Grenze International Journal of Engineering and Technology, January Issue

Smart Surveillance with Hand Gesture Detection for Silent Emergency Alerts

Dr. Anil V Turukmane[1], Vishnu Vardhan Reddy[2], Ruthvik Reddy Anupati[3] and Reethwik Reddy Poreddy[4]
[1-4]School of Computer Science and Engineering, Vellore Institute of Technology, Amaravathi, AndhraPradesh, India
Email: anil.turukmane@vitap.ac.in, vishnu24004@gmail.com, anupatiruthvik@gmail.com, poreddyreethwik@gmail.com

***Abstract*— Student safety in our learning environments is challenged with ragging, mental heath challenges and medical emergencies. Pre-existing types of monitoring like audio surveillance and wearable devices are frequently versatile because of concerns with privacy, false alarms, and outlets. This paper proposes Silent Alert, a real-time gesture-based emergency detection system that can use computer vision and machine learning to recognize discrete distress.To accomplish that, the system uses MediaPipe to detect the locations of hands and OpenCV to process the live video. Several classifiers such as Random Forest, SVM, KNN, DNN and CNN+LSTM are trained upon a custom hand gesture dataset, as well as how to effectively integrate the deep learning-based YOLOv8 network in finding the gestures. When distress signals are detected, Silent Alert records image frames and indicates to the authorities via secure email alerts so that immediate action can be taken. Revolving around very low intrusiveness and being able to easily meld into the current campus infrastructure, Silent Alert presents a scalable solution with a minimal privacy overhead that increases the effectiveness of emergency response in educational environments as well as other crucial spheres.**

***Keywords*— Silent Alert System, Artificial Intelligence, Open-CV, Computer Vision, YOLOv8, Hand Gesture Recognition.**

I. INTRODUCTION

In the current situation where the world of academia is changing at a very high rate, the Student safety and mental health are becoming scorching issues, especially in an active and fast-paced academic world. In spite of the technological-based learning, uneven cases of ragging, harassment, complexion of the mind, and health crisis are still evident. Traditional safety systems are frequently insufficient, and accidents either do not go noticed or unreported as people fear to do so out of stigma or because the victims lack the capacity to report them without exposing themselves to further harm. This demonstrates why discreet, real-time solutions are necessary to enable students to communicate distresses without speaking and without direct confrontation.

Silent Alert System satisfies this requirement via intelligible distress signal system by means of describing an emergency via gestures. Based on the artificial intelligence (AI) and computer vision, the system watches video-feeds in real-time, tracking webcams or CCTV cameras, recognizing emergency gestures and responding to the authorities immediately. It is a non-intrusive design to make sure that students can communicate distress without making noise and attracting the unnecessary attention.

The main technologies used by this system are MediaPipe to track and know landmarks of the hand and OpenCV to read and process the video.

Grenze ID: 01.GIJET.12.1.534_3

Machine learning-based and deep learning-based models are utilized to classify extracted features, and these are Support Vector Machine (SVM), K-Nearest neighbor (KNN), Hybrid Stacking Classifier, and YOLOv8 to recognize gestures quickly and accurately. The metrics used to measure the performance are precision, recall, and the F1-score, so as to provide the least number of false alarms with proper results of accurate detection.

When facing a distress gesture has been detected, the program takes a screenshot of the associated frame of a video and securely sends it through SMTP directly to the campus authorities so that speedy action can be taken. The privacy is taken seriously regarding frame processing since only the necessary frames are processed and it is not singly recorded. It also lowers the cost of additional hardware as it is compatible with already existing CCTV or webcam infrastructure.

Compared to the current state of monitoring options, e.g., audio detection, wearable, or behavioral analytics, the Silent Alert System does not create errors of false positives due to any background noise, privacy, and compliance challenges, as well as misclassification risks. Its modular and scalable nature means that it can be deployed in hostels, libraries, dormitories and other settings and applications such as other area of use and application than in academia, such as healthcare, aging communities and transport.

This experiment aims to build a secure, effective, and ever-keen real-time recuperation movement identification system, that suits the study scope; which is training offices or schools. The synthesis of the AI-based detection system with the possibility to transmit the evidence instantly will enable the students to ask for assistance without fear and providing higher academic safety and the example of an ethical and efficient solution of the safety issue.

II. LITERATURE REVIEW

Gesture based safety system has experienced blistering growth in the past few years, due to effectiveness of technology in farm of computer visioning, deep learning and human and computer interaction. This section will clinically examine the most applicable studies that have informed the design and the deployment of the suggested Silent Alert system.

