuracy.N-gram generation follows, where sequences of n words are extracted from the preprocessed comments. These n-grams serve as the basis for feature extraction, encompassing the frequency of each n-gram, the average length of the n-grams, and the entropy of the n-grams.Subsequently, a classifier is trained on the extracted features, enabling it to categorize new comments as either spam or legitimate. Finally, the performance of the classifier is evaluated on a separate test dataset, measuring its accuracy, precision, recall, and F1-score.

Chapter 3

System Design and Workflow:

3.1 Proposed System

The proposed method entails a comprehensive approach to spam comment detection using YouTube data. The initial step involves preprocessing the dataset, a collection of YouTube comments. This preprocessing likely encompasses tasks such as data cleaning and tokenization. Subsequently, the comments undergo text vectorization using TF-IDF (Term Frequency-Inverse Document Frequency), transforming them into numerical features suitable for machine learning analysis. The method employs two machine learning models, Multinomial Naive Bayes, and Support Vector Machine (SVM). These models are trained on the preprocessed data, and their performance is assessed using various metrics. To enhance user accessibility, a user-friendly GUI application is created with tkinter, enabling users to interact with the trained models for spam comment classification intuitively. For effective data handling and machine learning tasks, the method utilizes pandas and scikit-learn. The training function loads and preprocesses YouTube comments data, trains the models, and saves them along with the TF-IDF vectorizer using pickle. The architecture also incorporates a Flask web application as an interface, where users can input comments and receive predictions on whether the comment is classified as spam or not.

3.2 Dataset Description

The dataset is a comprehensive compilation of YouTube comments, enriched with corresponding metadata, offering a nuanced understanding of user interactions on the platform. Each row in the dataset represents an individual comment posted on a YouTube video, while the columns provide intricate details about the comment itself, the author, and the associated video.

Here’s a more detailed breakdown of the dataset columns:

- **COMMENT ID:** This unique identifier for each comment facilitates seamless referencing within the dataset, aiding in the identification and tracking of specific comments over time.
- **AUTHOR:** The ”AUTHOR” column captures the name or username of the individual responsible for the comment, adding a layer of personalization to the dataset and offering insights into the identity of the commenter.
- **DATE:** This timestamp in the ”DATE” column precisely indicates when each comment was posted on the YouTube video, enabling a temporal analysis of user engagement and interaction patterns.
- **CONTENT:** The ”CONTENT” field encapsulates the textual essence of the comments, capturing the sentiments, opinions, or interactive elements expressed by users on the YouTube platform.
- **CLASS:** Represented as a binary label in the ”CLASS” column, comments are categorized as either spam (1) or non-spam (0). This binary classification is instrumental for training and evaluating spam detection models, crucial for maintaining a clean and engaging user experience.

3.3 Workflow

The system’s workflow is organized into sequential steps, offering a concise overview with a focus on gestures and their corresponding controls.

