© May 2025 | IJIRT | Volume 11 Issue 12 | ISSN: 2349-6002

A Survey on Advanced Surveillance System for Intelligent Threat Detection

Podili Muni Jyothieswar[1], Pilli Charitha[2], Chinthala Pavani[3], Erugu Yekalavya[4], Koduru GunaSekhar[5], Nemallapudi Sukumar[6]

Department of Computer Engineering, Gokula krishna college of Engineering Sullurpet, AP, India

Abstract: Smart surveillance systems have undergone remarkable advancements with the integration of deep learning techniques, significantly enhancing their ability to monitor and secure environments. These systems use artificial intelligence to detect, track, and analyze human activities in real time, ensuring heightened security, improved efficiency, and automation. By leveraging deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), these systems can accurately recognize patterns, detect anomalies, and respond to potential threats autonomously with minimal human intervention. The adoption of deep learning in surveillance has led to substantial improvements in real-time threat detection, facial recognition, and behavioral analysis. These systems are widely applied in public safety, traffic monitoring, and crime prevention. However, they also pose challenges related to computational costs, privacy concerns, and the necessity of high-quality video feeds. This report explores the architecture, implementation, benefits, and drawbacks of deep learning-based surveillance systems, emphasizing their transformative role in modern security infrastructure.

Keywords—Artificial Intelligence (AI), Deep Learning (DL), Image Processing (IP), TensorFlow, Single Shot Multi-Box Detector (SSD), Raspberry Pi.

INTRODUCTION

The escalating threat of armed robberies, characterized by their severe physical and psychological impact on victims, necessitates robust and efficient surveillance systems. Unlike simple robbery, these violent crimes often result in significant trauma and even loss of life, underscoring the urgent need for enhanced security measures. This paper presents the design and implementation of an Intelligent Surveillance System (ISS), a cost-effective yet high-performance solution aimed at mitigating the risks associated with such incidents.

The ISS utilizes readily available hardware components, including a Raspberry Pi, a Raspberry Pi Camera, and a Passive Infrared (PIR) Sensor, to achieve comparable functionality to traditional, more expensive surveillance setups. The system operates by initiating a continuous monitoring process via the camera. A software script actively detects any movement within the camera's field of view. Upon detecting motion, script triggers the capture of a series of consecutive images, which are then compressed into a zip file and uploaded to a cloud server.

On the cloud server, another software script monitors for new file uploads. Upon receiving a zip file, the script extracts the captured images and applies Image Processing techniques to enhance their quality. These processed images are subsequently fed into a pre-trained Deep Learning model, specifically designed for object detection. The model analyzes the images to identify potential weapons. If any weapons are detected, the system promptly notifies the user via email and SMS, alerting them to review the surveillance footage and contact law enforcement.

This automated, cloud-based approach ensures real-time threat detection and rapid response, minimizing the potential for harm. By leveraging affordable hardware and advanced AI, the ISS offers a practical and effective solution for enhancing security in environments prone to armed robberies, contributing to increased public safety and reduced trauma for potential victims.

LITERATURE SURVEY

One of the first discussions on motion detection in security systems is by B. Jadhav in his paper "Motion Activated Security System" (IRJET, 2016). Jadhav introduces a motion-based security system using PIR sensors and Raspberry Pi. This system provides real-time monitoring with automatic alerts whenever movement is detected, making security monitoring more efficient and less reliant on human intervention.

Rahman (2008) in "Motion Detection for Video Surveillance" talks about how motion detection is used to enhance video surveillance. Rather than relying on continuous human monitoring, the system automates the detection of movement in video feeds. This ensures that any potential threat is quickly noticed, making surveillance much more responsive. Another interesting aspect of modern motion detection systems is the use of IoT. R. Chandana (2015) in "Smart Surveillance System using Thing Speak" discusses the integration of cloud platforms such as ThingSpeak for real-time monitoring. This setup not only detects motion but also sends alerts over the internet, ensuring that users are notified wherever they are. This cloud-based solution increases the accessibility and flexibility of surveillance systems.

The hardware components such as the Raspberry Pi 3 Model B and Raspberry Pi Camera play a crucial role in the functioning of many motion detection systems. The Raspberry Pi 3 Model B (Raspberry Pi, 2025) is a compact, cost-effective, and powerful tool for creating a motion detection system. It can run algorithms, process data from the sensors and cameras, and send alerts to users.

