Target Identification, Tracking & Engagement System

By

Capt Abdul Mutalib

Capt Junaid Munir Shad

Capt Faraz Hanif

Capt Ausaf Anis

Supervised by:

Col Saif Ullah Khalid, PhD

Submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology, Islamabad,
in partial fulfillment for the requirements of B.E Degree in Electrical (Telecom) Engineering.

May 2023

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"Target Identification, Tracking & Engagement System"

is carried out by

Capt Abdul Mutalib

Capt Junaid Munir Shad

Capt Faraz Hanif

Capt Ausaf Anis

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical (Telecom.) Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor
Col Saif Ullah Khalid, PhD
R&D Wing, MCS

Date: ___________

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues and most of all supervisor ,**Col Saif Ullah Khalid, PhD** without your guidance. The group members, who through all adversities worked stead fastly.

Plagiarism Certificate (Turnitin Report)

This thesis has ____ similarity index. Turnitin report endorsed by Supervisor is attached.

____

Capt Abdul Mutalib

00000325148

____

Capt Junaid Munir Shad

00000325157

____

Capt Faraz Hanif

00000325156

____

Capt Ausaf Anis

00000325164

____

Signature of Supervisor

ABSTRACT

This project is designed to detect and track a target and finally engage it using a control system autonomously. The system is implemented on a hardware turret and uses a combination of image processing and machine learning techniques to identify and track potential threats in real-time. The deep learning algorithm is trained on a dataset of various enemy uniforms and weapons, allowing it to recognize potential threats accurately. The system has two main modules, a recognition module, and a tracking and engaging module, which work together to detect and track potential threats autonomously. Data set is scaled down to four types, first is Pakistan Army Uniform soldier and Indian Army (enemy) uniform and other classes are weapons, animals and civilians. Model is developed and is trained to detect Pakistani Soldier, Indian Soldier, a civilian and a Gun threat involving our data set. Model has the capability of engaging only Indian uniform soldiers and recognizing Pakistan Army soldiers and civilians to protect them. We propose a model that provides a visionary sense to a machine or robot to identify the unsafe target and can also engage when a target is obvious in the edge.

Table of Contents

List of Figures

Chapter 1: Introduction

Target detection and engagement systems have become crucial components of modern military operations. These systems are designed to detect and engage targets, both stationary and moving, with precision and accuracy. They are essential in maintaining situational awareness, and providing critical information to the decision-makers for timely response. Despite the advances in technology, target detection and engagement systems continue to face several challenges and limitations including data processing and analysis, and system integration. These challenges require a multidisciplinary approach, combining expertise from various fields such as engineering, computer science, and military operations.

This project focuses on the problem of detecting threat using computer vision techniques. The proposed solution utilizes deep learning algorithms to train a model to identify threat in real-time using a camera. This solution has important implications for national security, as it can be used to detect Indian soldiers in disputed border regions.

The value of this project is significant in terms of national security. With the increasing tensions between India and its neighboring countries, particularly in the disputed border regions, the ability to detect Indian soldiers in real-time using computer vision techniques can be crucial for maintaining peace and preventing any untoward incidents. The proposed solution utilizing deep learning algorithms has the potential to be a game-changer in the field of border security, as it can provide real-time monitoring and detection of Indian soldiers in critical areas. Moreover, the project can also pave the way for further research in the field of computer vision and deep learning, particularly in the context of national security.

1.1 Overview

Target detection and engagement systems are essential components of modern military operations. These systems are designed to detect, track and engage targets. With the advancement of technology, these systems have become increasingly complex, incorporating a range of sensors, data processing algorithms, and communication systems. The project aims to develop a cost-effective and easily deployable system that can be used by security personnel to monitor and secure disputed border regions. The system is designed to be user-friendly and efficient, allowing security personnel to quickly and accurately identify Indian soldiers in the area.

To achieve this, a deep learning algorithm, known as convolutional neural networks (CNNs), has been utilized to analyze and classify images of Indian soldiers. The project will also involve the collection and annotation of datasets to train and test the model.

Overall, this project aims to provide a practical solution to a critical national security issue using computer vision and deep learning techniques. The proposed solution has important implications for border security and has the potential to be adapted and deployed in other similar scenarios.

1.2 Problem Statement

The issue of national security at disputed border regions requires constant monitoring and surveillance to ensure the safety and protection of citizens and the nation. However, this task is often challenging and risky for soldiers who are required to be on duty for extended periods, leading to fatigue and reduced vigilance, which can compromise the effectiveness of their work. The current target detection and engagement systems used by the other military powers suffer from several limitations. Such systems are often expensive and require regular maintenance, which can be a significant financial burden for military organizations.

1.3 Proposed Solution

The proposed solution for this project is to develop a real-time system that can accurately detect Indian soldiers in different situations and environments using a Single Shot MultiBox Detector (SSD) based model. The system uses computer vision and deep learning techniques to analyze and classify images of Indian soldiers. To achieve this, the system will be trained on a dataset of annotated images of Indian soldiers to improve the accuracy of the model. Overall data set consists of six classes including Indian soldiers, Pakistani soldiers, civilians, animals and weapons. The proposed solution is user-friendly and efficient, allowing security personnel to quickly and accurately identify Indian soldiers. The system is designed to operate in real-time and can process images from a camera or other video stream, making it well-suited for border surveillance applications. The SSD-based model allows for the detection of multiple objects in a single image, making it well-suited for identifying Indian soldiers in complex situations. The proposed solution is cost-effective and can be easily deployed in various border regions with minimal hardware requirements. Overall, the solution uses advanced computer vision and deep learning techniques to provide a practical solution to a critical national security issue, with important implications for border security and potential for deployment in other similar scenarios.

1.4 Working Principle

The project mainly works on the principles of image processing amalgamated with machine learning algorithms. The project is divided into different modules and every module is inter-woven with the next module. The list of modules is as under:

- Dataset creation, processing and annotations
- Model creation, training and validation

- Decision based upon Output
- Integration

1.4.1 Dataset creation, processing and annotations:

The first step is to collect images of Indian soldiers, Pakistani soldiers, weapons, civilians and animals in various situations and environments. The Indian and Pakistan soldiers images are obtained from publicly available databases, online sources, or captured from videos available on internet.