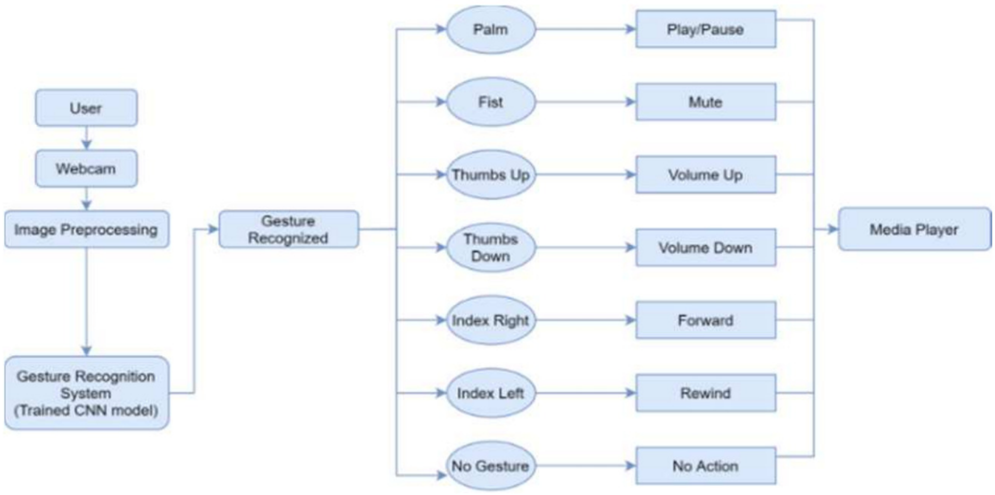

Figure 3.1: System Design Workflow

1. Image Acquisition and Pre-processing: Users perform hand gestures in front of the webcam. Using OpenCV, live video frames are collected and subsequently transformed into black and white images This conversion aims to enhance the accuracy of gesture prediction. The processed images are organized into specific directories for storage.To facilitate user interaction, the system offers two modes: ”train” and ”test.” Users can choose either mode, with the ”train” mode utilizing images for model training, while the ”test” mode evaluates the model's accuracy. The images captured are stored in directories based on user input.During operation, two frames are displayed on the screen, allowing users to capture images frame-by-frame using the read function. A simulated mirror image enhances the user experience. Gesture performance takes place within the Region Of Interest (ROI), defined as the bounding box. Frames are extracted from the ROI, resized to 120x120x1, and the image count in each directory is dynamically displayed on the screen. User interactions, such as capturing images and assigning them to specific gestures.Throughout the capture process, a small frame provides real-time feedback with visualizations of the preprocessed images. The collected images are then stored in the dataset. Users can conclude the data collection phase by pressing the escape key on the keyboard.

2.Feature Extraction: In the Feature Extraction module, the focus is on constructing a robust Convolutional Neural Network (CNN) architecture for the accurate extraction of features from preprocessed images. This is a pivotal step in the system workflow, crucial for precise gesture classification.The architecture of the CNN model is structured with care. It begins with a hidden input layer, establishing the foundation for subsequent operations. Two convolution layers follow, where essential features are identified. After each convolution layer, Rectified Linear Unit (ReLU) activation is applied, enhancing the non-linear characteristics of the model. Additionally, MaxPooling layers are incorporated to down-sample the spatial dimensions of the feature maps, ensuring efficient feature extraction.The feature extraction process is then channeled through a flattening layer, preparing the extracted features for input into fully connected layers. Two such layers are included, each serving a distinct purpose. The first fully connected layer utilizes ReLU activation, contributing to the model's capacity to capture intricate patterns in the features. The second fully connected layer employs softmax activation, facilitating gesture classification by assigning probabilities to each gesture class.

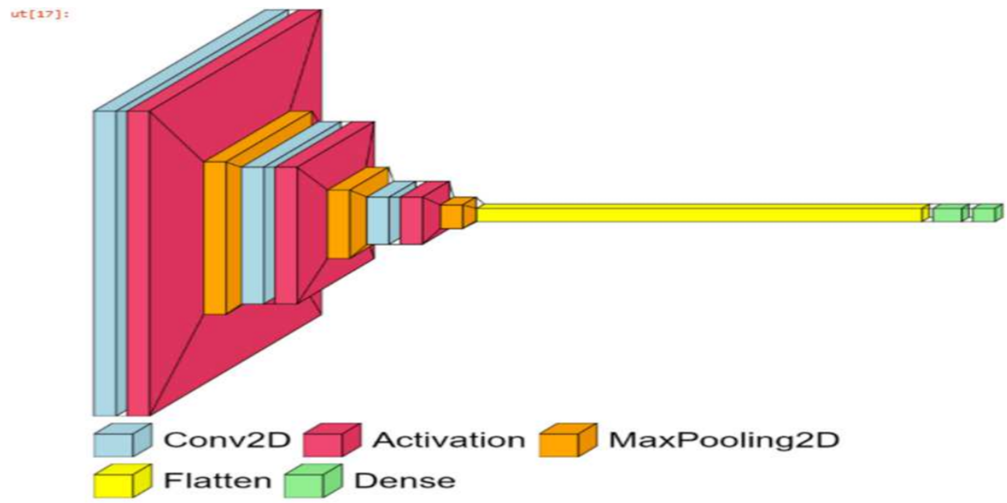

Figure 3.2: Architecture of CNN

The carefully designed Convolutional Neural Network (CNN) architecture plays a crucial role in enabling the system to effectively recognize and utilize the unique features present in hand gestures for accurate classification.Once the architecture is defined, the next step is the compilation of the model. During this process, the optimization algorithm, in this case, the Adam optimizer, is employed. The Adam optimizer dynamically adjusts the learning rates for each parameter in the network, facilitating quicker convergence and enhancing the overall training efficiency. It essentially guides the model towards optimal parameter values during the training phase.For assessing the performance of the model during training categorical crossentropy is chosen as the loss function.To gauge the effectiveness of the trained model, accuracy is adopted as the performance metric. This comprehensive design and configuration of the CNN model in the Feature Extraction module lay the groundwork for subsequent stages, where the model is trained and evaluated on the dataset.

3.Train the model: In the subsequent stage of the workflow, the model is trained using the ImageDataGenerator class, which facilitates the generation of batches of images for both training and validation purposes.The training process is executed using the fit function, which involves iterating through the dataset for a fixed number of epochs. During each epoch, the model learns to recognize patterns and features in the hand gesture images, gradually improving its ability to accurately classify them.Upon completion of the training phase, the trained model is saved in JSON format.This phase involves training the model by generating image batches, iterating through the dataset for a specified number of epochs.

