IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Border Security System Using Face Detection Algorithm

Srilakshmi Aouthu
Department of ECE ,
Vasavi College of Engineering ,Hyderabad

Ismail Shaik
Department of ECE , MRE
Hyderabad Osmania University

Abstract – This paper introduces a GSM-based border security system that integrates Raspberry Pi, Arduino Uno, and machine learning algorithms to enhance surveillance and threat detection along border areas. The system relies on cameras as primary input devices, with video feeds processed in real time by the Raspberry Pi. Advanced machine learning algorithms analyze the visual data, accurately detecting and classifying unauthorized activities or intrusions. The Arduino Uno manages hardware-level operations and facilitates communication between system components. Upon detecting a potential security breach, the system activates an alert mechanism. A GSM module transmits instant notifications—including images, video evidence, and location details—to border security personnel, ensuring a swift response and reducing the need for human intervention. This proposed solution is cost-effective, scalable, and fully automated, improving situational awareness and reinforcing national security infrastructure by integrating machine learning with GSM communication and embedded hardware.

Keywords— Raspberry pi V4, Arduino Uno, Geo tagging, Intruder Alert, IoT, Machine learning**.**

I. INTRODUCTION:

Globalization and Border Security Challenges Globalization has made the world more interconnected, facilitating trade and the movement of people across borders. As a result, traditional physical barriers between countries are becoming less effective. Instead, modern concepts like "Smart Borders" and "Smart Walls" are gaining prominence.

India shares a vast 15,000 km border with seven neighboring countries, spanning diverse terrains and extreme weather conditions. Thousands of soldiers patrol these regions, often enduring harsh environments that put their lives at risk. To enhance efficiency and safety in border security, India’s Ministry of Defense introduced the Comprehensive Integrated Border Management System (CIBMS)—a high-tech solution aimed at strengthening border management.

Comprehensive Integrated Border Management System (CIBMS)

CIBMS is designed to modernize border security through its D4R2 approach—Deter, Detect, Discriminate, Delay, Respond, and Recover. This system integrates advanced technologies such as CCTV cameras, thermal imaging, night vision, radars, laser beams, and underground sensors to monitor borders across land, air, water, and tunnels. Additionally, it connects security forces through smart networks and intelligence tools, enabling quick and informed decision-making.

By leveraging technology alongside human efforts, CIBMS reduces the reliance on constant physical patrols, enhances soldier safety, and makes border security more efficient. The Indian government has implemented CIBMS in areas where physical fencing is impractical, replacing traditional barriers with cutting-edge surveillance technologies. One such initiative, BOLD-QIT (Border Electronically Dominated Quick Response Interception Technique), integrates microwave communication, optical fiber cables, cameras, and intrusion detection systems to reinforce border security.

II. Literature Survey

This section reviews prior research to explore the advantages of Smart Borders and the use of technology in border surveillance and management.

- Intrusion Detection Systems: Paper [3] surveys various intrusion detection systems in smart environments, highlighting their benefits and anomalies. Paper [4] presents an infrared sensor-based border security system, though it suffers from false-positive reports.
- Automated Border Defense: Paper [5] proposes a GPS and satellite-based automatic weapon activation system for warfare scenarios. Paper [6] discusses IoT-enabled intruder identification frameworks using IR sensors.
- Surveillance Technologies: Paper [7] introduces a robot with an IR sensor and Pi camera to identify terrorists and send notifications. Paper [8] explores the use of thermal imaging cameras (FLIR) for object and infiltrator detection.
- Human Detection and Rescue Operations: Paper [9] presents a human thermal infrared acquisition platform for accurate target identification. Paper [10] proposes an algorithm for a rescue robot to navigate collapsed areas. Paper [11] focuses on a home security system using Raspberry Pi and IoT, while Paper [12] demonstrates path-learning algorithms for efficient rescue operations.
  The paper first presents a simple diagram illustrating the system's design, followed by an overview of its functionality. It then provides a step-by-step explanation of its operation and concludes with the actual hardware setup, detailing the system's construction and implementation. Block Illustration

