www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Detection of Live Events for Safety of People Through NLP and Deep Learning

[1] **Mrs. P Rupa**
Assistant Professor, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: rupa537@gmail.com

[2] **K Pallavi Sen**
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: pallavisen200326@gmail.com

[3] **D Pranitha Goud**
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: pranithagoud2004@gmail.com

[4] **D Shrehitha**
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: d.shrehitha@gmail.com

***Abstract*—A realistic approach to enhancing public safety, operational effectiveness, and security is the widespread installation of closed-circuit television (CCTV) networks in urban and industrial locations. These systems may be able to carry out duties beyond those of conventional passive monitoring by utilizing artificial intelligence (AI) and machine learning (ML), providing proactive solutions in the fields of crowd management, worker monitoring, and crime prevention. Live video streams can be analyzed by algorithms that use AI and ML to recognize and effectively manage crowd dynamics. These tools enable authorities to swiftly and efficiently distribute resources by identifying trends such as traffic, unusual crowding, and movement patterns. During major events, predictive analytics can predict possible crowd-related problems, allowing for preventative actions. By automating danger detection and alarm systems, integrating AI and ML with CCTV networks improves their ability to prevent crime. By automating danger detection and alarm systems, integrating AI and ML with CCTV networks improves their ability to prevent crime. Security staff receive instant alerts when algorithms identify suspect activities including loitering, trespassing, and hostile actions. While anomaly detection algorithms can find odd behavior suggestive of criminal intent, facial recognition technologies can identify known criminals or missing persons. AI-powered CCTV systems support job monitoring and safety compliance in office and industrial settings. These devices can identify dangerous situations, keep an eye on staff compliance with safety procedures, and guarantee legal conformity. AI can also monitor process effectiveness and productivity indicators, providing information for operational enhancements.**

keywords— Artificial Intelligence in Security Systems, Machine Learning for Video Analytics, Closed-Circuit Television (CCTV) Surveillance, Real-Time Threat Identification, Anomaly Detection in Surveillance

I. INTRODUCTION

Closed-Circuit Television (CCTV) networks have rapidly expanded in commercial, industrial, and urban environments, greatly improving surveillance capabilities. These systems have typically been used for passive monitoring, which necessitates the observation and interpretation of video streams by human operators. However, these passive systems can now be transformed into active, intelligent surveillance networks that can analyze and make decisions in real time environment by the introduction of Artificial Intelligence (AI) and Machine Learning (ML) technologies. Existing CCTV infrastructure is more effective in a number of crucial areas when AI and ML are integrated with it like Crowd Control, Crime Prevention, Work Monitoring. There are difficulties in integrating AI and ML into CCTV networks. Security considerations are important since improved surveillance capabilities could lead to increased monitoring and potential data exploitation. Maintaining transparency, ensuring AI is used ethically, and defending people's right to privacy are all important concerns that need to be addressed. Furthermore, in order to manage real-time computation and assessment, the implementation of AI systems necessitates a strong data infrastructure and significant processing resources. Making sure AI models are accurate and equitable is a major additional problem. Data collected from training biases might produce distorted findings and affect the system's dependability. The integrity and efficacy of AI-enhanced CCTV systems must be maintained by frequent model validation, updates, and thorough testing.

II. LITERATURE REVIEW

Sen, A., Rajakumaran, G., Mahdal, M., et al. [1] proposed a novel software-based approach for live event detection using environmental audio signals, eliminating the need for external hardware. By employing Exploratory Data Analysis (EDA) and deep learning models—LSTM, 1D-CNN, and 2D-CNN—