A. Hand Gesture Recognition with Machine Learning Algorithms

In this paper, the use of supervised machine learning of SVM, Decision Trees, and Naive Bayes in recognition of hand gestures is discussed. The data set comprises of the image of the gesture which is static and the processing of such data is carried out through the edge detection and histogram equalization process. The authors prove that performance of gesture recognition systems particularly relies on the pre processing of images and extraction of features. The conclusion that could be drawn about the study is that SVM demonstrates better classification accuracy in comparison to other algorithms and it suggests using the ensemble techniques to achieve an improved level of precision and alleviate sensitivity to noise.

B. Real-Time Hand Gesture Recognition Using Media-pipe and Open-CV

The paper presents Media-pipe Hands, a high-fidelity real-time hand tracking pipeline which works both on CPU and GPU environments. The system can recognize 21 3D landmarks of each of the hands based on only one RGB frame and it is very suitable when used in gesture recognition of human to computer interaction(HCI). The framework can be used to create dynamic gestures as long as it is combined with Open-CV categorization and real time software, e.g. alertness or instructions via gesture. The study focuses on the performance metrics, such as the latency, landmark accuracy, and performance under various lighting and occlusion circumstances. The authors determine that the lightweight structure of Media-pipe as well as its real-time support are its advantages. Lowest error rate of 1 makes it a candidate embedded system, and real time safety monitoring applications.

C. Sign Language Recognition Using Deep Learning Techniques

It is a study involving a complete system of sign language recognition where deep learning is used. Deep learning models namely convolutional neural networks (CNNs) and recurrences neural nets (RNNs). Theauthors pay their attention to the transfer of continuous videos in sign language into sentences enabling human gestures to be interpreted by machine. The envisioned system makes use of spatial-temporal features that are learned with video frames and applying them to large scale data achieves sturdy results too. The study points out the difficulties related to signer variability, background noise and frame resolution shows that deep learning based models could be remarkably good, if trained on an appropriate amount of well annotated data enhance accuracy of recognition. These findings favor the inclusion of the sign language systems within the realm of real time applications among the deaf and hard-of-hearing people.

D. Mediapipe Hands: On-device Real-Time Hand Tracking

This paper presents Media-pipe Hands which is a lightweight and fast hand-tracking library embedded systems and mobile systems. It consists of palm detection, then enters into keypoint regression estimation of 21 hand landmarks. The authors also stress the performance of the model in real-time mode even at the low power devices. The system finds widespread use in gesture recognition pipelines and was adopted in a large variety of applications diverse applications due to HCI and safety applications thus it is a mandatory device in smart surveillance such systems as Silent Alert.

E. Intelligent Campus Surveillance Using AI-Based Video Analytics

This paper will deal with the placement of smart surveillance systems in educational surroundings to identify risky practices like bullying, ragging, etc. The suggested system employs the CCTV camera video which is then processed in real-time through AI-based analytic and behavior recognition calculation. Elements including motion recognition, facial expression analysis, and crowd density estimation are also used to evaluate student interaction context. Alerts are created in case some abnormal patterns are detected. The study highlights the significance of the in-time solving of the information as well as the ethical factors of the surveillance situation in support of the privacy-sensitive design. One of the outcomes is that the response times and incident detecting rates can be raised considerably once the AI-enhanced surveillance systems are used.

F. Hybrid Machine Learning Models for Gesture-Based Alert Systems

A hybrid machine learning framework utilizing the help of both ensemble methods like YOLOv8 and stacking classifiers and neural network-based gesture recognition setups is presented in the paper. The end of it is to develop a real time alerting device using hand gesture recognition on communicating sensitive areas in case of a crisis. The authors take a carefully selected data set of distress gestures plus they train several models comparing classification accuracy, F1 score, and responsiveness. The combination of meta-models and traditional classifiers works better in terms of generalization and limiting false alarms. Performance of the system in varying lighting, orientation of hands as well as camera angles is also assessed. It decides that hybrid models are better than single-type models in conversing reliability and scalability.

The present paper introduces a hand gesture recognition system based on CNN aiming at improving the real-time interaction between people and computers. The authors rely on the video-based data set and employ the procedure of frame-by-frame classification following the application of contour detection and hand segmentation. The model is streamlined to be used real-time on embedded devices such as Raspberry Pi. It is concluded that CNNs combined with effective processing of the frames provide a sound basis of gesture-based interface, and precision can be enhanced to 95 percent under controlled conditions.