The block diagram illustrates the fabrication and construction of the system

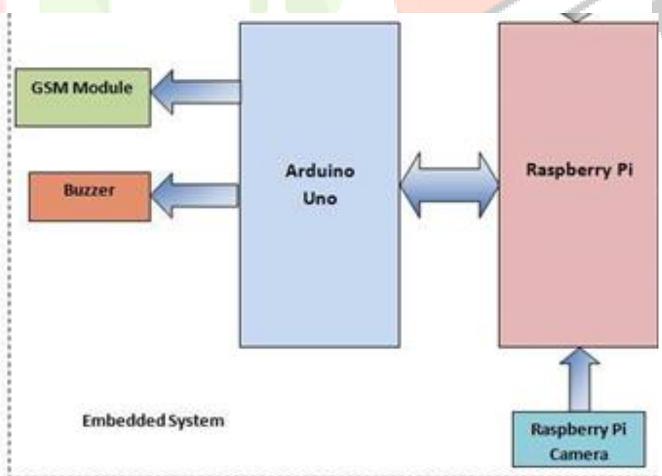


Fig 1: Block Ilustration of the proposed technique

III. Methodology

A. System Overview

The system employs a Pi Camera connected to a Raspberry Pi to monitor the border and stream live video to a server at the Base Station. Upon detecting a person, it first determines whether they are wearing a uniform. If a uniform is detected, a face recognition algorithm verifies their identity against stored records. If the face matches, the system continues monitoring without raising an alert. However, if the person is not in uniform or their face does not match the database, an alarm is triggered to notify security of a potential intrusion. This approach effectively distinguishes intruders from authorized personnel.

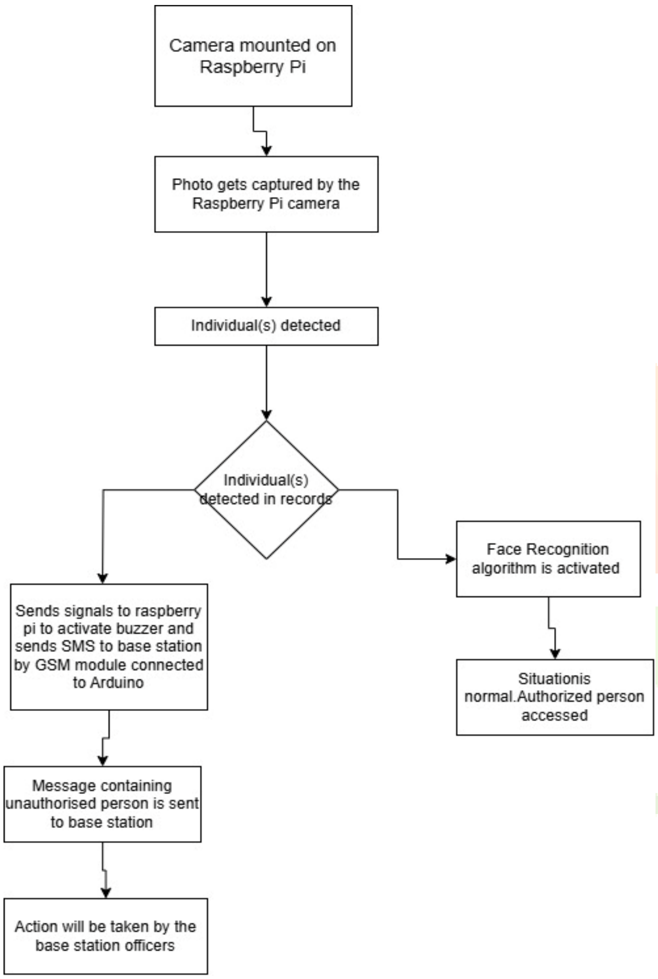


FIG 2: System overview of the proposed technique

B. Detailed Methodology

The detailed methodology explains the Intruder Alert System followed by steps in Training of Models in machine learning algorithms to detect the intruder.

1)Intruder Alert System

1. The system uses a NoIR Pi Camera with a Raspberry Pi, which is suitable for night vision and low-light conditions. It captures images and video to detect movement.