For motion detection, PIR (Passive Infrared) Sensors are commonly used. These sensors detect changes in infrared radiation when objects move in their vicinity. According to Homemade Circuits (2021), these sensors are very effective in detecting human movement and are easy to integrate into Raspberry Pi-based systems.

Machine learning has taken motion detection to the next level. TensorFlow offers pre-trained models that make implementing object detection easy. V. Rathod (2021) highlights the availability of these models in the TensorFlow Model Zoo, which developers can use to recognize objects in videos, providing smarter surveillance. Additionally, Aidouni (2019) emphasizes that to evaluate these models, it’s crucial to look at metrics like Mean Average Precision (mAP) and Intersection over Union (IoU). These metrics measure the accuracy of the models in detecting objects and help improve their reliability.

Metrics like mAP and IoU are essential in evaluating the performance of object detection systems. According to Kukil (2022), mAP gives us an idea of how precise and reliable a model is by calculating its precision and recall. On the other hand, IoU helps assess how much overlap exists between the predicted and actual locations of objects (Baeldung, 2022). These two metrics are key to improving and comparing the efficiency of motion detection models.

motion detection technologies have advanced significantly with the integration of affordable hardware, cloud platforms, and machine learning. The ease of access to tools like Raspberry Pi and TensorFlow has empowered developers to create efficient and cost-effective surveillance systems. As AI and edge computing technologies continue to evolve, the future of motion detection looks promising, with more intelligent and responsive systems likely to emerge.

METHODOLOGY

The complete design of the proposed system is discussed in this section. The proposed method takes the video as an input and applies a different operation to find the solution to the problem. The flow diagram of the proposed system is shown in Figure The following sections describe the main components of the proposed system.

A. Hardware Configuration

The hardware configuration of the proposed intelligent surveillance system is theoretically based on several key components. At the core is the Raspberry Pi 3 Model B, envisioned as a low-cost microcomputer capable of connecting to a display or being accessed remotely. The system also incorporates the Raspberry Pi Camera Module v2, intended for capturing high-quality video and still images. A Passive Infrared (PIR) sensor is included for motion detection, designed to detect human movement within its specified range. Finally, a Wi-Fi source, such as a mobile hotspot or modem, is necessary to provide network connectivity for data transmission.

TABLE I. HARDWARE COMPONENTS

No.	Component	Model/Size/Range
1	Raspberry Pi	Model B Board/1 GB RAM
2	Raspberry Pi Camera	Module v2
3	PIR Sensor	25cm - 20m
4	A source of Wi-Fi	Mobile Hotspot/Modem

In theory, the Raspberry Pi functions as the system's central processing unit, coordinating signals from the PIR sensor and controlling the Raspberry Pi camera. A Python program is meant to be implemented on the Raspberry Pi to manage the interaction between the PIR sensor and the camera. This program is designed to monitor the PIR sensor for motion detection, control the camera to capture images when motion is detected, zip the captured images, upload the zipped file to a cloud service using an API token, and manage storage by deleting older image files.

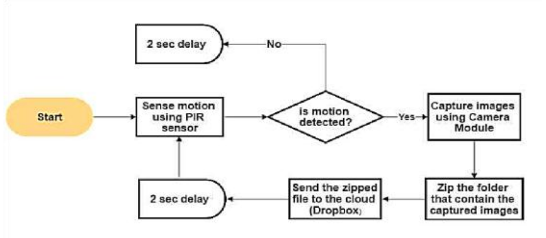


Fig.3.1: Raspberry Pi Software Setup Flowchart.

B. Software Configuration

Following the hardware setup with the Raspberry Pi, the software configuration at the local machine server is another crucial element. [1] While the Raspberry Pi handles motion detection and image capture, the local machine server is responsible for weapon detection and notifying the user. [2]

The software configuration is a critical component of the intelligent surveillance system, residing on a local machine server. It complements the Raspberry Pi's hardware setup by handling the tasks of weapon detection and user notification, which are not performed by the Raspberry Pi itself.

TensorFlow's Object Detection API is a key software tool employed in this configuration. This API leverages computer vision techniques to detect, locate, and trace objects within images or videos. TensorFlow provides a collection of pre-trained models for this purpose, known as the TensorFlow Object Detection Model Zoo.