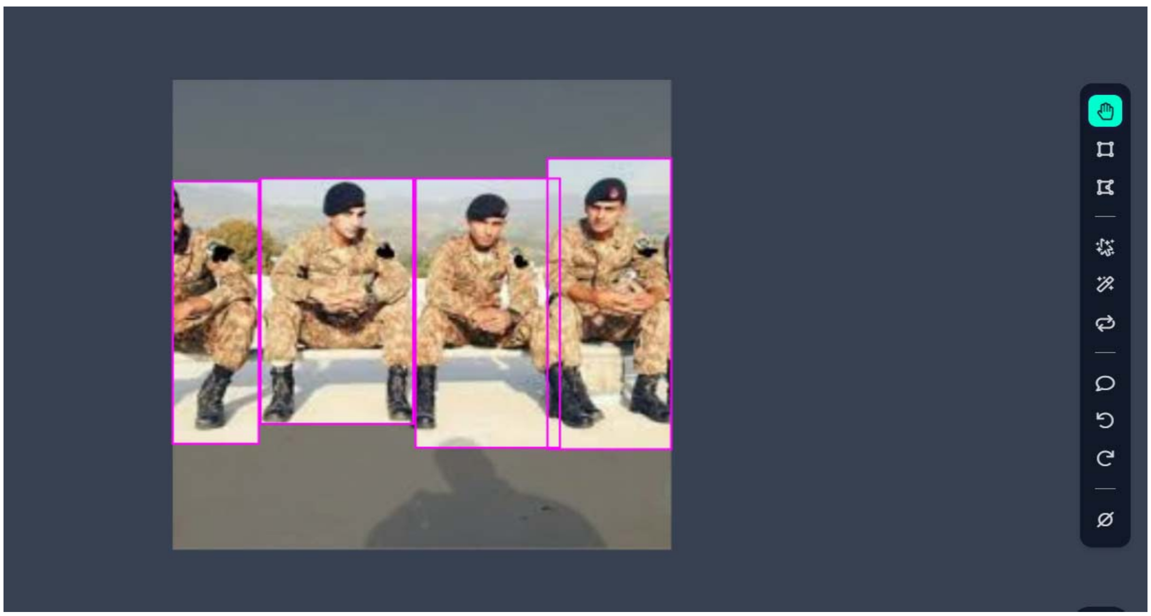


Fig 1

1.4.1.1 Indian Soldiers Dataset:

This dataset is obtained from online sources comprising of vast sets of Indian soldiers in different terrains/ environment. This data set is 7500 in total comprising 3500 stills and rest 4000 captured from videos available on online sources.