2. The camera sends the video stream to a machine learning model stored on a server at the army base station

for object detection.

3. The video can also be monitored in real-time for security checks. The Raspberry Pi helps stream the video to the trained model on a PC (server) to identify intruders

4. A GSM Module (SIM 800L) connected to an Arduino board sends an SMS alert or email notification to security personnel if an intruder is detected.

5. When an intruder is detected, the photo and timestamp are saved to the Google Firebase cloud service for future reference.

6. The captured photo is encrypted using the AES encryption technique, ensuring security with a unique cryptographic key for each photo.
7. The encryption key is also stored on the cloud for later decryption.

8. Only designated officials have access to the cloud account to verify and authenticate photos when need.

2) Training of the model

Two machine learning models, Faster R-CNN Inception V2 COCO and EfficientDet D0 (512x512), were tested to determine the best performance. The training process followed these steps:

1. Machine Learning Setup: TensorFlow Object Detection API was used to train the models on a PC.
2. Pre-Trained Models: TensorFlow offers various datasets; this system used the COCO dataset for Faster R-CNN.
3. Custom Dataset Creation: A dataset was built using 300 images, including 200 soldiers in uniform and 100 intruders collected from Google and real-time sources.
4. Face Recognition: The same dataset was used for detecting soldiers' faces.
5. Data Splitting: The dataset was divided 80% for training (240 images) and 20% for testing (60 images).
6. Image Processing: Images were resized and labeled as "Soldier" (if in uniform) or "Intruder" (if in civilian clothing).
7. Labeling Format: The PASCAL VOC format was used to label images, saved as XML files.
8. CSV & TF Records: XML files were converted into CSV files for training and testing, then TF Records were created for model input.
9. Label Map & Config File: A label map defining object labels was created, followed by a training config file for model selection.
10. Model Training & Results: After training, the Faster R-CNN Inception V2 COCO model performed best, as shown in Table 1.
11. Final Model Export: The trained model's inference graph was exported for real-time detection.

C. Model Training Flowchart

The flow chart in Figure 3 represents the training model methodology as implemented in the present system. The same sequence has been employed for EfficientDet D0 512*512 model