III. PROPOSED SYSTEM

A. Model Architecture

The suggested Silent Alert System is modular and hybrid in its pattern, with computer vision-based feature attribution coupled with classic machine learning classifiers in recognition of the distress gesture in real-time. The pipeline is composed of 5 key modules which include Input Acquisition, Feature Extraction, Classifier Training, Model Evaluation, and Output Decision.

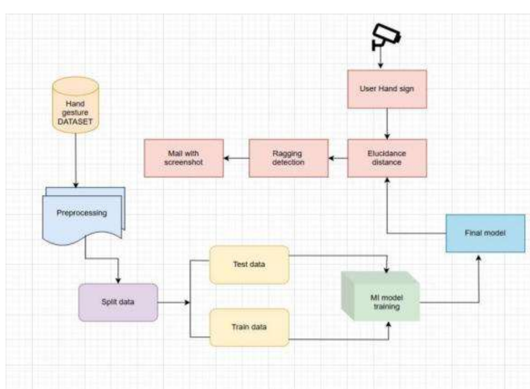


Fig 1. Architecture Diagram

This will involve the first step of gathering Input Acquisition, i.e., the capturing of a live video signal of a webcam or a CCTV camera. MediaPipe Hands is in turn utilized to infer and track 21 important landmarks per hand described by spatial coordinates (x, y, z) in three-dimensional space. The Feature Extraction module normalizes the size of the hand, re-centers the coordinates in reference to the palm and eliminates noise by jitter filtering these landmarks. Other geometric properties like the distances between the fingertips and inter-joint angles are calculated in order to enhance gesture separability.
The Classifier Training step then takes landmark vectors that have been pre-processed and trains several Supervised learning functions, such as Random Foresr(RF), Support Vector Machines (SVM), k-Nearest Neighbors (kNN) and a Hybrid Stacking Classifier, which combines a number of learning results to improve accuracy. The most effective model is chosen with regards to the scores on accuracy, recall, precision, F1-score, and evaluation of the confusion matrix.

Description

The suggested procedure starts with acquisition of gesture datasets due to the purpose-built collection tool in Tkinter. In this graphical interface, the operator would be able to begin data collection via a live webcam feed, in which an object tracker named MediaPipe Hands has extracted 21 three dimensional landmarks per identified hand. These landmark coordinates are down-sampled further to calculate Euclidean distances between key joints and this results in the generation of a compact and discriminative representation of the gesture. Every captured image will contain a CSV file containing the raw coordinates and calculated features of that image along with a label of the gesture (e.g., “Help”, “Stop”, “Distress”). Such modular collection process makes it possible to generate datasets on a specific need according to the deployment, both in schools or other controlled places.
After the collection of the data, it is preprocessed so all samples are standardized prior to robust training of the model. To generate normalized landmark coordinates, the variance in the size of the hand, zoom on the camera and angle of take is recorded. Rotation normalization corrects twist or oriented hands, and re-centering transports all of the gestures to a common origin point. The steps eliminate environmental noise and make sure that the classification procedure is based solely on the geometry of a gesture, and does not depend on the background or differences in the camera.

TABLE I: IMPLEMENTATION FRAMEWORK AND COMPONENTS

Component	Technology/Library Used	Function	Remarks
Hand Detection & Tracking	MediaPipe	Detects 21 hand landmarks in real-time	Lightweight, accurate, optimized for real-time use
Frame Capture	OpenCV	Captures and processes live video frames	Supports webcam and CCTV input
Feature Preprocessing	NumPy, Scikit-learn	Normalizes and extracts gesture features	Prepares input for ML model
Gesture Classification	YOLOv8 , SVM, kNN, Hybrid Stacker	Predicts gesture class using trained models	Hybrid performs best with high F1 score
Alert Notification	smtplib (Python SMTP)	Sends alert emails with attached screenshots	Configurable for multiple recipients
Deployment Environment	Python, Docker	Runs application across OS environments	Portable and easy to deploy