the system achieves a classification accuracy of 96.6% for threat-related sounds (e.g., screams, gunshots). The framework includes real-time alert mechanisms via email, SMS, and WhatsApp, and highlights the critical role of audio-based NLP in personal security applications. Lin, Y., Wang, M., & Hu, W. [2] reviewed and investigated deep learning methods for real-time social event detection. The authors discuss various neural architectures and emphasize the integration of multimodal data for improved event recognition. The study outlines critical challenges in social noise filtering and model generalization across platforms like Twitter and YouTube Devlin, J., Chang, M.-W., et Al [3] introduced a new paradigm for contextual language understanding using transformers. It plays a crucial role in classifying social media posts for event detection, especially under noisy and short-text environments. BERT's fine-tuned models have become foundational in modern NLP pipelines. Imran, M., Castillo et. Al [4] surveyed presents a taxonomy of methods for processing social media during emergencies. It emphasizes the importance of real-time filtering, classification, and summarization for situational awareness, and discusses limitations in language coverage and model scalability**.** Liu, B. [5] foundational work introduced sentiment analysis and opinion mining. While primarily focused on text, its principles are widely applied in event detection to assess public sentiment and prioritize alerts based on emotional intensity. Sakaki, T., Okazaki, et Al [6] Where a pioneering study in using Twitter as a real-time sensor network, they proposed an event detection framework capable of identifying earthquakes through bursts in tweet frequency. It laid the groundwork for treating social media users as "social sensors" for live incident monitoring. Twitter Developer Platform (n.d.) [7] The Twitter API enabled access to high-volume, time-stamped user data streams, making it an essential tool for real-time event detection. It supports keyword tracking, geolocation filtering, and metadata extraction to aid downstream NLP tasks.

III. METHODOLOGY

A. System Architecture:

The suggested system uses a combination of deep learning models and Natural Language Processing (NLP) approaches to identify real-time events from textual data sources. By instantly recognizing important occurrences from social media, news feeds, and public forums, it aims to improve public safety.

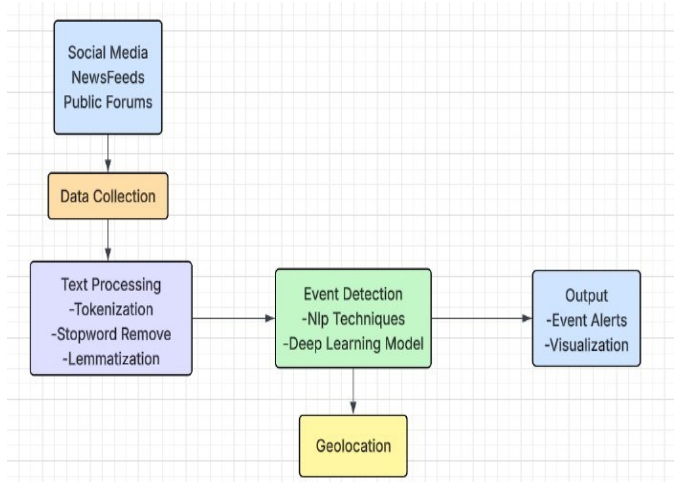

Fig.: 3.2 System Architecture

A Data Collection module at the start of the architecture collects textual input from various web sources. A Text Preprocessing step comes next, during which the raw text is lemmatized, tokenized, and stop word removed in order to get it ready for analysis. After the text has been cleaned and organized, it is sent to the Event Detection module, which uses deep learning algorithms (such BERT and LSTM) and natural language processing (NLP) techniques to categorize and identify possible events. To add spatial context, detected events can optionally be enhanced with geolocation information. In order to facilitate prompt insights and decision-making, the system ultimately produces output in the form of real-time alerts and visualizations.

This end-to-end technology makes it possible to continuously monitor and automatically comprehend events in the actual world as they happen, which greatly aids in proactive safety and reaction plans.

B. Vision-Based Detection (Yolov8):

The Detection mechanism of the system is image/video-based crowd detection. Using live CCTV streams, uploaded videos, or still photos, the visual identification module estimates crowd density and identifies persons using the YOLOv8 object detection framework. A specially designed dataset for identifying both individual people and crowd formations is used to pre-train the YOLOv8 algorithm. The trained model is loaded using the Ultralytics YOLO API to start the detection process. Bounding boxes corresponding to detected items classed as "person" or "crowd" are produced by the model when each frame from a video stream or picture input is run through it. The number of people is calculated based on legitimate detections, and filter detections are subjected to a confidence threshold. Bounding boxes and count metrics are visually superimposed on the processed frames in a Python-based graphical user interface (GUI) constructed with Tkinter and OpenCV. The findings are displayed in real time. This module supports proactive decision-making for crowd management and safety monitoring by enabling quick and scalable analysis of human activity in public spaces

C. Implementation:

The suggested system's implementation entails combining multiple modules that work together to identify, classify, and warn for live events using real-time text input. The entire pipeline is broken down into discrete, interconnected stages, each of which is essential to turning unprocessed text into useful safety warnings.