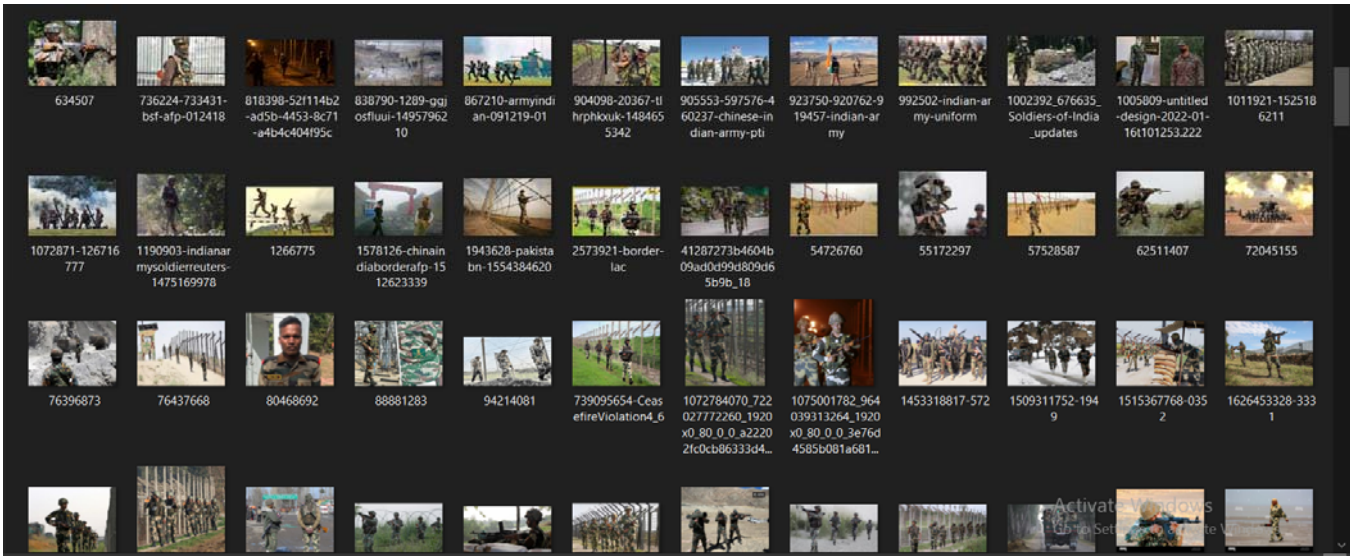


Fig 2

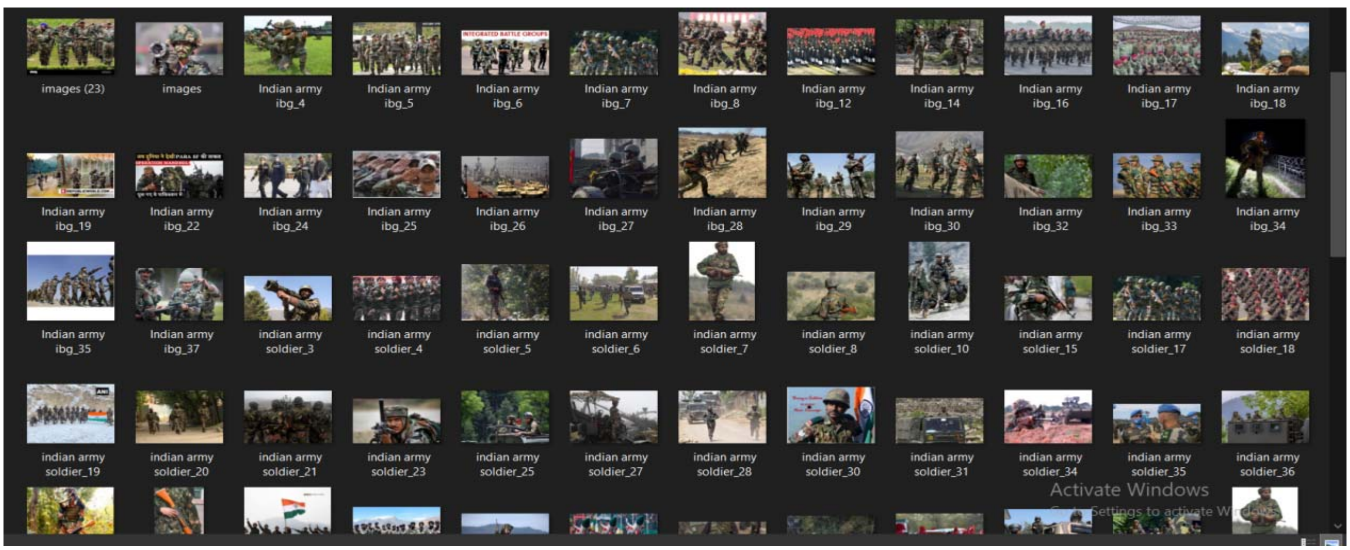


Fig 2-a

1.4.1.2 Pakistani Soldiers Dataset:

This dataset is obtained from online sources comprising of vast sets of own soldiers in different terrains/ environment. This dataset has been labeled dataset comprising of friendly forces in various terrains to differentiate between friends and foe. This data set is 3500 in total.

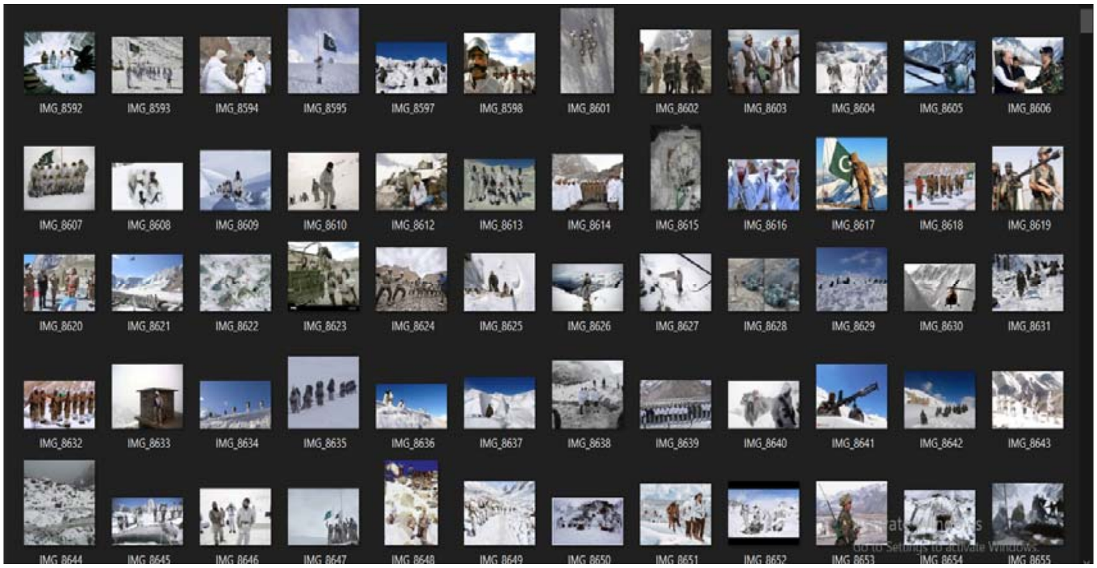


Fig 3

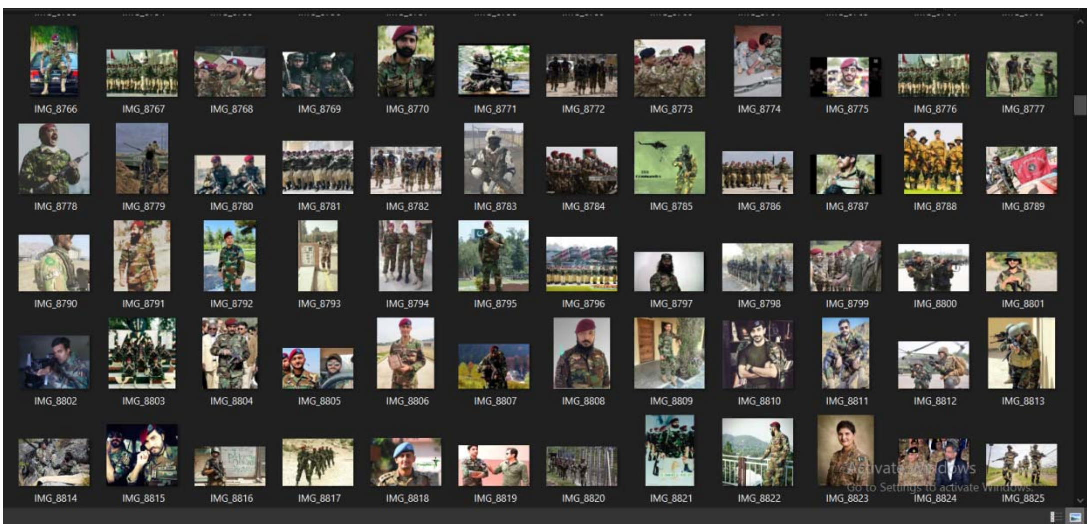


Fig 4

1.4.1.3 Other Dataset:

Other dataset classes includes animals, civilians, and weapons of different categories.

1.4.2 Dataset training and processing:

The prepared dataset is used as input to train object detection models using machine learning. The training process is divided into two functions train and validate. The train function is responsible for training the model using the training data loader. It iterates through each batch of training data and performs the following operations:

- Sends the images and targets to the specified device (GPU or CPU)
- Computes the loss using the model and loss function
- Computes the gradients of the loss with respect to the model parameters using back propagation
- Updates the model parameters using the optimizer
- Logs the training loss for each iteration
- Returns the list of training losses

The validate function is responsible for evaluating the model using the validation data loader. It iterates through each batch of validation data and performs the following perations:

- Sends the images and targets to the specified device (GPU or CPU)
- Computes the loss using the model and loss function
- Logs the validation loss for each iteration
- Returns the list of validation losses

1.4.3 Decision based upon Outputs:

The program detects the presence of Indian soldiers in the video frames captured by the webcam. If an Indian soldier is detected, the program calculates the centroid of the bounding box around the soldier and sends the relative polar coordinates of the centroid to an Arduino board via serial communication. The Arduino board is programmed to activate the gun when the Indian soldier is detected. The program also displays the video frames with the bounding boxes around the detected objects (if any) in a window named "Webcam Detection".