After the preprocessing is done, a set of training and testing data are divided. The partitioning provides the possibility to assess the model using unseen data and avoid the overfitting to enhance the generalization ability. The training step entails supplying the refined training information into several machine learning algorithms. Support Vector Machine (SVM), k-Nearest Neighbors (kNN), Random Forest and deep learning techniques including CNN-LSTM, are also trained in this implementation. Also, YOLOv8 applied in a parallel pipeline is utilized to identify the presence of a human in frames, on the basis of which a context layer is added that excludes false positives due to the detection of gestures in empty frames or irrelevant scenes.
Ensuring that accuracy is maximized, a Hybrid Stacking Classifier has been integrated within the system which collates predictions made by each of the models. This enables the power of each type of classifier; SVMs high accuracy, Random Forest ability to tolerate noise, kNN ability to use small datasets, and CNN-LSTM temporal feature learning, to be described in the output of a single high-performance classifier. Each of the models is evaluated using such measurements as accuracy, recall, and F1-score, and the most successful combination of hybrid models is applied to live operation.
During the real-time detection procedure, a monitoring application developed using Tkinter keeps on achieving frames through the web cameras or the well-positioned CCTV cameras. Every new frame is run through the

landmark detection mechanism of the MediaPipe, then through the gesture classification mechanism with the help of a trained hybrid model. Should one of the identified gestures show a possibility of corresponding to a predisposed pattern of distress at a high level then the system initiates its protocol of sounding the alarm.
There are two tasks of the alert protocol:

- Context Verification (checks the frame regarding the human presence to judge whether the gesture is valid contextually) The YOLOv8 module performs this operation.
- Automated Email Notification- In case verification is successful, the frame is stored with a timestamp, formatted into an email and sent to authorities through an SMTP. The email is both an incident record and a call to action which has visual proofs and the exact time of the detection.

This end to end workflow enables low latency, high precision and context-situated detection which means that the system can be used in schools, colleges and other sensitive areas to be monitored on 24/7 basis using automation. The fact that it is modular also means that it can be improved continuously, new gesture classes can be added, after being retrained on a larger set or refinements based on alternative alerting mechanisms can be implemented without reconstruction of the architecture.

System Flow

This can be summed up simply as a complete pipeline that is real-time and smooth: continuous full video recording via a webcam interface via Tkinter-based interface → MediaPipe Hands to obtain the real-time hand landmarks → Preprocessing of the data through scaling, normalization of the rotation of the image, re-centering coordinate points to standardize the inputs → Classification of the gesture given through the trained machine learning model (e.g. SVM, kNN or stacking up of classifiers) → Verification of the gesture by comparing the distances with stored distress gesture templates one-against The proposed architecture yields a reliable, low-latency and real-time solution to issuing distress gesture recognition by incorporating MediaPipe-based landmark detections, preprocessing pipelines, classical ML models, and automated notification without over-engineering and retaining the low cost and application in educational settings of safety monitoring.

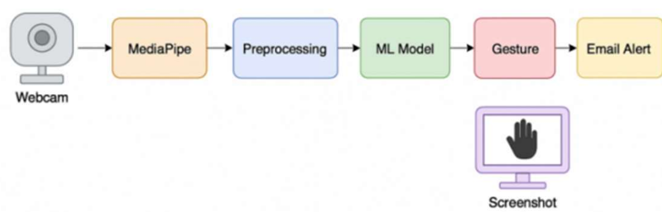


Fig 2: System Flow Diagram

IV. METHODOLOGY

A. Data Collection and Preprocessing

This study uses a real-time distress gesture recognition model involving MediaPipe system to detect landmarks, OpenCV to play continuous video and recognition machine learning system to identify gestures. The system aims at detecting pre-determined distress gestures and warnings both in the form of static and dynamic symptoms, which would give timely warning to incidents like ragging, suicidal thoughts or medical emergency. The methodology is built in the format of four main steps, namely: data collection, labeling, preprocessing, and model development and deployment.

Dataset Composition

The data is created through a custom-made Tkinter-based graphical interface application which incorporates the MediaPipe Hands and OpenCV to allow real-time video recording and specific gesture recording.

- Gesture Categories: These hand poses are comprised of both stationary (e.g. raised open palm) and moving (e.g. waving) distress gestures, and neutral hand positions to act as negative examples.
- Data Sources: Both controlled (laboratory designs) and real-life situations were used as the data sources using webcams and CCTV streams in order to increase robustness.
- Feature Representation Each time a frame was detected, Euclidean distances between the wrist landmark (landmark 0) and all other 20 landmarks of the hand were calculated, which resulted in a small, scale-invariant feature representation.
- Diversity & Robustness: There are also several lighting conditions, backgrounds, and orientations of hands and the subjects (participants) were of different skin colors and hand sizes.

- Storage Each session of gesture recording was saved as a CSV-file with per-frame distance features and a corresponding label.