4. Media control using predicted Hand gestures: The trained model stored in JSON format is loaded to predict hand gestures effectively in the prediction phase.The loaded model is then employed to classify hand gestures based on the features extracted during the training.The integration of PyAutoGUI facilitates the mapping of predicted gestures to corresponding keyboard controls. Additionally, Streamlit is employed to create a user-friendly interface comprising three web pages. The first page introduces the project, the second features a video demonstration, and the third serves as the interactive demo page for gesture-based media player control.Upon clicking the start button, the webcam is activated, enabling users to perform hand gestures within the defined Region Of Interest. The CNN model predicts the gestures in real-time, and PyAutoGUI, through conditional if-else statements, triggers the associated keyboard controls based on the recognized gestures. Each gesture is linked to a specific keyboard key, and the control function activates once per gesture prediction. Exiting the system is easy—users just press the escape key. The video frame shows the predicted gesture and the action it triggers, making it simple for users to control the media player seamlessly with hand gestures.

5.Web App and Deployment: The system includes a web application featuring about introduction page,demo page.To make the system accessible, the web application is deployed on streamlit.io.

Chapter 4

Implemention

4.1 System Modules

1. Data Acquisition and Preprocessing Phase:
Data Acquisition:
Acquire YouTube comment data from diverse sources and concatenating CSV files into one.
Data Preprocessing:
Concatenate data from multiple files into a unified DataFrame, ensuring a comprehensive dataset. Discard extraneous columns such as "COMMENT_ID," "DATE," and "AUTHOR" to streamline relevant features. Perform a split, isolating features (X) representing the comment content and the target variable (y) denoting spam classification. Employ the TF-IDF vectorization technique to convert textual comment data into numerical features suitable for machine learning analysis.

2. Exploratory Data Analysis Phase:
Data Inspection:
Conduct a detailed exploration of the dataset to unravel its inherent structure, uncovering patterns, and gaining insights into the nature of YouTube comments.
Data Visualization:
Leverage data visualization techniques, including histograms, word clouds, and confusion matrices, to visually represent comment distributions, highlight prevalent terms, and identify potential correlations between features.

3. **Classification and Prediction:**
Model Selection:
Opt for the Multinomial Naive Bayes classifier, an effective choice for text classi-

fication tasks due to its simplicity and efficiency.
Model Training:
Train the selected model on the preprocessed training data, allowing it to learn patterns and relationships within the YouTube comment dataset.
Model Evaluation:
Evaluate the model's performance using a range of classification metrics such as accuracy, precision, recall, and F1-score to gauge its effectiveness in distinguishing between spam and non-spam comments.

4. **Model Deployment Phase:**
Integration with Flask App:
Develop a dynamic Flask web application, providing an interactive platform for users to engage with the spam comment detection system seamlessly.
User Interaction:
Enable users to submit comments through a user-friendly form, enhancing the user experience.
Prediction Display: Present the model's predictions in a clear and concise manner within the web interface, ensuring users receive immediate feedback on the spam classification of their comments.

4.2 Algorithm

Multinomial Naive Bayes (MNB):
The Multinomial Naive Bayes algorithm is a probabilistic classification algorithm based on Bayes' theorem.It is widely used for text classification tasks, including spam detection, due to its simplicity and effectiveness.It assumes that features are conditionally independent given the class label, making it particularly suitable for processing and classifying text data.

Support Vector Machine (SVM):
Support Vector Machines are a powerful supervised learning algorithm used for classification and regression tasks.SVMs are effective for text classification, image recognition, and spam detection, among other applications.SVM finds a hyperplane that best separates data into different classes. It aims to maximize the margin between the classes while minimizing the classification error.

TF-IDF (Term Frequency-Inverse Document Frequency):
TF-IDF is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents (corpus).TF-IDF is used for text feature

extraction and is crucial in natural language processing tasks, such as information retrieval and text mining.It calculates the product of two metrics, Term Frequency (TF) and Inverse Document Frequency (IDF). TF measures how often a term occurs in a document, while IDF measures the rarity of a term in a corpus..