The specific pre-trained model utilized in this system is SSD MobileNet v2. This model is chosen for its efficiency in object detection. SSD MobileNet is designed to detect objects with a speed of 22 milliseconds and a coco mAP (mean average precision) of 22.2. The SSD architecture, which underpins this model, uses a single convolution network to predict bounding box locations and classify them in a single pass.

The software configuration also involves a custom dataset, comprising 7000 images categorized into three classes: pistols, knives, and guns. This dataset is formatted using PASCAL VOC (Visual Object Classes). The software tools and languages used include Python and Anaconda Jupyter Notebook. Additionally, Python libraries such as Dropbox, Twilio, OpenCV, and Numpy are utilized.

The software configuration leverages TensorFlow and a pre-trained model (SSD MobileNet v2) for weapon detection, working in conjunction with other software tools and a custom dataset to provide the intelligent surveillance capabilities of the system

TABLE II.SOFTWARE FRAMEWORK AND TOOLS

No.	Framework/Tools /Language	API/Format/Model
1	TensorFlow	Object Detection API
2	Pre-trained Model	SSD MobileNet V2 fpnlite 320x320
3	Custom Dataset	PASCAL VOC (Visual Object Classes)
4	Python	Anaconda Jupyter Notebook
5	Dropbox, Twilio, OpenCV, Numpy	Python Libraries Package

C. Dataset Collection

the dataset used for training the weapon detection model is a custom dataset. It is designed to contain 7000 images, categorized into three distinct classes. These classes are chosen to represent the weapons most commonly encountered in armed robbery and home invasion scenarios.

TABLE III. CLASSES AND NUMBER OF IMAGES

Top	Weapon	Number of Images
1	Pistol	2700
2	Knife	2700
3	Gun	1600

The dataset is divided into two records, a "train" record, and a "test" record, the "train" record contains 80% of the total number of images which it is 5600 images, these images are collected from the three classes, 2160 images from each of the "Pistol" class and "Knife" class, also, 1280 images from the "Gun" class. The rest 20% goes to the "test" record which it is 1400 images, as same as the "train" record, these images are collected from the three classes, 540 images from each of the "Pistol" class and "Knife" class, also, 320 from the "Gun" class.

Each class has three types of images. Images that contain the object without any background, images that contain the object in a normal environment were taken from self-defense videos, and last images that contain the object in real-life incidents that were taken by CCTV cameras.

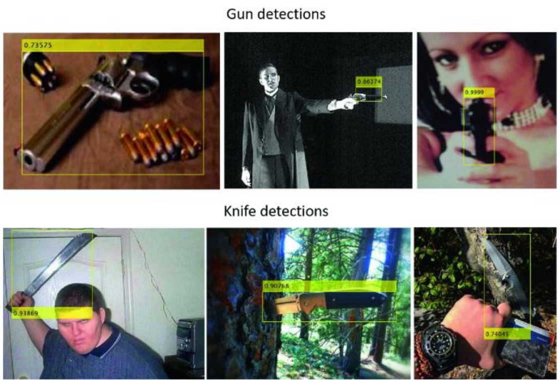


Fig.3.2: Sample From Pistol &knife Class Images.

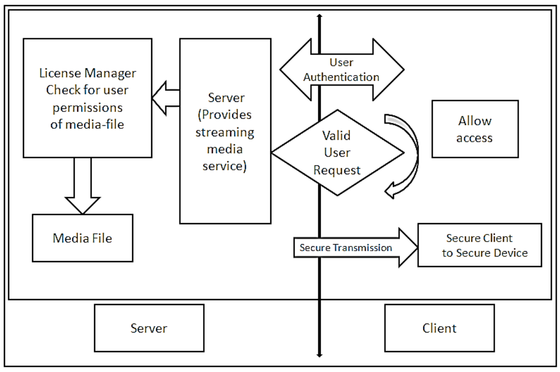


Fig. 3.3: Server Software Setup.