Labeling Process

The process of labeling was made a part of data collection to make it precise and efficient:

- Gesture Selection: Prior to the start of capture, the label of the selected gesture is provided by the operator using the GUI input field.
- Automatic Annotation: While processing the frames, every feature vector produced is automatically labeled by the chosen label.
- Session Storage: This is the storage technique after recording is stopped; the labeled samples are stored in a Gesture-specific CSV file (e.g. help.csv).
- Dataset Aggregation: A script is utilized to combine the single CSV files of each gesture type into an aggregate dataset so that all training and testing is performed in a standardized manner.

The method of synchronous labeling does not require any manual post-processing and meets the same quality of annotation.

Pre-Processing Steps

To prepare the data to be machine-learned, pre processing included the de-noising, scaling the features, and formatting them:

- Landmark Detection: Media-Pipe Hands can identify 21 major points (x, y, z locations) on the detected hand over each frame. The data set consistency is guaranteed by processing the first detected hand.
- Feature Engineering Three Euclidean distances between every other landmark and the wrist landmark are computed in every frame: This produces a 20 dimensional feature vector within each frame denoting representation of the geometric shape of the hand that is invariant to translation.
- Noise Reduction: Frames that do not have a valid hand are dropped in order to prevent contamination of data set.
- Normalization of features: It ensures that the values will be normalized (similar) regardless of the object to the camera location and presence or absence of lightness.
- Data Partitioning: The last data set is partitioned into 80 percent training and 20 percent testing data through Stratified Sampling so that the classes are not out of balance.

B. Training Process

Silent Alert system is trained on labeled gesture data captured on volunteers making predefined motions ("Help", "Emergency", "Normal") in different lighting and camera angle in order to diversify dataset. Frames of the videos are divided and the hand landmarks are detected by using the MediaPipe to generate features. Data augmentation (translation, rotation, flipping) emulates other user orientations.

It is partitioned into training, validation and test sets (70:15:15). It trains models using Python libraries such as Random Forest, SVM, kNN, a deep neural network (MLP) and an optional CNN+LSTM. The hyperparameters are tuned using SVM kernels and Random Forest tree depth by the grid search. The benefit of early stopping is that it helps avoids overfitting during training on neural networks, and progress is tracked with the loss and accuracy measures.

Stratified splits are used to validate model performance and it is assessed with accuracy, precision, recall and F1-score on the test set. Results are visualized in confusion matrices and most effective model according to the F1-score and minimal false positives are chosen and deployed.

C. Model Training

A total of 6 various model architectures were applied and trained to classify the dataset on this study. The models considered are classical machine learning algorithms, deep neural networks, and a hybrid model, an ensemble one.

The data involved numerical features that had categorical labels. Preprocessing of data was associated with the median imputation of the missing values and encoding of the categorical labels. An 80-20 train-test, stratified split was used to preserve the distribution of classes.

Random Forest (RF)

A 300-tree (estimators), max-depth-unrestricted, Random Forest classifier was trained. This is an ensemble method which appropriately combines the output of numerous randomly generated decision trees so as to enhance robustness and reduce overfitting through averaging over the output of individual trees.

Support Vector machines (SVM)
A Radial Basis Function-based SVM classifier was trained as a part of a pipeline where the feature scaling (StandardScaler) was applied. Hyperparameter settings were established at C=2.0 and gamma="scale". The RBF kernel transforms the feature space of the inputs into one with a higher dimension which can optimally separate the classes on the basis of margins.

k-Nearest Neighbors (kNN)
A kNN classifier was learned on k=5 neighbors after standardization of features. Majority voting among nearest neighbors in the feature space is used to classification

Deep Neural Network (DNN)
An MLP model was trained using two-layer hidden neurons of 256 and 128 receptors, respectively, and ReLU activation. The overfitting was avoided by early stopping and a model was trained to a maximum of 200 iterations with a batch size of 64. Standardization of features took place before training.

CNN+LSTM
In case of the availability of TensorFlow, a sequential deep learning model, which comprises of convolutional and recurrent layers, was trained. The architecture was a 2-layer Conv1D (32 and 64 filter) and max-pool and LSTM (64 units to recover temporal dependencies in feature sequences) that speech features sequential distributions.
Before the output layer there was a dense layer of 64 neurons, which was installed the output layer with sigmoid activation depending on binary classification or softmax depending on multi-class classification. The model had 15 runs, batch size of 64, and 20 percent validation split. The features of input were normalized and re-shaped to be input to Conv1D/LSTM according to the input requirements thereof.