Fig 5 & 5-a

Fig 5-b

1.4.4 Integration:

The system is integrated with an Arduino microcontroller board for alerting the user about the detection of Indian soldiers. When the Indian soldier class is detected, the system sends the polar coordinates of the centroid of the bounding box to the Arduino. The Arduino then activates a the gun to fire at Indian soldiers.

The arduino triggers the gun via servo motor connected to the board. The system sends the polar coordinates in the form of a string in the following format: "{distance},{angle}". The string is encoded and sent to the Arduino using the pyserial library. The Arduino then reads the incoming data, parses it, and uses the parsed values to activate the gun.

The integration of the Arduino with the object detection system provides a real-time alert mechanism that can be used in scenarios where it is crucial to detect and identify the presence of Indian soldiers.

1.5 Objectives

1.5.1 General Objectives:

"To build an innovative state of the art software integrated hardware prototype powered by Machine Learning (ML) and to develop a real-time object detection and tracking system, integrated with an Arduino-based hardware setup, to detect the presence of Indian soldiers in a given area and trigger an alert through a toy gun, with the aim of enhancing security and safety in border areas providing a smart weapon system for Pakistan Army."

1.5.2 Academic Objectives:

- To gain practical experience in developing and implementing computer vision algorithms using deep learning techniques.

- To explore the potential applications of computer vision and machine learning in the field of security and surveillance.
- To investigate the effectiveness of object detection models in identifying specific objects of interest in real-time video streams.
- To gain experience in integrating hardware components (such as cameras, stepper motors, and microcontrollers) with software systems.
- To evaluate the performance of the proposed system in terms of accuracy, efficiency, and scalability.
- To contribute to the academic literature on computer vision and machine learning by presenting the results of this project in research papers and conference proceedings.
- To develop skills in project management, collaboration, and communication through working on a multidisciplinary team to complete the project.

1.6 Scope

The scope of this model is to detect and track Indian soldiers in real-time using a trained deep learning model along with differentiating between classes defined. The model can be integrated with a camera or a video stream and can be used for security purposes. The model can be further extended to detect other objects and classes, depending on the requirements of the project.

1.7 Deliverables

1.7.1 Object of interest:

The object of interest in our project refers to the Indian soldiers and Pakistani soldiers in the context of border security. The model uses a combination of image processing and machine learning techniques to detect the presence of indian soldiers and track their movements. The goal is to provide real-time monitoring and alerting the security personnel in order to enhance situational awareness and prevent potential security threats. Additionally, the model can also detect other objects of interest such as pistols, civilians, and animals.

1.8 Relevant Sustainable Development Goals

There are several Sustainable Development Goals (SDGs) that are relevant to target detection and identification:

1. SDG 9: Industry, Innovation, and Infrastructure - Target detection and identification technologies are essential for the development of smart cities and intelligent transportation systems, which can improve efficiency, safety, and sustainability.
2. SDG 11: Sustainable Cities and Communities - Target detection and identification can help improve public safety and reduce crime, making cities more livable and sustainable.
3. SDG 16: Peace, Justice, and Strong Institutions - Target detection and identification can be used to prevent and respond to crime and terrorism, promoting peace and security.
4. SDG 17: Partnerships for the Goals - The development and implementation of target detection and identification technologies require collaboration between governments, private sector, and civil society, promoting sustainable and inclusive partnerships.

Overall, the use of target detection and identification technologies can contribute to several SDGs by improving safety, efficiency, and sustainability in different contexts. However, it is important to ensure that these technologies are developed and used responsibly and ethically, with consideration for potential social and environmental impacts.

1.9 Structure of Thesis

Chapter 2 contains the literature review and the background and analysis study this thesis is based upon.

Chapter 3 contains the design and development of the project.

Chapter 4 introduces detailed evaluation and analysis of the code.

Chapter 5 contains the conclusion of the project.

Chapter 6 highlights the future work needed to be done for the commercialization of this project.

Chapter 2: Literature Review

A new product is launched by modifying and enhancing the features of previously launched similar products. Literature review is an important step for development of an idea to a new product. Likewise, for the development of a product, and for its replacement, related to traffic system, a detailed study regarding all similar projects is compulsory. Our research is divided into the following points.

- Industrial Background
- Existing solutions and their drawbacks
- Research Papers

2.1 Industrial background

The application of object detection models in industrial settings has a wide range of potential uses. One such use could be in security and surveillance systems, where the model can be used to detect and identify potential security threats in real-time. This can be particularly useful in high-security areas such as airports, seaports, or government buildings.

Another potential application of object detection models in industrial settings is in quality control and inspection. The model can be trained to detect and identify defects or anomalies in products, parts, or components during the manufacturing process. This can help companies identify and correct issues before the products reach the market, improving product quality and reducing waste. Furthermore, object detection models can be used in autonomous robots for various industrial tasks such as warehouse management, stocktaking, and delivery. By detecting and identifying objects in their environment, robots can navigate through the space and perform tasks more efficiently and accurately.

The potential applications of object detection models in industrial settings are diverse and can bring many benefits, including improved security, quality control, and automation.

2.2 Existing solutions and their drawbacks

To achieve detection and tracking, there are many techniques/algorithms which have their own pros and cons. Some of the important algorithms are:

- Frame Differencing algorithm [19],[20]
- Background Subtraction algorithm[21]
- Optical flow value algorithm[22]
- Mean-Shift algorithm[23]

2.2.1 Frame differencing algorithm

Frame differencing is a common technique used in computer vision and image processing to detect motion in video sequences. However, like any algorithm, it has some drawbacks. Here are some of the limitations of frame differencing:

- Sensitivity to noise: Frame differencing is sensitive to noise, which can lead to false detections of motion.
- Limited detection accuracy: Frame differencing can only detect motion that occurs between consecutive frames.
- Inability to track object movement: Frame differencing is not designed to track the movement of an object in a scene.