D. Server Software Setup

The server software setup involves a Python-based program designed to process images and detect weapons. This program is executed after the model has been trained for weapon detection. It functions by receiving images from Dropbox and applying object detection to identify potential threats. The program consists of several functions, organized into a seven-part structure. First, it checks the internet connection and prompts the user to verify and reboot the system if no connection is detected. If a connection exists, the program proceeds to check for new files in Dropbox and downloads them using the appropriate API. The downloaded file is then unzipped, and a path is specified for saving the extracted image file. To enhance the images captured by the Raspberry Pi camera, image processing techniques are applied, including OpenCV Image Equalizer, Gaussian Blur, and Image Rotation. Object detection is then performed on the images to identify any weapons using deep learning algorithms. In the event that a restricted object is detected, the user is notified through email, SMS, and optionally, voice alerts to a nearby security guard station. These alerts are color-coded (Red, Orange, and Yellow) to indicate the level of detection accuracy and urgency. Finally, the program deletes the downloaded files from Dropbox to prevent confusion and minimize memory consumption, while images containing important details are sent to the user via email. deleting the other files that were downloaded from Dropbox to eliminate any confusion between the uploaded and downloaded files, note that every file that was downloaded and the object detection was applied to will be removed to achieve one of the most important goals of this project which it is reduce memory consumption and any image that contain important details will be sent to the user by email if the object detection model detected any of the restricted object.

Algorithm

1. Raspberry Pi Motion Detection and Image Capture Algorithm

This algorithm outlines the process executed by the Raspberry Pi to detect motion using a PIR sensor and capture images using a camera module.

Input: PIR sensor data.

Output: Zipped folder of captured images.

Algorithm: Raspberry Pi Motion Detection and Image Capture

1. Start
2. Continuously monitor the PIR sensor for motion.
3. If motion is detected:

a. Activate the camera module to capture images.

b. Store the captured images in a designated folder.

c. Compress the folder containing the images into a zip file.

d. Upload the zipped file to a cloud storage service (Dropbox).

4. Repeat step 2.

2. Server-Side Weapon Detection Algorithm

This algorithm details the process performed by the local machine server to detect weapons in images received from the Raspberry Pi.

Input: Zipped folder of images from cloud storage.

Output: User notification (email, SMS, voice alert) upon weapon detection.

Algorithm: Server-Side Weapon Detection

1. Start
2. Continuously check for new zipped files in the cloud storage.
3. If a new file is found:

a. Download the zipped file.

b. Extract the images from the downloaded zip file.

c. For each extracted image:

i. Apply image processing techniques (e.g., OpenCV functions).

ii.Utilize a pre-trained deep learning model (SSD MobileNet V2) to detect weapons.

d. If a weapon is detected in any image:

i.Determine the threat level based on detection accuracy (e.g., Red, Orange, Yellow).
ii. Alert the user via email, SMS, and/or voice alert.
e. Delete the downloaded zip file and extracted images.
4. Repeat step 2.
3. Deep Learning Model Evaluation Algorithm
This algorithm describes the process of evaluating the performance of the weapon detection model.
Input: A test dataset of images with known labels.
Output: Evaluation metrics, including accuracy, precision, recall, and F1-score.
Algorithm: Deep Learning Model Evaluation
1. Input a test dataset of images with corresponding ground truth labels.
2. Use the trained weapon detection model to predict labels for the images in the test dataset.
3. Construct a Confusion Matrix to compare the predicted labels with the ground truth labels.
4. Calculate the following evaluation metrics based on the Confusion Matrix:
a. Accuracy
b. Precision (for each class)
c. Recall (for each class)
d. F1-Score (for each class)
5. Output the calculated evaluation metrics.

RESULTS AND DISCUSSION

After training the SSD MobileNet we need to evaluate the performance as an overall evaluation and evaluate each class as a classification evaluation, but to understand the evaluation and implementation results there are a set of matrices that must be known. TensorFlow provides an Evaluation Function that returns key object detection evaluation matrices, notably Mean Average Precision (mAP) and Mean Average Recall (mAR). Mean Average Precision is a crucial performance metric for assessing machine learning and deep learning models, though it relies on other metrics. One such fundamental metric is Intersection over Union (IoU), which quantifies the overlap between predicted and actual object bounding boxes, indicating the accuracy of object detection and localization.

- Intersection over Union (IoU) is a metric that evaluates how well the detection and prediction are, and it quantifies the degree of overlap between two regions.

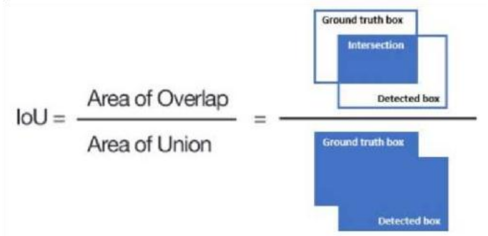


Fig. 4.1: Intersection Over Union Calculation.