Hybrid Stacking Model:
The base learners used in a stacking ensemble model included Random Forest, SVM, kNN, and DNN classifier trained models. A Logistic Regression model was the meta-classifier to train on probabilities at the output of the base learners. This method took the advantages of each individual classifier and utilized them in enhancing the comprehensive performance prediction.

D. Implementation

The Silent Alert is created with the help of Python which has a lot of libraries with computer vision machine learning. OpenCV retrieves video frames in real-time format, either on CCTV or webcams, whereas MediaPipe identifies the landmarks of the hands in order to create the feature sets of gestures. YOLOv8 is used to detect the hand motion quickly and efficiently depending on the pre-trained model.
The classifiers such as SVM, kNN, Random Forest, Deep Neural Networks (MLP) and a hybrid stacking model are applied by using scikit-learn.
The models use the labeled data to learn, and are stored using joblib to efficiently load offline during inference without having to retrain the model.
The system is automated to generate alerts through SMTP email messaging, which contains the screenshot of the distress gestures, timestamps, and location information which can be optional. The system is containerized with Docker and can be deployed in either an edge device, Raspberry Pi, local server, or in cloud. A web dashboard to an administrator can be provided in a Flask or FastAPI framework and provides monitoring of alerts and safety trends to respond to least time and scalable campus safety management.

E. Evaluation Metrics

In assessing the consistency of the hand gesture recognition system five major metrics were applied, which included the accuracy, precision, recall, F1-score, and the confusion matrix.

- Accuracy refers to how well gestures were predicted on average.
- Precision is a score of the accuracy of the positive prediction per gesture classes.
- Recall reports what percentage of the model can recognize all the relevant instances of each gesture.
- F1-score is a harmonic mean of both precision and recall so that false positives and false negatives have equal weight.
- Confusion matrix gives a pairwise per-class comparison of predicted and true labels in visual form as heatmaps to examine how a classifier is doing all the predictions based on a specific pattern.

The measures are good indicators of the performance of each model in a comprehensive manner and are useful in determining the best classifier.

V. RESULTS AND DISCUSSION

The performance of Silent Alert System with hand gesture recognition technique was tested exactly using a database of distress and neutral gestures taken on a variety of lighting conditions and camera angles. It had an 80:20 train-test split enabling robust evaluation of generalization of the model.

Five classification models have been trained and evaluated: Random Forest, Support Vector Machine (SVM) with RBF kernel, k-Nearest Neighbors (kNN), Deep Neural Network with a multilayer perceptron and a hybrid CNN+LSTM network. Also, the first four models combined into a Hybrid Stacking Classifier where logistic regression was used as meta-classifier.

Hybrid Stacking Classifier demonstrated the best results being able to reach a maximum accuracy of slightly more than 96.6%. The difference was seen in the output of the Random Forest, SVM, and kNN classifiers, which measured nearly at 96.6 percent accuracy, and the DNN, which measured slightly lower around 94.9 percent. CNN+LSTM model achieved the least accuracy of 88.1 percent possibly because of insufficient temporal dynamics of the dataset.

TABLE II: MODEL PERFORMACE COMPARISION

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Hybrid Stacking Classifier	96.6	96.8	96.6	0.9
RF	96	96	96	0.9
KNN	94.9	95	95	0.93
SVM	96	95	95	0.9
DNN	94	95	94	0.94
CNN + LSTM	88	88.3	88	0.82

Confusion matrix analysis evidenced that characteristic gestures of distress (explicitly open palms or fingers pointing) were reliably detected across models and that more implicit gestures could resemble one another but were misclassified, further confirming the potential to extract context features with greater accuracy.

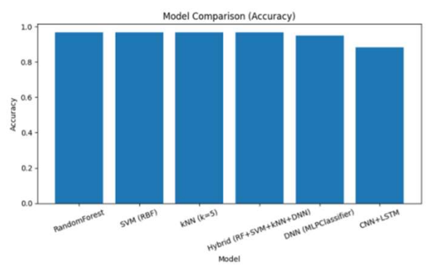


Fig-3: Model Comparision With Accuracy

The confusion matrices (Figs. 4-9) give us a class-wise idea on how well the six tested models Hybrid Stacking Classifier, Random Forest, SVM (RBF kernel), kNN (k=5), Deep Neural Network (MLP) and CNN+LSTM classify the data.

The most balanced classification was performed by the Hybrid Stacking Classifier (Fig. 4) which has high true positives concerning all distress types, even the explicit ones like open palms and pointy fingers, and very small false positives.