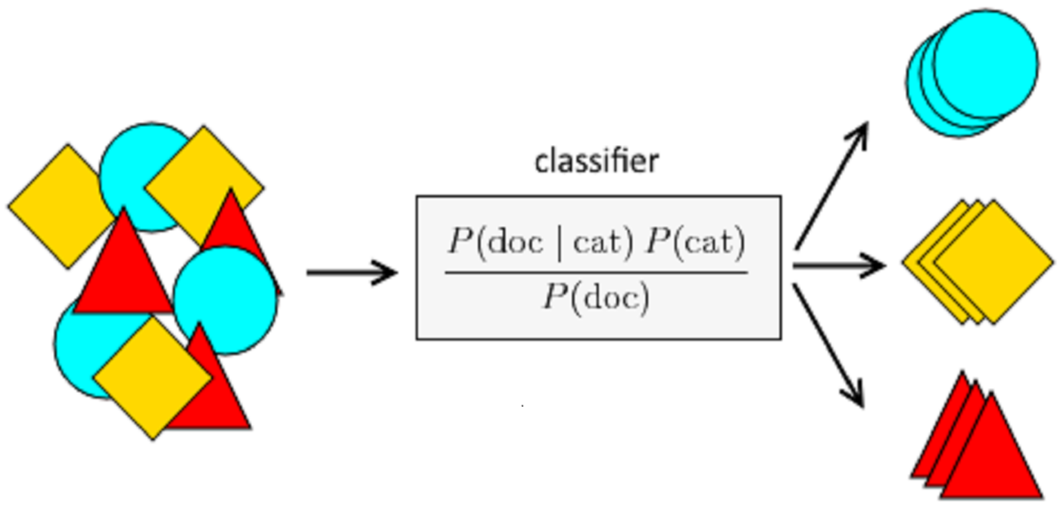

Figure 4.1: Multinomial Naive Bayes

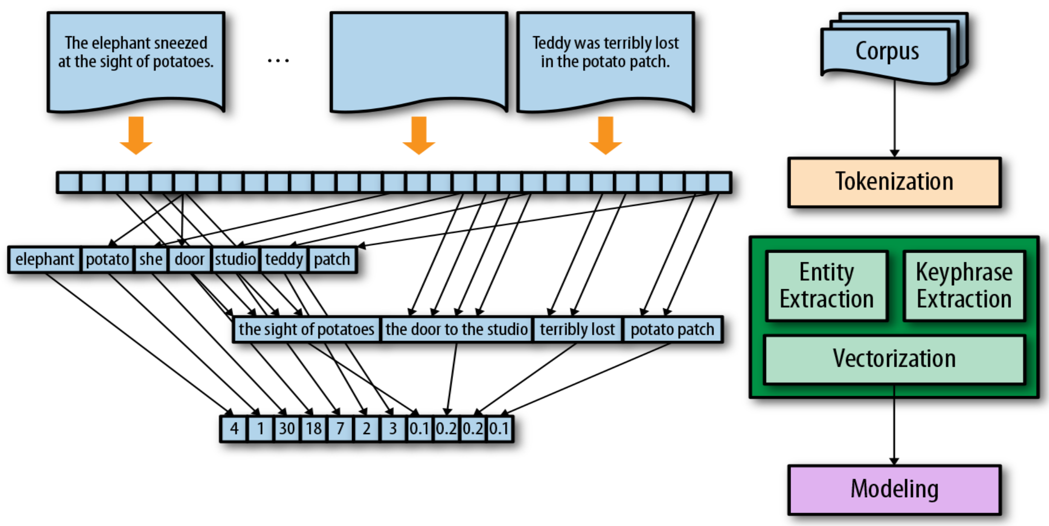

Figure 4.2: Text Vectorization

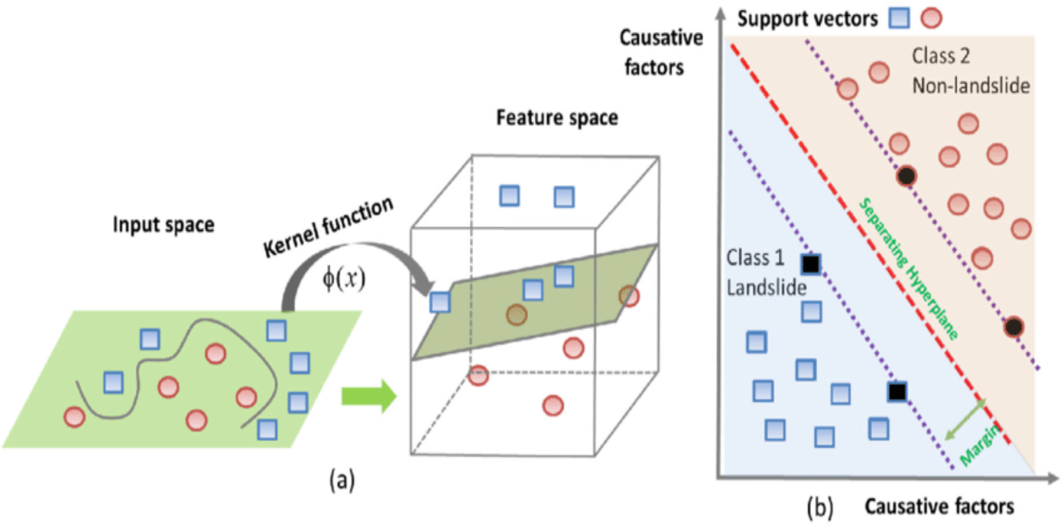

Figure 4.3: Support vector machine

Chapter 5

Results and Discussions

5.1 Experimental Results

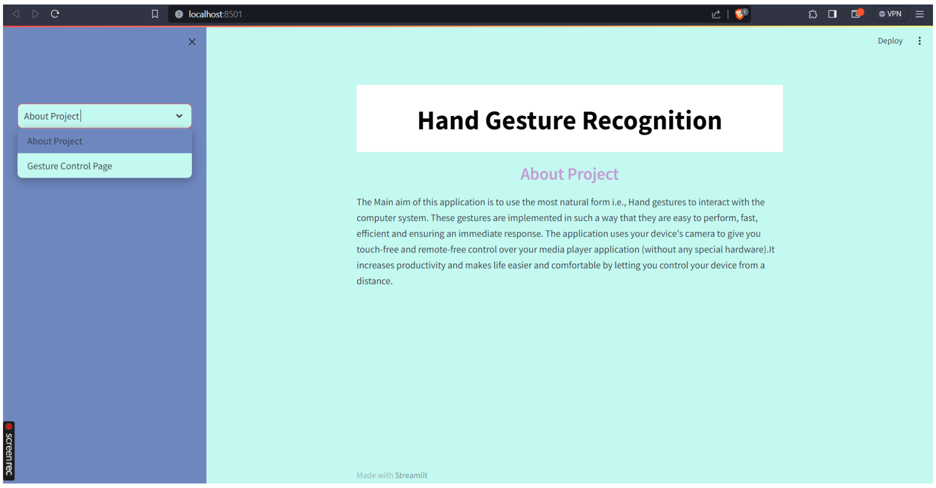

Figure 5.1: Web Page Home

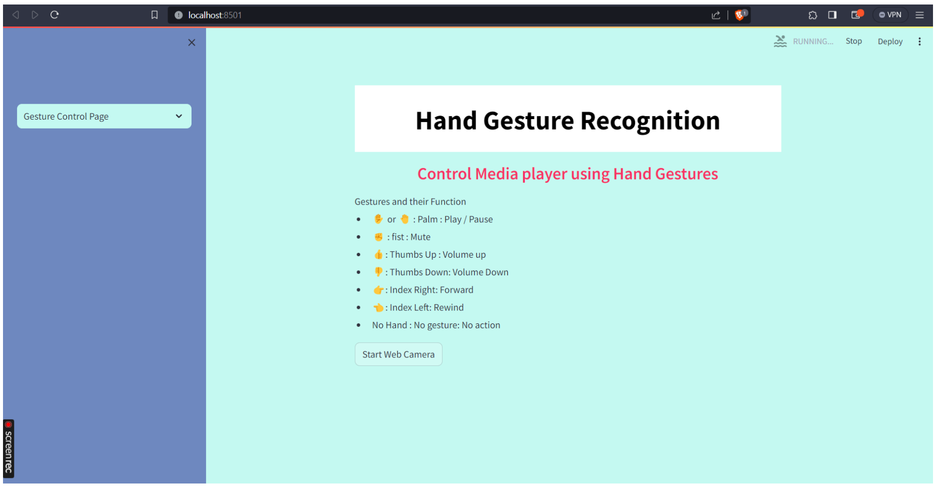

Figure 5.2: Gesture based Media player control Web Page

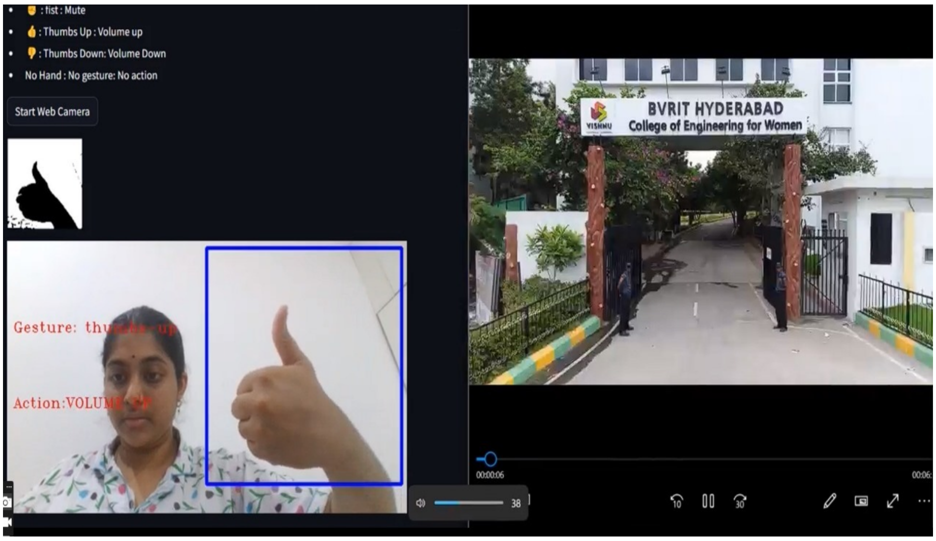

Figure 5.3: Volume Up Gesture

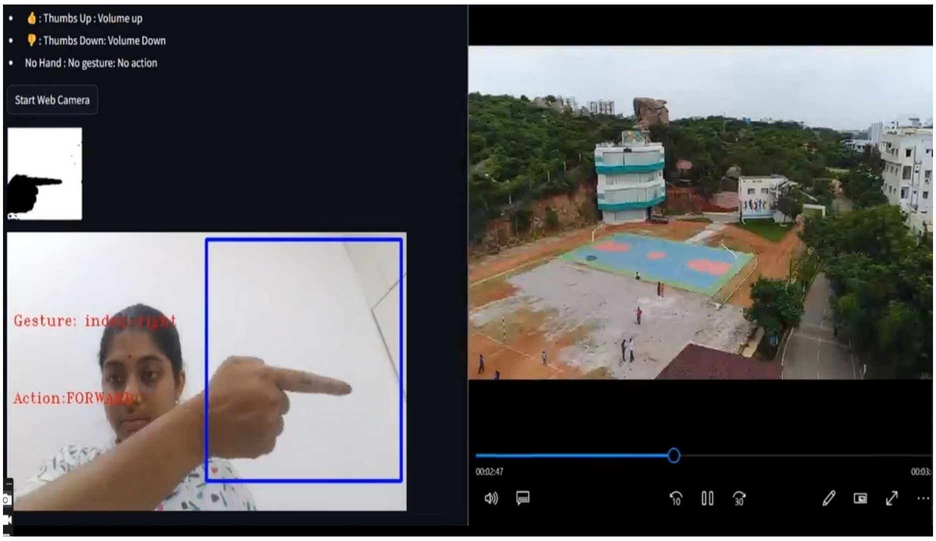

Figure 5.4: Forward Gesture

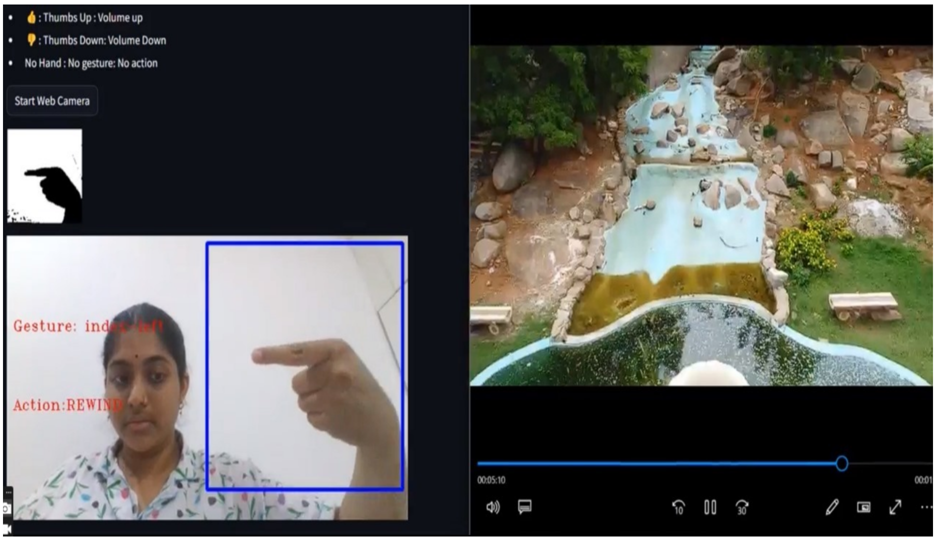

Figure 5.5: Backward Gesture

5.2 Analysis

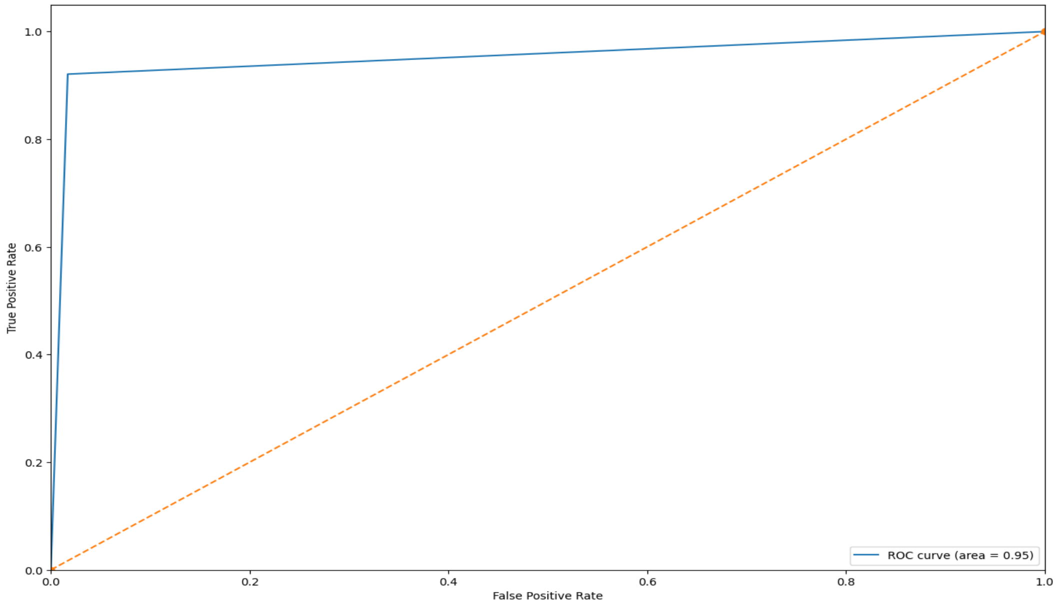

Figure 5.6: ROC curve for Multinomial Naive Bayes

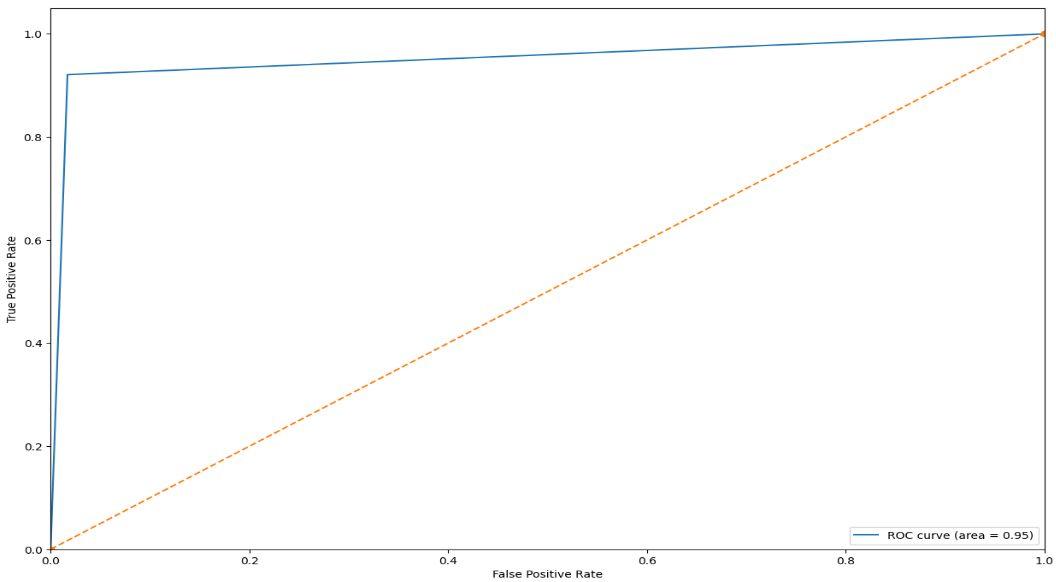

Figure 5.7: ROC curve for SVM

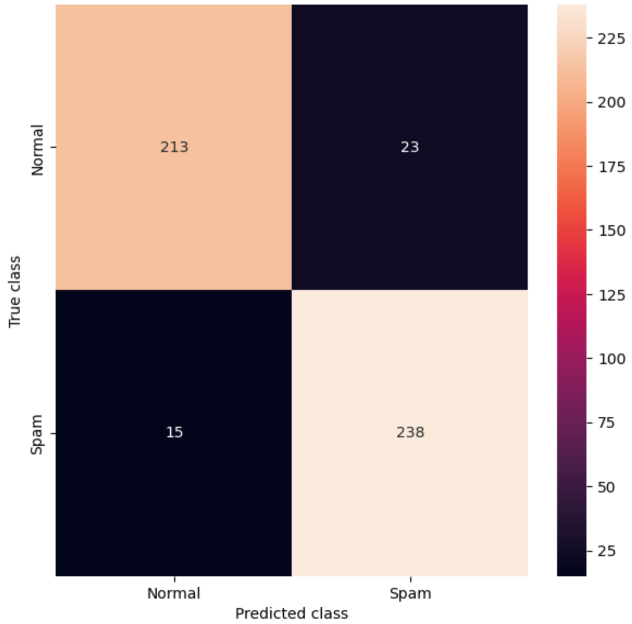

Figure 5.8: Confusion matrix of MNB

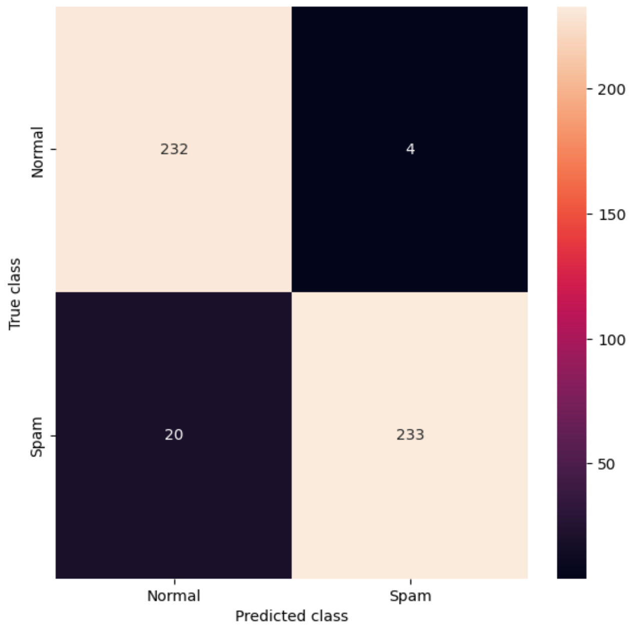

Figure 5.9: Confusion matrix of SVM

Chapter 6

Conclusions and future works