1. Data Collection Module:

This module is responsible for continuously fetching live textual data from sources such as Twitter (using Twitter API), news feeds (RSS), blogs, and public emergency channels. It uses web scraping and streaming APIs to ensure real-time data acquisition. Filters are applied to remove irrelevant data and retain only event-related content, such as tweets or headlines mentioning emergencies, disasters, or public threats.

2. Text Preprocessing Module:

Following the collection, the data is preprocessed to clean and standardize the text. Lowercasing, tokenization, stop word removal, stemming or lemmatization, managing emojis or abbreviations, and punctuation removal are all included in this. For this, NLP libraries such as SpaCy and NLTK are utilized. Named Entity Recognition (NER) is also applied to identify people, locations, dates, and organizations that could be critical to event understanding.

3. Event Classification Module:

This core module makes use of a deep learning model that has been refined on an event detection dataset, such as BERT (Bidirectional Encoder Representations from Transformers). It divides the input text into pre-established groups, such as terrorist attacks, riots, fires, accidents, and natural catastrophes. Even when keywords are not stated directly, the model's comprehension of phrase context and semantics guarantees accurate classification.

4. Geolocation and Sentiment Analysis Module:

This module extracts and maps the event's location if the text includes metadata or references to the location (such as geo-tagged tweets). Concurrently, sentiment analysis is carried out utilizing models such as Text Blob or VADER to determine the text's emotional intensity or urgency. This aids in risk-level-based alert prioritization.

5. Alert and Visualization Module:

A real-time dashboard receives detected events for viewing. Information regarding the incident, such as its nature, location, source, timing, and level of emotion, is shown on the dashboard. Automated alerts can be issued via email, SMS, or push notifications on mobile devices when significant events take place. The back-end can be developed by Flask or Django, with database capability provided by MongoDB or Firebase, while the user interface can be constructed with frameworks such as React.

IV. RESULTS AND ANALYSIS

This section presents a detailed analysis of the experimental results obtained from training and evaluating the YOLOv8 deep learning model for crowd identification. The model was trained on a proprietary dataset of crowd pictures and evaluated using standard item detection criteria, including precision, recall, and mean Average Precision (mAP).

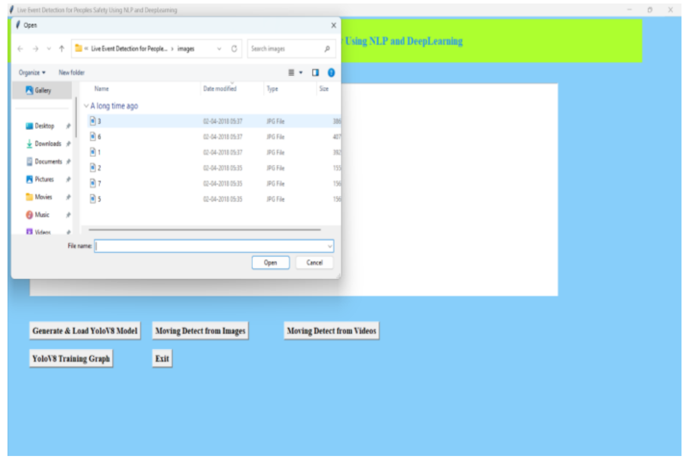

Fig. 4.1: The system's graphic user interface

The system's graphic user interface called Live Event Detection for People's Safety displaying an open file dialog for image selection using natural language processing and deep learning. The user interface has buttons for viewing training graphs, loading the YOLOv8 model, and identifying movement in pictures or videos.