- Confusion Matrix

The Confusion Matrix is a tool used to evaluate the performance of the object detection aspect of the system. It's conceptualized using an example where the model's task is to identify the presence of a "Pistol" within an image. The model's predictions about the presence or absence of the pistol can be either correct or incorrect. The Confusion Matrix helps categorize these predictions. In the context of object detection, the determination of whether a prediction is a True Positive (TP), False Positive (FP), or False Negative (FN) is theoretically aided by the Intersection over Union (IoU) threshold.

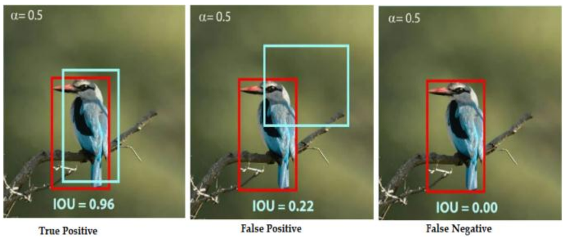


Fig.4.2: Object Detection Prediction Types and Errors.

Confusion Matrix = [[N + 1: x [N + 1]]
In object detection, the validity of prediction can be (TP, FP, or FN) is decided with the help of the IoU threshold.

Fig.4.3: Object Detection Prediction with Help of the IoU.

- Precision (P) calculates the performance of the deep learning model by calculating the proportion of predicted positives that are actually correct. Simply it is the "True

Positives" out of total detections. The value ranges from 0 to 1.

- Precision = TP/(TP + FP)

- Recall

Recall (R) is similar to Precision (P) and measures the proportion of actual positives that were predicted correctly. It is calculated as the "True Positives" out of all "Ground Truths". Ground truth means known objects. The value of Recall also ranges from 0 to 1.

Recall = TP/(TP + FN)

- Average Precision

It represents the area under the Precision-Recall (PR) curve, providing a scalar summary of the curve. A high AP indicates both high precision and high recall across various confidence threshold values, with AP values ranging from 0 to 1. Average Recall (AR), on the other hand, describes the area doubled under the Recall x IoU curve. The Recall x IoU curve plots recall results for each IoU threshold between 0.5 and 1.0. AR values also range from 0 to 1

- Average Recall

According to the document, Average Recall (AR) describes the area doubled under the Recall x IoU curve. The Recall x IoU curve plots recall results for each IoU threshold where IoU is between 0.5 and 1.0, with IoU thresholds on the x-axis and recall on the y-axis. The range for AR is between 0 to 1

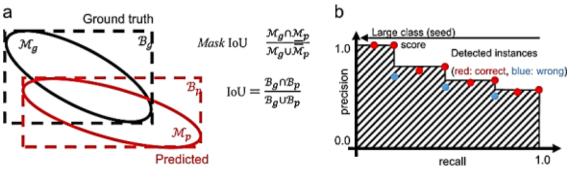


Fig. 4.4: Overall Object Detection Model Evaluation.

Classification Evaluation

The classification evaluation uses a confusion matrix to assess each class, comparing actual target values with deep learning model predictions. For TensorFlow 2, Confusion Matrix = [[N + 1: x [N + 1:: (7). Accuracy, the fraction of correct predictions, is calculated. The F1 score, the harmonic mean of precision and recall, is also used. Hardware and software configurations work simultaneously. Image Processing (IP) techniques like Histogram Equalizer, Gaussian Blur, and Image Rotation are applied. The system outputs warnings via email, SMS, and sound. The system aims to reduce costs compared to traditional surveillance systems.

System Outputs

Since the implemented system targets the top three most used weapons and the classes represent these weapons, the Confusion Matrix will be 4 x 4.

Fig.4.5.1: shows a 4 x 4 Confusion Matrix form.

		PREDICTED classification			
	Classes	a	b	c	d
ACTUAL classification	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

Fig. 4.5.2:4 x 4 Confusion Matrix Form.

```
Saving confusion matrix...

Confusion Matrix:
[[449.   0.   3. 106.]
 [  0. 321.   7.  57.]
 [  2.   7. 532.  49.]
 [  5.   7.  20.   0.]]

  category  precision_@0.5IOU  recall_@0.5IOU
0    Knife           0.984649        0.804659
1      Gun           0.958209        0.833766
2   Pistol           0.946619        0.901695
Done!
```

Fig. 4..5.3: Confusion Matrix Results.