2.2.2 Background Subtraction algorithm

Background subtraction is a common technique used in computer vision to detect moving objects in a video stream. It works by comparing each frame of the video to a background image to identify pixels that have changed. While background subtraction can be effective, there are several drawbacks to this approach:

- Lighting changes: Background subtraction assumes that the lighting conditions in the scene are consistent.
- Static background: Background subtraction assumes that the background is static or slowly changing over time.
- Noise: Noise in the video stream can interfere with the background subtraction algorithm, causing false positives or false negatives.

2.2.3 Optical flow value algorithm

Optical flow is a technique used in computer vision to track the movement of objects in a video stream. It works by analyzing the movement of pixels between consecutive frames to estimate the velocity of objects in the scene. While optical flow can be effective, there are several drawbacks to this approach:

- Large motions: Optical flow assumes that the motion between frames is small. If there are large motions, such as fast-moving objects or sudden changes in camera position, it can cause errors in the optical flow estimation.
- Low-texture regions: Optical flow relies on texture information to track the movement of objects. If there are regions in the image with low texture, such as smooth surfaces or areas of uniform color, it can be difficult to estimate the optical flow.
- Computational complexity: Optical flow algorithms can be computationally intensive, particularly when analyzing high-resolution video streams. This can limit the real-

time performance of optical flow-based systems or require specialized hardware to run efficiently.

2.2.4 Mean Shift algorithm

Mean shift is a popular non-parametric clustering algorithm used in computer vision and machine learning. The algorithm works by iteratively shifting the centroid of a cluster towards the mode of the data distribution until convergence. While mean shift can be effective, there are several drawbacks to this approach:

- Parameter tuning: Mean shift has several parameters that need to be tuned for optimal performance, including the bandwidth parameter that controls the size of the search window. Finding the right parameters can be challenging and time-consuming, and may require expertise in computer vision and machine learning.
- Computational complexity: Mean shift can be computationally expensive, particularly for large datasets or high-dimensional feature spaces. This can limit the scalability of mean shift-based algorithms or require specialized hardware to run efficiently.
- Sensitivity to initialization: Mean shift can be sensitive to the initial centroid or seed point used to start the iterative process. This can lead to different clustering results depending on the initialization, and may require multiple runs with different initializations to ensure robustness.

Chapter 3

Design and development of the project

3.1 Project Hardware

3.1.1 Tripod module:

A three-legged support structure tripod stand that is used in videography to stabilize cameras and other equipment. In the context of the project, a tripod stand is used to hold the camera and the turret assembly.

The tripod stand provides a stable platform for the camera and the turret, which is essential for accurate tracking and detection. The stand also allows for easy adjustment of the camera's position and angle, which is necessary for covering a wide area.

The tripod stand used in the project is made of lightweight and durable materials, such as aluminum or carbon fiber. It is designed to be portable and easy to set up, making it suitable for outdoor use. The legs of the stand are adjustable in height, which allows the camera to be positioned at various heights.

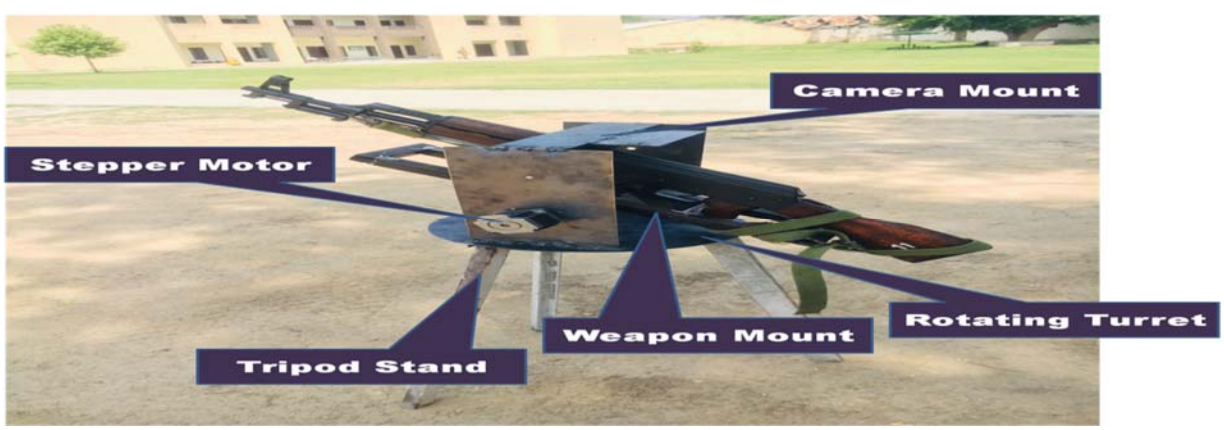


Fig 6

3.1.2 Camera module:

Fig 7

The camera module used in this project is mounted on a turret, which is capable of rotating horizontally and vertically to capture a wider field of view. The turret is controlled by stepper motors with drivers, which allows for precise and accurate movement of the camera.