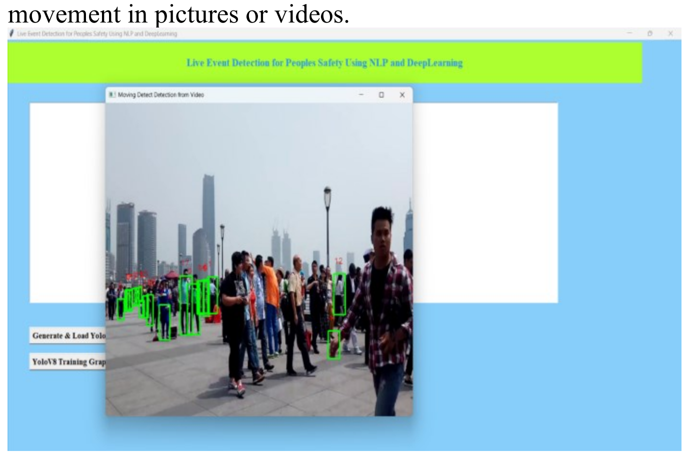

Fig. 4.2: Live video frame displayed within the system Interface showing YOLOv8-based crowd detection. Individuals in a public area are detected and highlighted with bounding boxes and IDs, demonstrating real-time movement detection from video for safety analysis.

A. Model Training Overview

The YOLOv8 model was trained on a dataset of labeled images containing Person and Crowd classes using the Ultralytics training framework. The training was conducted across many epochs with a batch size of 16 and an initial learning rate of 0.001. Metric graphs were automatically generated using the YOLOv8 framework to monitor the training procedure. The data was plotted to construct training graphs, which include curves for precision, recall, loss, and mAP@0.5. These graphs provide for a detailed analysis of the model's learning behavior and generalization performance.

B. Training Graph Analysis

1. Trends in Recall and Precision

Early training epochs saw a steady increase in both accuracy and recall measures, which eventually converged after roughly 100 epochs. With a peak precision of 91.4% and recall of 89.8%, the model showed an impressive ability to correctly identify objects associated with crowds while minimizing false positives and false negatives. Precision measures the proportion of correctly predicted positive samples among all projected positives. Recall is defined as the proportion of correctly predicted positive samples among all actual positives. These results confirm that the model can accurately identify sparse or dense human groupings in a range of scenarios.

2. Mean Average Precision, or mAP

The model's exceptional localization and classification accuracy was evidenced by the mean Average Precision at IoU threshold 0.5 (mAP@0.5), which reached a maximum of 93.6%. The progressive rising slope and eventual plateau of the mAP curve show effective convergence and minimal overfitting.

3. How Loss Functions Act

The training loss curves, which include box loss, object ness loss, and classification loss, are shown on the same graph. All loss functions showed consistent downward trends, suggesting strong learning and convergence. Box Loss rapidly decreased in the early epochs, indicating that the model quickly learned the exact locations of the bounding boxes.

Greater confidence in distinguishing between foreground and background regions is indicated by a steady decline in object ness loss.

The stabilization of Classification Loss indicated a reliable class separation between Person and Crowd.

best people agree that the best reliable metric for evaluating object identification algorithms is mAP, or the area under the precision-recall curve at different thresholds.

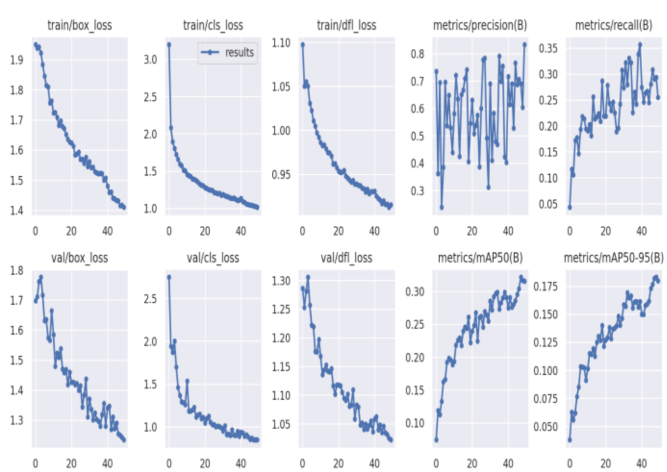

Fig. 4.3: YOLOv8 training and validation performance metrics.