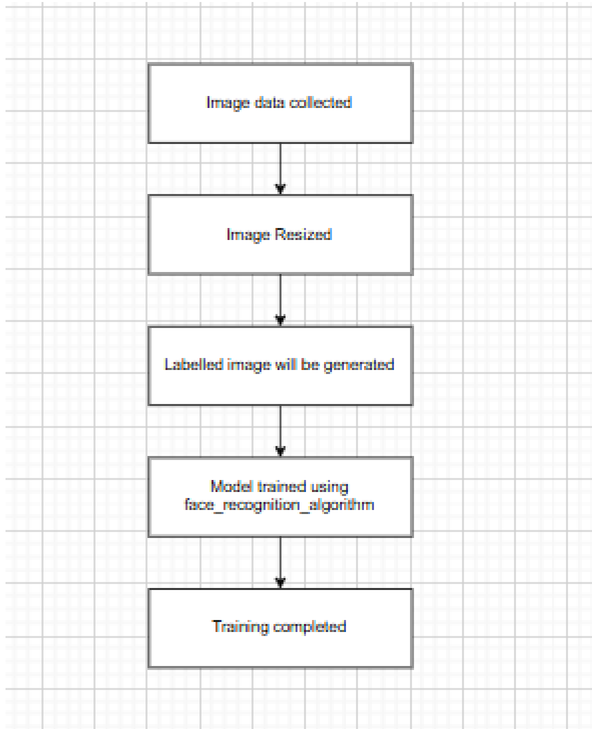


Fig 3: flowchart of the trained model

IV. RESULT ANALYSIS

1) Comparison faster_rcnn_inception_v2_coco model with EfficientDet D0 model

1. First model was trained using EfficientDet D0 512*512 model using the steps mentioned in Figure 2.
2. Total 2800 steps were initialized for training and when it was found that loss parameters were not reducing training was stopped. Thereafter, inference graph was generated which helps to create the working procedure of the model. Figure 14 demonstrates the loss and accuracy for the model trained using EfficientDet D0 512*512.
3. The maximum loss was above 3 as can be seen from Figure 13 and the minimum loss was around 0.9.
4. During the training, it periodically saved a checkpoint every five minutes. This checkpoint has been used to export inference graph (.pb file) that contains graph variables frozen as constants. Training took around 3 hours using NVIDIA GEForce GT 710.
5. The second training was performed using faster_rcnn_inceptionV2 model with a predefined initial learning rate of 0.0002 in the model file and decaying after define number of steps. However, for this case, the training steps were within the limits of initial learning rate steps so it won't decay as shown in Figure 13.
6. Total 3200 steps were initialized and after that training was stopped as loss was acceptable.
7. The minimum loss was 0.02 and maximum was 1.7 as shown in Figure 12.
8. Thus, faster_rcnn_inception_v2 model performed better than EfficientDet D0 512*512 model as this had better loss parameters. This model took around 2 hours for training. Hence, it is better not just from both loss perspective but also operates faster. This justifies its adoption for deployment in both intruder detection and face detection algorithm.

2) Intruder Detection Algorithm Results

The Faster R-CNN Inception V2 model was tested on images from the dataset, and it accurately detected objects with bounding boxes and prediction percentages displayed.

The model achieved over 90% accuracy, correctly identifying 8 out of 10 images

It was also tested on live video streaming, where it successfully detected intruders in real time (Figures 8a & 8b).
SMS alerts were sent promptly by the GSM Module, ensuring security officers were informed immediately.

Intruder images (both normal and disguised in army dress) were saved on the cloud after being encrypted with a unique key using AES encryption, along with timestamp and location details.

A Confusion Matrix (Figure 6) was used to calculate the model's precision and recall, confirming its reliability.

3) Face Detection Algorithm Results

1. To prevent imposters from tricking the system by wearing a uniform, a face detection algorithm was used with the Faster R-CNN Inception V2 COCO model to verify soldiers' identities.
2. The model was trained for 1600 steps, achieving a maximum loss of 1.3 and a minimum loss of 0.0317.
3. The algorithm accurately detects soldiers' faces, distinguishing between a real soldier (Soldier X) and an intruder.
4. Figure 9c shows the correct stepwise implementation of face detection after intruder detection.
5. Real-time testing showed successful results, with Figure 9a & 9b showing face detection in live video, and Figure 9c displaying the system overview from Figure 2.
6. The system also successfully identified a specific soldier (Soldier Y) among three individuals in uniform.