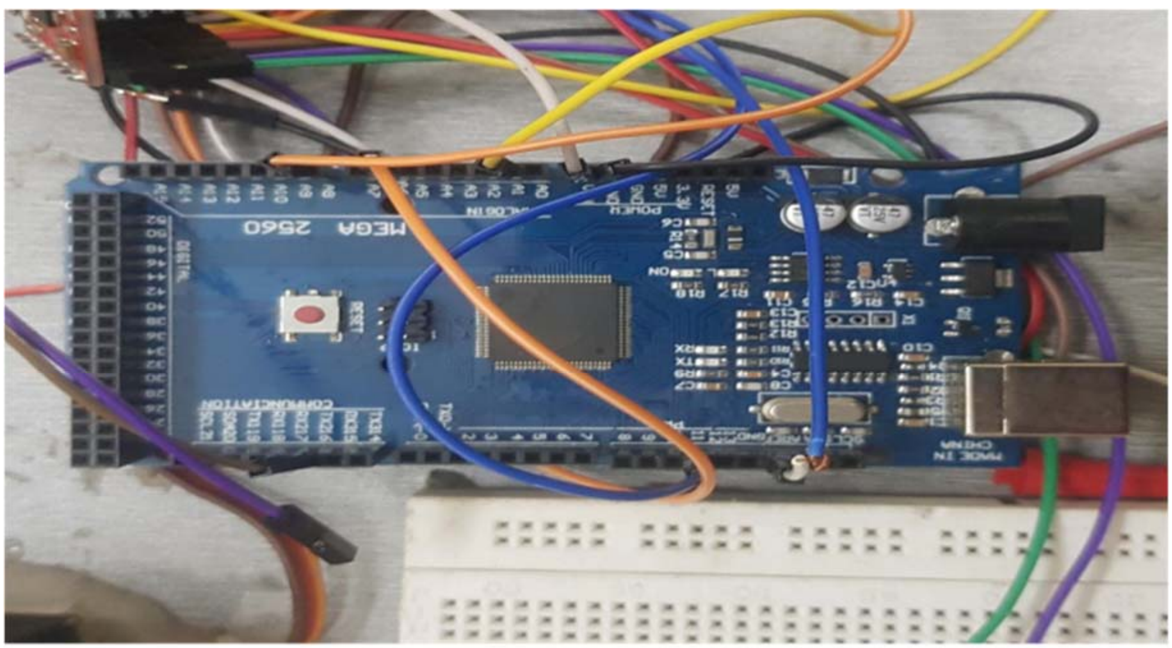
Stepper motors are commonly used in robotics and automation due to their ability to accurately control position and speed. They work by converting digital pulses into rotational motion, with each pulse moving the motor by a fixed amount. This allows for precise control of the motor's position and speed, making it an ideal choice for controlling the camera turret.

To control the stepper motors, dedicated drivers were used. These drivers provide the necessary power and signals to drive the motor, as well as protection from over-current and over-temperature conditions. The drivers were connected to the microcontroller, which allowed for easy control of the motors through software.

By using stepper motors and drivers, the camera turret was able to move smoothly and accurately, allowing for precise control over the camera's field of view. This was essential for the success of the project, as it allowed for accurate detection and tracking of the target objects.

3.1.3 Micro controller

The system is integrated with an Arduino microcontroller board for alerting the user about the detection of Indian soldiers. When the Indian soldier class is detected, the system sends the polar coordinates of the centroid of the bounding box to the Arduino. The Arduino then triggers a to alert the user about the presence of Indian soldiers.



Microcontroller Arduino Mega 2560

Fig 8

3.1.4 Stepper Motor

This Bipolar NEMA23 2A Stepper Motor is very useful in the various application, especially which demands low speed with high precision.

Fig 9

3.1.5 TB6560 Driver

The TB6560 stepper motor driver module is used to drive 2-phase and 4-phase stepper motors. It used a Bi-polar H-bridge configuration with MOSFETS. This type of driver module uses a step pin or direction pin to determine the direction of the stepper motor.

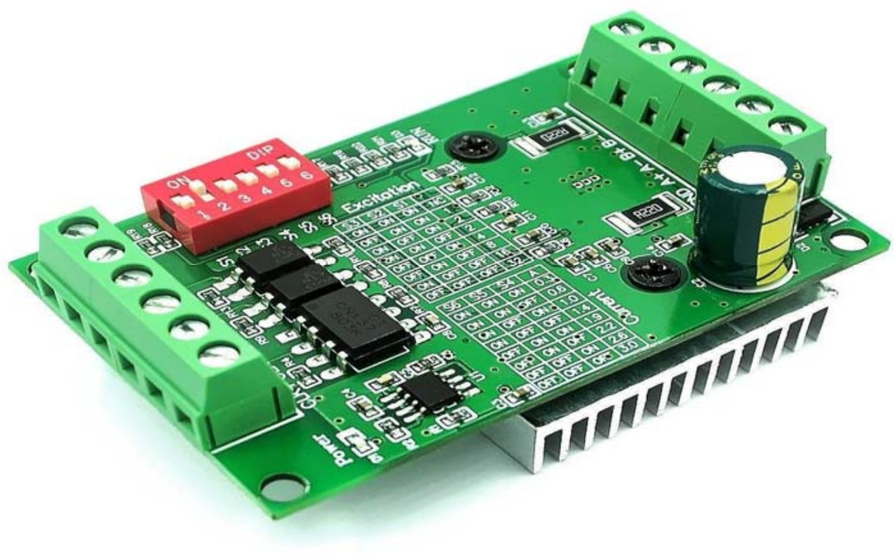


Fig 10

3.1.6 Power Supply

DC 12V 10A switching power supply. It is ideal for CCTV security system or for DC water pump. Specification Input: 110V/220V Output: 12V DC regulated.

Fig 11

3.1.7 HP Spectra 360

HP Spectre X360 is a Windows 10 Professional laptop with a 13.30-inch display that has a resolution of 1920x1080 pixels. We are using this laptop for data processing and analysis.

Fig 12

Chapter 4
Evaluation & Analysis of Code

4.1 Main Code

Object detection on a live video stream from a webcam and sends the location of the detected object to an Arduino board connected via serial communication. Here is a breakdown of the script:

- Imports the necessary libraries: numpy, cv2, torch, time, and serial.
- Defines the list of class names (CLASSES) and the total number of classes (NUM_CLASSES).
- Creates the object detection model by calling the create_model function from the models module and loading the trained weights from a file using torch.load.
- Sets the detection threshold to 0.8.
- Initializes the webcam using OpenCV's VideoCapture function.
- Connects to the Arduino board via pyserial.
- Enters an infinite loop that captures frames from the webcam and performs object detection on each frame.
- Converts the frame to RGB, scales it to the range of [0, 1], and transposes the dimensions from (height, width, channels) to (channels, height, width) to match the input format of the model.
- Calls the model's forward pass on the preprocessed frame and retrieves the predicted bounding boxes and labels for each object detected in the frame.
- Extracts the bounding boxes for Indian soldiers and calculates the centroid of the first detected Indian soldier.

- Calculates the relative position of the centroid from the center of the frame and converts it to polar coordinates.
- Sends the polar coordinates of the Indian soldier to the Arduino board via serial communication.
- Displays the original frame with the bounding boxes and labels drawn on it.