The classification evaluation is shown in Fig.9 which it is represented in the form of a Confusion Matrix, Precision, and Recall for each class but we can also calculate the total accuracy with the help of the Confusion Matrix. Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.

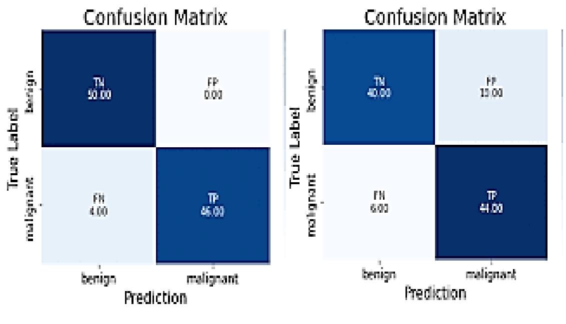
Accuracy = (TP + TN)/(TP + FP + FN + TN) =83.19%

To finish the classification evaluation, we need to calculate the F1 score, The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall. We use harmonic mean because it is not sensitive to extremely large values, unlike simple averages.

TABLE IV. F1 SCORE RESULTS

No.	Class	F1 Score
0	Knife	0.8856
1	Gun	0.8916

2	Pistol	0.9236



(a) SSD MobileNetV2 object detection model (b) ResNet50 classification model.

Fig. 4.5.4: SSD Mobile Net V2 Output.

The hardware configuration works simultaneously with the software configuration, the hardware captures images and sends them to the cloud the server downloads them and applies IP and object detection on the images to inform the user of any restricted object in the scene of shown in the fig

Fig.4.5.5: Hardware Config OutputVs.Software Config Output

Applying IP techniques guarantees that the system will give performance similar to traditional surveillance systems. Since the Raspberry Pi camera can give images out of contrast when it is indoors, a Histogram Equalizer needed to be applied to add more contrast, and in order to remove the "salt & paper" noise resulted by applying the Histogram Equalizer, Gaussian Blur needed to be applied too, and since the Raspberry Pi Camera can be upturned an image rotation need to be applied after Histogram Equalizer and Gaussian Blur.

Before Histogram equalization After Histogram equalization

Fig.4.5.6: before and after Histogram equalization

The system output is 3 types of warnings to inform the user, email alert, SMS alert, and sound alert.

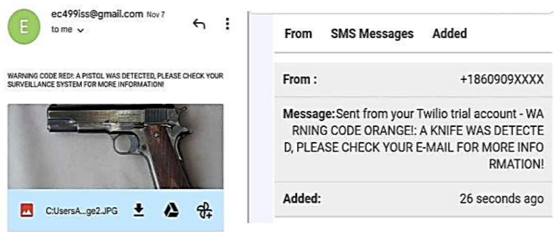


Fig. 4.7: Code Red Alert in Email and SMS.

One of the goals of this research is to reduce the cost of the traditional surveillance system and by studying the price improvement we achieved this goal. Raspberry Pi 3 Model B can cost 35$, also the Raspberry Pi Camera can cost 14.95$, and the PIR sensor can cost 35.99$. So it is around 85.95$ in total while the CCTV cameras have multiple brands but based on the specifications that are close to the raspberry pi camera that can cost around 173.90$.

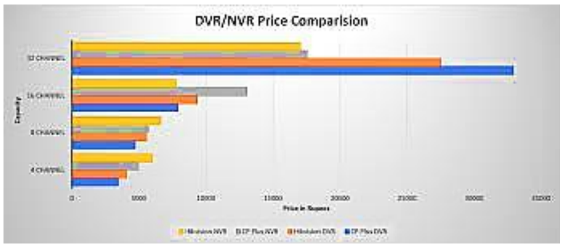


Fig:4.5.7:CCTV Prices in Compared to ISS Prices in Dollars.

CONCLUSION

In an era where security threats are becoming increasingly sophisticated, the integration of deep learning into surveillance systems has emerged as a game-changing solution. This paper explored the development and implementation of a smart surveillance system leveraging deep learning techniques to enhance security and monitoring capabilities. By utilizing state-of-the-art neural networks, such systems can perform real-time detection, classification, and tracking of objects, ensuring higher efficiency and accuracy compared to traditional surveillance methods.