6.1 Conclusion

In conclusion, we developed a spam comment detection system using machine learning techniques and a graphical user interface (GUI) built with Tkinter. The implementation involved preprocessing YouTube comments, training Multinomial Naive Bayes and Support Vector Machine classifiers, and evaluating their performance. The classifiers demonstrated reasonable accuracy, as evidenced by confusion matrices and ROC curves. The GUI provides a user-friendly interface for classifying new comments, extending the project's usability. The code is structured effectively, with the inclusion of pickling for model and vectorizer persistence. Future enhancements could involve refining text preprocessing, exploring alternative vectorization methods, and implementing hyperparameter tuning for model optimization. Overall, this project provides a solid foundation for spam comment detection, offering a practical tool for users to assess the spam likelihood of comments in a real-world scenario.

6.2 Future scope

- **Integration with Youtube:** Integrating the spam comment detection system with YouTube involves using the YouTube API for real-time comment analysis. This integration allows for personalized settings, automated comment moderation, and a dashboard for content creators to monitor engagement.Ultimately, the integration aims to enhance user experience by offering real-time spam detection and facilitating a cleaner and more engaging comment section on YouTube channels.By automatically identifying and filtering out spam comments in real-time, the system alleviates the burden on content creators, saving time and fostering a more positive online environment.

- **User Authentication:**

  Implementing user authentication enhances the spam comment detection system by allowing users to personalize their experience. Authenticated users gain the ability to manage monitored videos, receive tailored alerts, and contribute to the system's feedback loop. This ensures a more personalized and interactive user experience, fostering user engagement and contributing valuable insights to improve the system's overall effectiveness in combating spam.