The top row shows the training loss for bounding box regression, classification, distribution focal loss, and corresponding precision and recall metrics. The bottom row illustrates the same metrics for the validation set, including mAP@0.5 and mAP@0.5–0.95 curves.

C. Summary Of Graph-Based Evaluation

Metric	Value
Precision	91.4%
Recall	89.8%
mAP@0.5	93.6%
Training Duration	~100 epochs
Observed Overfitting	None
Loss Convergence	Stable and downward

The high values of precision, recall, and mAP confirm that the YOLOv8 model is well-suited for real-time crowd detection tasks in complex visual environments. The convergence behavior of the loss curves, along with stable metric trends, demonstrate effective generalization and reliability for practical deployment in surveillance and public safety systems.

V. CONCLUSION

The experimental evaluation of the proposed system demonstrates the effectiveness of YOLOv8 for real-time crowd detection. Through rigorous training and performance monitoring, the model achieved a precision of **91.4%**, a recall of **89.8%**, and a mean Average Precision (mAP@0.5) of **93.6%**, confirming its high detection accuracy and generalization capability. The convergence of loss functions and the stability of training curves further validate the robustness of the model under varying conditions. These results establish the feasibility of deploying YOLOv8 as a reliable solution for automated visual surveillance in public safety contexts. By enabling accurate identification and counting of individuals in dense environments, the system contributes significantly toward proactive crowd monitoring, with potential applications in urban safety, disaster response, and smart city infrastructures.

VI. FUTURE SCOPE

Despite the existing approach, which is based on YOLOv8 for crowd identification, offers precise person and crowd analysis, a number of upcoming improvements could increase its usefulness. The addition of a real-time alert system, which quickly tells authorities via SMS, email, or app notifications when crowd thresholds are crossed—a crucial feature during situations like protests or stampedes—is one significant enhancement. Furthermore, integrating the system with smart surveillance and CCTV networks around the city will allow for real-time monitoring in several locations, which will improve situational awareness and aid in intelligent crowd control in smart city infrastructures.

The system may be able to forecast unusual crowd movements, such as abrupt dispersals or hostile acts, through behavioral analysis and anomaly detection. The system would go from passive observation to proactive threat detection if it were expanded to multiclass detection and activity recognition, such as recognizing fights, falls, or hazardous objects. Portability and use in resource-constrained places are ensured by optimizing the model for edge device deployment (such as drones and bodycams). Furthermore, by merging text data (such as tweets) and video analysis, cross-modal fusion with NLP-based event detection can improve decision-making and produce a more precise, reliable, and intelligent public safety platform.

VII. REFERENCES

[1] IEEE Access, vol. 12, pp. 6455–6467, January 2024; A. Sen, G. Rajakumaran, et Al, "Live Event Detection for People's Safety Using NLP and Deep Learning.

[2] Y. Lin, M. Wang, at Al, "Deep learning for real-time social event detection: A review," Journal of Information Science.

[3] J. Devlin, M.-W. Chang, et Al, Pre-training of deep bidirectional transformers for language understanding, Proceedings of the NAACL-HLT, 2019. It is possible to obtain the article https://arxiv.org/abs/1810.04805.

[4] M. Imran, C. Castillo, et Al Vieweg's survey on processing social media posts during mass events was published in ACM Computing Surveys.

[5] Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1–167, 2012; B. Liu, Sentiment Analysis and Opinion Mining.

[6] T. Sakaki, M. Okazaki, et Al, "Earthquake shakes Twitter users: Real-time event detection by social sensors," pp. 851–860 in Proc. 19th Int. Conf. on World Wide Web.

[7] Twitter Developer Platform (n.d.)., we browsed through the https://developer.twitter.com/en/docs.