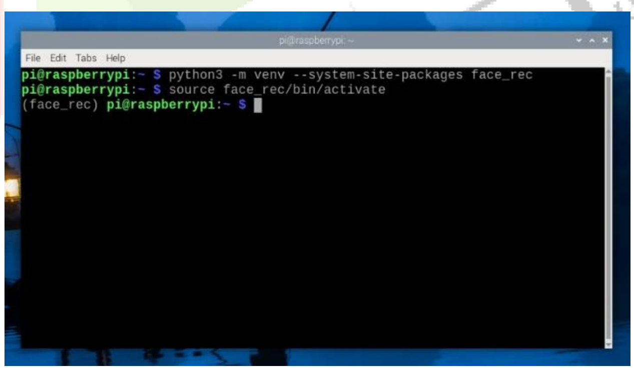


Fig 4: Creating a virtual environment

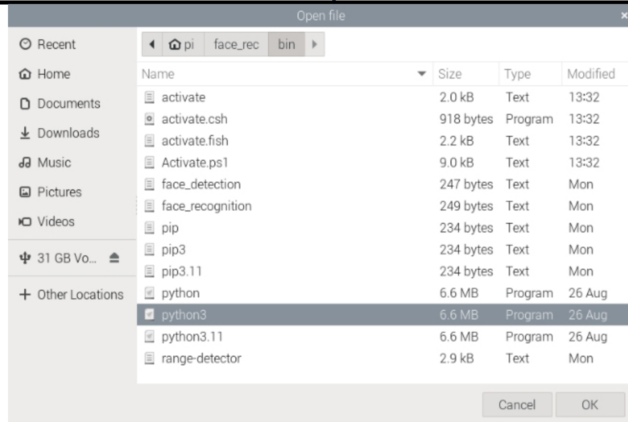


Fig 5: Thonny select environment

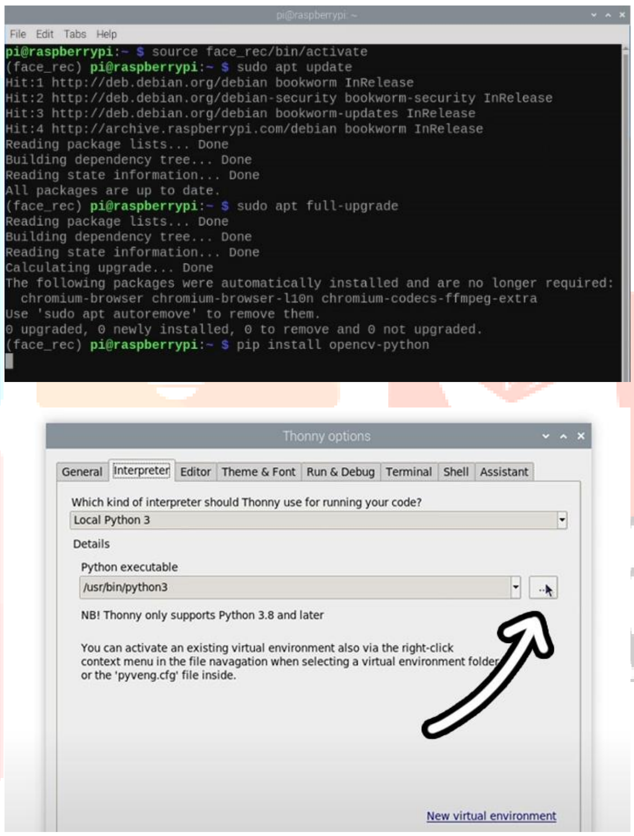


Fig 6: CLI installation

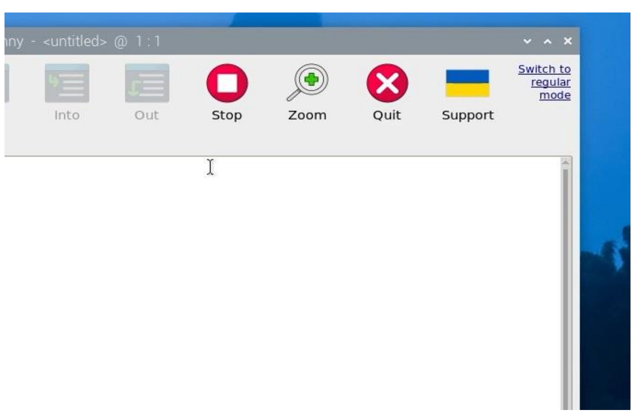


Fig 7 : Thonny exit initialization

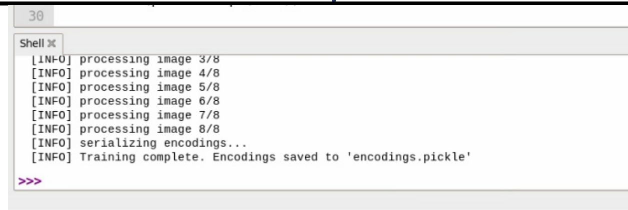


Fig 8: Training the model

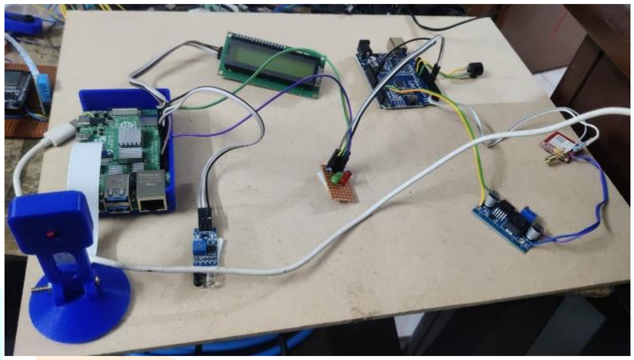
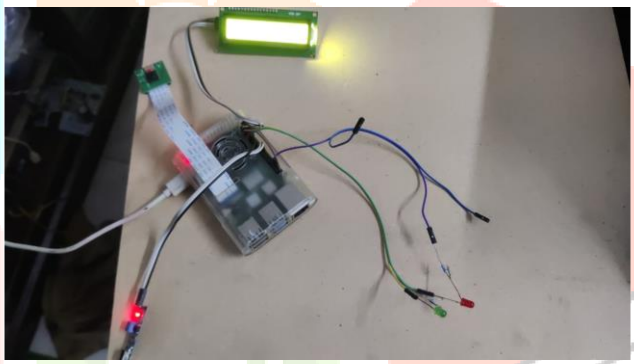


Fig 9: Hardware Setup of the project



Key Features of the System (Simplified)

1. Wireless Border Security – A cost-effective solution for monitoring borders without heavy armed patrols, especially in harsh terrains.
2. Advanced Machine Learning – Uses the best-performing model after testing multiple ML algorithms.

3. Strong Data Security – Only authorized officials can access data using encryption and limited access controls.

4. Customizable & Scalable – Can be upgraded with modern sensors and larger datasets to further improve accuracy and performance.

V. CONCLUSION

The proposed GSM-based border security system, integrating Raspberry Pi, Arduino Uno, and machine learning, offers a reliable and efficient solution for real-time border surveillance. By utilizing camera-based monitoring and intelligent data processing, the system effectively detects potential threats while minimizing false alarms. The Raspberry Pi handles image processing and machine learning analysis, while the Arduino Uno manages hardware components, ensuring smooth system operation and communication. The integration of GSM technology enables instant alerts and real-time communication with security personnel, improving response time in critical situations. With its modular design, cost-effectiveness, and scalability, the system is well-suited for deployment in remote