- **Multi-language Support:**

  Implementing multi-language support is crucial for enhancing the spam comment detection system's global applicability on YouTube. By accommodating diverse languages commonly used on the platform, the system becomes a valuable tool for content creators and users worldwide. This feature ensures that the system can effectively analyze and classify comments in various languages, contributing to a more inclusive and comprehensive approach to content moderation. With this capability, the system becomes adept at identifying spam across linguistic boundaries, offering a robust solution for a diverse and international user base on YouTube.

- **Ensuring adaptability to evolving spamming trends:**

  Ensuring the system's adaptability to evolving spamming trends is essential for maintaining its effectiveness in combating emerging forms of online spam. As spamming techniques continually evolve, the system must be equipped to identify and classify new patterns and tactics used by spammers. Regular updates and continuous monitoring of spamming trends will be crucial. This adaptability ensures that the spam comment detection system remains resilient in the face of evolving challenges, providing a proactive defense against emerging forms of online spam and preserving its efficacy in safeguarding user experiences and content quality on platforms like YouTube.

Bibliography

[1] V. D. D. D. D. J. N. Gayathri Devi Nagalapuram, Roopashree S, "Controlling media player with hand gestures using convolutional neural network," *IEEE Access*, 2021.

[2] M. B. M. N. Mohamed and N. Jomhari, "A review of the hand gesture recognition system: Current progress and future directions," *IEEE Access, vol. 9*, pp. 157422–157436, 2021.

[3] S. G. Yashas J, "Hand gesture recognition: A survey," *International Conference on Applied Machine Learning (ICAML), IEEE Volume*, pp. 3–8, 2019.

[4] S. N. Sharma P, "Gesture recognition system," *International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) , IEEE Volume*, pp. 1–3, 2019.