Waits for the user to press the 'q' key to exit the program. Releases the webcam and destroys all OpenCV windows.

```python
import numpy as np
import cv2
import torch
import time

from models import create_model

device = torch.device("cpu")


CLASSES = [
    'background', 'indian soldier', 'pakistani soldier', 'pistol', 'civilian', "animal"

]
NUM_CLASSES = 6

model = create_model(num_classes=6).to(device)
model.load_state_dict(
    torch.load("/Users/Muhammad Haseeb/Desktop/sold detect/model100.pth", map_location=device)
)

model.eval()

detection_threshold = 0.8

cap = cv2.VideoCapture("/Users/Muhammad Haseeb/Desktop/sold detect/indian army 2.mp4")
```

Det.py 3 ×

C: > Users > Muhammad Haseeb > Desktop > sold detect > Det.py > ...

```python
while True:
    ret, frame = cap.read()
    if not ret:
        break

    orig_frame = frame.copy()
    frame = cv2.cvtColor(orig_frame, cv2.COLOR_BGR2RGB).astype(np.float32)
    frame /= 255.0
    frame = np.transpose(frame, (2, 0, 1)).astype(float)
    frame = torch.tensor(frame, dtype=torch.float).cpu()
    frame = torch.unsqueeze(frame, 0)

    with torch.no_grad():
        outputs = model(frame)

    if len(outputs[0]['boxes']) != 0:
        boxes = outputs[0]['boxes'].data.numpy()
        scores = outputs[0]['scores'].data.numpy()

        boxes = boxes[scores >= detection_threshold].astype(np.int32)
        draw_boxes = boxes.copy()

        pred_classes = [CLASSES[i] for i in outputs[0]['labels'].cpu().numpy()]

        for j, box in enumerate(draw_boxes):
            cv2.rectangle(orig_frame,
```

C: > Users > Muhammad Haseeb > Desktop > sold detect > Det.py > ...

```python
            cv2.rectangle(orig_frame,
                        (int(box[0]), int(box[1])),
                        (int(box[2]), int(box[3])),
                        (0, 0, 255), 2)
            cv2.putText(orig_frame, pred_classes[j],
                        (int(box[0]), int(box[1]-5)),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0),
                        2, lineType=cv2.LINE_AA)

    cv2.imshow('Webcam Detection', orig_frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

4.2 Microcontroller Code

Code explanation

- First, the code includes the necessary libraries for controlling the stepper motors and serial communication.
- Next, it defines some constants such as the number of steps per revolution for the stepper motors and the delay between steps.
- The code initializes the serial communication to communicate with the Python script.
- It sets up the two stepper motors by specifying the pins connected to the motor coils and the step mode.
- The code waits for a signal from the Python script to start tracking.
- Once the signal is received, the code reads the X and Y coordinates of the detected object from the serial buffer.
- The code calculates the number of steps required to move the stepper motors to the desired position based on the current position and the target position.
- The code then moves the stepper motors to the target position by stepping through the required number of steps.
- The code repeats steps 6-8 until it receives a signal to stop tracking from the Python script.

This Arduino code aims to track an object detected by an external system, such as a machine learning model, using two stepper motors controlled by the Arduino board. The stepper motors are connected to the pins defined as motor1_pin1, motor1_pin2, motor2_pin1, and motor2_pin2. The code starts by creating AccelStepper objects for the two motors and setting their maximum speed and acceleration. The step size of the motors is also defined as step size. In the setup() function, the serial communication is initialized, and the initial position of the motors is set to 0 using set Current Position().

In the loop() function, the code first reads a frame from the video stream and detects an object in the frame using an external system. If an object is detected, the code calculates its position and distance from the center of the frame. Based on this, the code calculates the number of steps needed to move the motors in the X and Y directions to track the object.

The motors are then moved using the move() function of the Accel Stepper library, which takes the number of steps as an argument. The run() function is used to update the motors continuously until the next iteration of the loop() function.

sketch_apr29a | Arduino 1.8.19

File Edit Sketch Tools Help

sketch_apr29a §

```
#include <AccelStepper.h>

// Define the stepper motor connections
#define motor1_pin1 10
#define motor1_pin2 11
#define motor2_pin1 12
#define motor2_pin2 13

// Create the stepper motor objects
AccelStepper motor1(AccelStepper::DRIVER, motor1_pin1, motor1_pin2);
AccelStepper motor2(AccelStepper::DRIVER, motor2_pin1, motor2_pin2);

// Set the maximum speed and acceleration for the motors
const float max_speed = 1000.0;
const float acceleration = 100.0;
motor1.setMaxSpeed(max_speed);
motor1.setAcceleration(acceleration);
motor2.setMaxSpeed(max_speed);
motor2.setAcceleration(acceleration);

// Define the step size for the motors
const float step_size = 1.8; // 1.8 degrees per step

void setup() {
  // Initialize the serial communication
  Serial.begin(9600);

  // Set the initial position of the motors to 0
  motor1.setCurrentPosition(0);
  motor2.setCurrentPosition(0);
}

void loop() {
  // Read the frame from the video stream
  // ...
```

sketch_apr29a | Arduino 1.8.19

File Edit Sketch Tools Help

sketch_apr29a §

```
  // Read the frame from the video stream
  // ...

  // Detect the object in the frame
  // ...

  // If the object is detected, track it with the motors
  if (len(outputs[0]['boxes']) != 0) {
    // Get the position of the detected object
    int x = (boxes[0][0] + boxes[0][2]) / 2;
    int y = (boxes[0][1] + boxes[0][3]) / 2;

    // Calculate the distance from the center of the frame
    int dx = x - (frame_width / 2);
    int dy = y - (frame_height / 2);

    // Calculate the number of steps to move the motors
    int steps_x = dx / step_size;
    int steps_y = dy / step_size;

    // Move the motors
    motor1.move(steps_x);
    motor2.move(steps_y);
  }

  // Update the motors
  motor1.run();
  motor2.run();
}
  // Check if the trigger is pulled
  if (digitalRead(trigger_pin) == HIGH) {
    // Fire the gun
    digitalWrite(gun_pin, HIGH);
    delay(gun_delay);
    digitalWrite(gun_pin, LOW);
  }