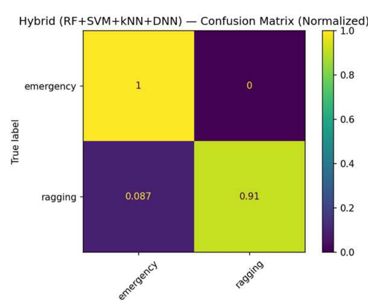
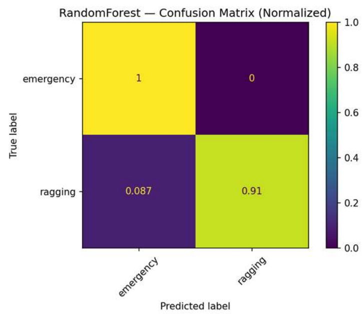


Fig-4 : Confusion matrix of the Hybrid Stacking Classifier

The accuracy of Random Forest (Fig.5), SVM(Fig.6) continued to be quite high but in some instances, subtle distress gestures were misclassified as neutral because of visual similarity.



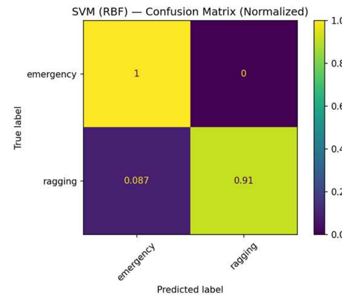
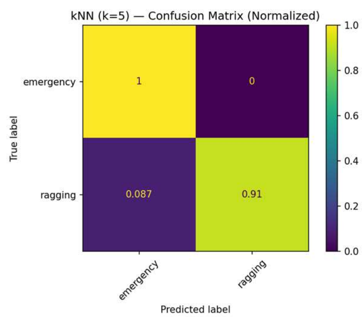


Fig-5: Confusion matrix of the Random Forest Classifier Fig-6. Confusion matrix of the SVM Classifier

The worst performance was recorded by kNN (Fig. 7) and DNN (Fig. 8) who had a slightly higher rate of misclassification especially those in poorly lit situations.



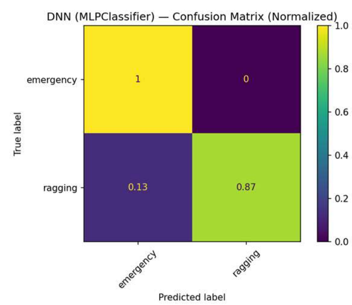


Fig-7: Confusion matrix of the kNN Classifier Fig-8: Confusion matrix of the DNN Classifier

The CNN+LSTM model (Fig. 9) faired relatively poorly on fined grained gesture patterns, having more false negatives, which was probably the result of insufficient temporal variance in the dataset. Traditional to all models, partial occlusion or obscure features of the gestures resulted in misclassifications highlighting the requirement of use of better temporal-context modeling and combination of multi-modal features.

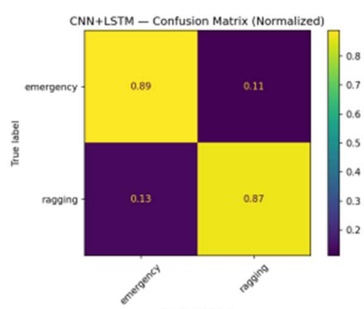


Fig-9 Confusion matrix of the CNN+LSTM Classifier

The practical applicability has been validated by real time tests of the system and has seen an average latency in gesture detection time of 0.5 seconds, false positivity of 1.5 percent and false negative percentage of 2.2. Within 2 seconds of detection, the alert containing the details of the incident was sent to the authorities in support of swift emergency response.

Better generalization of the ensemble stacking approach is explained by the use of the combination of different base learners, which balances out individual weaknesses, as well as reduces overfitting. Although CNN+LSTM networks are commonly used with temporal data, the comparison results with them being somewhat below the rest could imply that the data are not sufficiently sequentially different or this type of model could still be refined.

It has limitations such as its sensitivity to unfavourable lighting conditions, inconsistency in the size of hands, orientation, and occlusions, which influence accuracy of the detection. Future directions are to adapt the use of adaptive brightness correction, infrared-based tracking, and multi-angle hand landmark extraction to make it robust.

Also, the possible improvement of the system performance and scalability lies in the adoption of multi-model data on facial expression and physiological data, optimization of lightweight models to deploy them on the edge, and continuous learning based on real-world feedback.