border areas where continuous human supervision is impractical. In conclusion, this project presents a practical and automated approach to enhancing national border security, combining accuracy, efficiency, and advanced technology to strengthen surveillance and threat detection.

REFERENCES

[1] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," J. Cloud Comput., vol. 7, no. 1, pp. 1–20, 2018, doi: 10.1186/s13677-0180123-6.

[2] R. Karthick, A. M. Prabaharan, and ..., "Internet of things based high security border surveillance strategy," … Appl. Sci. …, no. May, 2019,[Online].

[3] M. Of, B. Security, and U. Iot, "Original Research Paper Engineering," vol. 2, no. 3, pp. 18–19, 2018.

[4] D. H. M. Ishra, D. E. K. Umar, and V. A. S. Inghal, "Border Security System using Arduino & Ultrasonic Sensors," vol. 06, no. 13, pp. 2489–2492, 2017.

[5] K. B. Sri, K. Sachdeva, S. Nikita, and D. C. Umayal, "Border Security Robot using IoT," Int. J. Adv. Res. Methodol. Eng. Technol., vol. 2, no. 3, pp. 1–6, 2018.

[6] D. A. Lshukri, L. Vidhya Lavanya, E. P. Sumesh, and P. Krishnan, "Intelligent border security intrusion detection using IoT and embedded systems," 2019 4th MEC Int. Conf. Big Data Smart City, ICBDSC 2019, pp. 1–3, 2019, doi: 10.1109/ICBDSC.2019.8645587.

[7] Research on the Multiple Factors Influencing Human Identification Based on Pyroelectric Infrared Sensors, Junwei Yan, Ping Lou, Ruiya Li, Jianmin Hu, and Ji Xiong, 2018, 604.

[8] Miyama, S.; Imai, M.; Anzai, Y.; "Rescue robot under disaster situation: position acquisition with Omni-directional video footage sensor detection and robot movement Intelligent Robots and Systems, IROS, 2003, vol.3, pp. 3132 - 3137.

[9] Aishwarya D, Dr. J Arokia Renjith, Enhanced Home Security Using IOT and Raspberry, IRJET, 2017, Volume: 04 Issue: 04, 3155-3158.