The proposed system significantly improves the automation of surveillance by reducing the reliance on human monitoring, which is prone to fatigue and errors. Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), enable precise object recognition and anomaly detection. This helps in identifying potential threats, unauthorized access, or suspicious behavior in real-time, allowing security personnel to respond proactively. Moreover, the incorporation of edge computing and cloud-based analytics further enhances the efficiency of surveillance, making the system adaptable for various environments, including public spaces, corporate settings, and critical infrastructure.

Despite its numerous advantages, the deployment of deep learning-based surveillance systems comes with its own set of challenges. Computational power requirements, privacy concerns, and data security remain key issues that need to be addressed. Ethical considerations regarding the misuse of surveillance data and potential biases in deep learning algorithms must also be carefully managed to ensure responsible deployment. Future advancements in artificial intelligence, along with improvements in hardware capabilities, are expected to mitigate some of these challenges and enhance the robustness of such systems.

In conclusion, smart surveillance systems using deep learning represent a significant advancement in security and monitoring technology. With continuous improvements in AI models and computational infrastructure, these systems are poised to become even more effective in ensuring public safety. However, a balanced approach is necessary to address ethical concerns and privacy issues while leveraging the full potential of deep learning in surveillance applications.

REFERENCE

[1] B. Jadhav, "MOTION ACTIVATED SECURITY SYSTEM," *International Research Journal of Engineering and Technology (IRJET).*

[2] M. J. Rahman, "Motion Detection for Video Surveillance," Depertment of Computer Science, Högskolan Dalarna, 2008.

[3] R.Chandana, "Smart Surveillance System using Thing Speak," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 04, no. 07, p. 1, 2015.

[4] "Raspberry Pi 3 Model B Specification," Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-3model-b/.

[5] "Raspberry Pi Camera Specifications," Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/documentation/accessories/camera.ht ml.

[6] "PIR Sensor Datasheet & Pinout Specifications," 2 OCT 2021. [Online]. Available: https://www.homemade-circuits.com/pirsensor-datasheet-pinout-specification-working/.

[7] TensorFlow, "TF Certification," 23 Sep 2021. [Online]. Available: https://www.tfcertification.com/blog/how-tensorflow-objectdetection-works-where-to-useit#:~:text=TensorFlow%20object%20detection%20is%20a,or%20vi deo%20by%20detecting%20objects..

[8] V. Rathod, "GetHub Tensorflow Object detection Model Zoo," [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_d etection/g3doc/tf2_detection_zoo.md.

[9] C. C. Jenny Mouzos, Weapon Involvement in Armed Robbery, Australian Institute of Criminology-Research and Public Policy Series, 2001.

[10] L. Kukil, "Mean Average Precision (mAP) in Object Detection," 9 AUG 2022. [Online]. Available: https://learnopencv.com/meanaverage-precision-map-object-detection-model-evaluation-metric/.

[11] Baeldung, "Intersection over Union for Object Detection," 19 Sep 2022. [Online]. Available: https://www.baeldung.com/cs/objectdetection-intersection-vs-union.

[12] A. Anwar, "Toward Data Sicience," 13 May 2022. [Online]. Available: https://towardsdatascience.com/what-is-averageprecision-in-object-detection-localization-algorithms-and-how-tocalculate-it-3f330efe697b.

[13] M. E. Aidouni, "Evaluating Object Detection Models: Guide to Performance Metrics," 5 OCT 2019. [Online]. Available: https://manalelaidouni.github.io/Evaluating-

Object-DetectionModels-Guide-to-Performance-Metrics.html.

[14] A. Bhandari, ""Everything you Should Know about Confusion Matrix for Machine Learning," 17 Apr 2020.

[15] "Machine Learning," Gogle Developers, [Online].Available: https://developers.google.com/machine-learning/crashcourse/classification/accuracy.

[16] "Raspbery Pi Price," [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-3-model-bplus/.

[17] "Raspberry Pi Camera Price," [Online]. Available: https://www.raspberrypi.com/products/camera-module-v2/.

[18] "Amazon-PIR Sensor Price," [Online]. Available: https://www.amazon.com/Adafruit-LK-918O-SANVFBACA-PIR-Motion-Sensor/dp/B00JOZTAC6/ref=sr_1_17?crid=37GAN5EHVFGTP&keywords=pir+motion+sensor&qid=1.

[19] "Amazon CCTV 8MP Price," [Online]. Available: shorturl.at/avFSZ.