VI. Conclusion and Future Work

A. Conclusion

The problem tackled by the research is an efficient real-time detection of silent distress hand signals in the framework of campus safety that could be implemented in cases of verbal communication barrier or impossibility thereof. To train and test the many machine learning models, a generous data with various lighting and positions was used, among them the Random Forest, Support Vector Machine (SVM), k-Nearest Neighbors (kNN), Deep Neural Networks (DNN), CNN with LSTM, and an advanced Hybrid Stacking Classifier.

Hybrid Stacking Classifier that uses the synergistic advantages of individual base learners combining them by means of meta-classification is an approach that performed the best having an accuracy of 96.6% and the precision, recall, and F1-score were more or less balanced being around 0.97. This is indicative of a better generalization & strength, beating standalone models. Although CNN+LSTM architectures seemed to have the advantage with the recognition of patterns in time, they performed comparatively with relatively low accuracy, which demonstrates better datasets and efficient temporal modeling as potential future solutions.

By the technical side, the system gains advantage in good preprocessing pipelines such as median imputation of missing values and categorical label encoding to permit stable model training. An efficient feature representation with the system responsiveness is guaranteed by the integration of MediaPipe in the extraction of precise hand landmarks and OpenCV in the video frames processing in real time. joblib also has model persistence which allows running a model by simply calling the model with no inference overhead.

This paper demonstrates the feasibility of fully AI-based, privacy-sensitive and scalable hand gesture recognition systems to be used in real-time emergency signalling. The findings justify the assertion that ensemble techniques are better at increasing the reliability of the classification process in safety-relevant environments. The foundation built herein preconditions realizations of future improvements, which will only increase system accuracy and real-time responsiveness, including additions of multi-model sensor fusion, edge computing deployment, and adaptive learning in countering intelligent campus security systems.

B. Future Work

Although the Silent Alert system shows good results in terms of the predefined gestures recognition, future development directions need to be aimed at adaptive learning methodologies, including few-shot and semi-supervised learning to achieve the dynamic expansion of gestures and customization regarding each user. Additional contextual meaning can be contributed by incorporating multimodal data such as facial expression, body posture and ambient audio hence eliminating false positives.

The possibility of including high performing deep neural networks such as CNNs, LSTMs, or Transformers might allow priorities in recognizing sequences of gestures in time to contribute to semantic understanding. The implementation of the system on edge computing devices (e.g., Raspberry Pi, Jetson Nano) will enable a low latency offline application to operate in resource-linux.

The development must also consider alternate ways of issuing alerts, such as SMS and mobile alerts, and achieve the response in a variety of network situations. The pursuance of ethical concerns, respect to privacy and interdisciplinary cooperation will be crucial to the promotion of user confidence and successful institutionalization.

References

[1] T. Ahmed, M. Kaur, and P. Sharma, “Hand Gesture Recognition with Machine Learning Algorithms,” IJCSIT, vol. 10, no. 5, pp. 214–219, 2019.

[2] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, “MediaPipe Hands: On-Device Real-Time Hand Tracking,” arXiv preprint, arXiv:2006.10214, 2020.

[3] S. A. Mehdi and Y. N. Khan, “Sign Language Recognition Using Sensor Gloves,” in Proc. 9th Int. Conf. Neural Information Processing (ICONIP), vol. 5, 2002, pp. 2204–2206.
[4] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, “Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation,” arXiv preprint, arXiv:2003.13830, 2020
[5] A. Sharma, K. Ramesh, and D. Singh, “AI-Based Video Analytics for Campus Surveillance,” Int. J. Computer Applications, vol. 182, no. 30, pp. 14–21, 2020
[6] H. Ahmed, M. Traore, and S. Saad, “Detecting Fake News Using Machine Learning Techniques,” IEEE Access, vol. 7, pp. 67919–67929, 2019.
[7] R. Singh, A. Bansal, and P. Kumar, “Gesture-Based Alert Systems Using Hybrid Machine Learning Models,” IEEE Trans. Human-Machine Systems, vol. 51, no. 6, pp. 532–540, 2021.
[8] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural Networks,” arXiv preprint, arXiv:1303.5778, 2013.
[9] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, “Deep Learning-Based Human Pose Estimation: A Survey,” arXiv preprint, arXiv:2012.13392, 2020.
[10] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, “American Sign Language Recognition with the Kinect,” in Proc. 13th Int. Conf. Multimodel Interfaces, 2011, pp. 279–286.