  // Update the motors
  motor1.run();
  motor2.run();
}
```

Chapter 5: Conclusion

In this thesis,we have developed a real-time object detection model using the Single Shot Detector (SSD) architecture to detect Indian soldiers along the border with Pakistan. The model is capable of accurately detecting Indian soldiers in real-time and sends their position to an Arduino connected to a camera with a turret and two steppers for tracking the soldier.

This model can greatly benefit the Indian military in monitoring the border and ensuring the safety of their country. By automating the detection process, it reduces the need for human monitoring and provides constant surveillance. Furthermore, it can detect Indian soldiers even in challenging weather and lighting conditions, which is a significant advantage over traditional monitoring techniques.

The dataset we created, including images of Indian and Pakistani soldiers, animals, civilians, and guns, is a valuable resource for future research in the field of object detection. The dataset can be used to train other models to detect similar objects and can also be expanded to include more categories. Overall, this project demonstrates the potential of object detection models in enhancing border security and can be applied to other similar scenarios, such as wildlife monitoring and surveillance of critical infrastructure.

Chapter 6: Future work needed to be done for the commercialization of this project

To commercialize a target tracking identification and engaging system, there are several key areas that need to be addressed. These include:

- **Improving the accuracy and reliability of the system:** A target tracking and identification system must be accurate and reliable, especially if it is to be used in critical applications such as defense or security. There may be a need to invest in further research and development to improve the system's accuracy, robustness, and resistance to interference.
- **Developing user-friendly interfaces:** To be effective, a target tracking system must be user-friendly and intuitive. This includes developing graphical user interfaces (GUIs) that allow users to easily configure the system, monitor its operation, and analyze its output.
- **Ensuring compatibility with existing systems:** The target tracking system must be compatible with existing systems, including those used by military, law enforcement, or security agencies. Compatibility issues may arise due to differences in protocols, hardware, or software, which need to be addressed through extensive testing and integration.
- **Meeting regulatory requirements:** Depending on the application and industry, there may be regulatory requirements that the system must comply with. This includes safety standards, data privacy regulations, and export controls. Compliance with these regulations must be ensured before commercializing the system.
- **Developing a business model:** A business model must be developed to determine how the system will be sold, marketed, and distributed. This includes identifying potential

customers, pricing the system, and establishing partnerships with suppliers, distributors, and other stakeholders.

- **Securing funding:** Commercializing a target tracking system requires significant funding for research, development, testing, and marketing. Securing funding from investors, government grants, or other sources may be necessary to finance the project.
- **Conducting field testing:** Before commercializing the system, extensive field testing must be conducted to ensure its effectiveness and reliability in real-world situations. This includes testing the system in various environmental conditions, against different types of targets, and with different users.

Overall, commercializing a target tracking identification and engaging system requires a comprehensive approach that addresses technical, regulatory, business, and funding challenges. Success will depend on the ability to develop a high-quality product that meets the needs, while also complying with regulatory requirements and creating a viable business model.

References and Work Cited

1. Petrou, M.M. and Petrou, C., 2010. *Image processing: the fundamentals*. John Wiley & Sons.

2. Sonka, Milan, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

3. Gonzales, Rafael C., and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

4. Xue, H., Liu, Y., Cai, D. and He, X., 2016. Tracking people in RGBD videos using deep learning and motion clues. *Neurocomputing*, *204*, pp.70-76.

5. Brunetti, A., Buongiorno, D., Trotta, G.F. and Bevilacqua, V., 2018. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, *300*, pp.17-33.

6. Zhang, X., Xu, Z. and Liao, H., 2022. Human motion tracking and 3D motion track detection technology based on visual information features and machine learning. *Neural Computing and Applications*, pp.1-13.

7. Junior, Julio Cezar Silveira Jacques, Soraia Raupp Musse, and Claudio Rosito Jung. "Crowd analysis using computer vision techniques." *IEEE Signal Processing Magazine* 27, no. 5 (2010): 66-77.

8. Moeslund, T.B. and Granum, E., 2001. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, *81*(3), pp.231-268.

9. Buch, Norbert, Sergio A. Velastin, and James Orwell. "A review of computer vision techniques for the analysis of urban traffic." *IEEE Transactions on intelligent transportation systems* 12, no. 3 (2011): 920-939.

10. Koushik, J., 2016. Understanding convolutional neural networks. *arXiv preprint arXiv:1605.09081*.

11. Kuo, C.C.J., 2016. Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, *41*, pp.406-413.

12. O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." *arXiv preprint arXiv:1511.08458* (2015).

13. Taye, Mohammad Mustafa. "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions." *Computation* 11, no. 3 (2023): 52.

14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.

15. Kumar, Ashwani, Zuopeng Justin Zhang, and Hongbo Lyu. "Object detection in real time based on improved single shot multi-box detector algorithm." *EURASIP Journal on Wireless Communications and Networking* 2020, no. 1 (2020): 1-18.

16. Kumar, Ashwani, and Sonam Srivastava. "Object detection system based on convolution neural networks using single shot multi-box detector." *Procedia Computer Science* 171 (2020): 2610-2617.

17. Brownlee, J., 2016. *Machine learning mastery with Python: understand your data, create accurate models, and work projects end-to-end*. Machine Learning Mastery.

18. Joshi, Prateek. *Artificial intelligence with python*. Packt Publishing Ltd, 2017.

19. N. Prabhakar, etal "Object Tracking Using Frame Differencing and Template Matching", Research Journal of Applied Sciences, Engineering and Technology 4(24): 5497-5501, 2012.

20. Amedome. M-D Kodjo, Y. Jinhua³Real-time Moving Object Tracking in Video´International Conference on Optoelectronics and Microelectronics (ICOM), 978-1-4673-2639-1/12, 2012.

21. PANDE R.P. etal ³detection of moving object with the help of motion detection alarm system in video surveillance Journal of Signal and Image Processing, Volume 3, Issue 3, pp.-118-121, 2012.

22. Hua-Jun Song and Mei-/L6KHQ³7arget Tracking algorithm based on Optical ORZ 0HWKRG XVLQJ &RUQHU 'HWHFWLRQ´ -RXUQDO RI PXOWLPHGLD WRROV DQG applications, volume 52, 121-131, march 2011.

23. Kai Du etal³MeanShift Tracking Algorithm with Adaptive Block Color Histogram´ Special Fund for Basic Scientific Research in the Central University (CHD2010ZY012), 978-1-4577-1415-3/12, 2012.