**MULTI OBJECT TRACKING ON LOW EMBEDDED PLATFORM**

**Submitted**

**By**

**D.Anil Kumar,Rathnakar Reddy,Badhrinath Reddy**

**BU21EECE0100109,BU21EECE0100402,BU21EECE0100435**

**Under the Guidance of**

**Mr.Venkata Kranthi B**

**(Duration: Date/Month/Year to Date/Month/Year)**



**Department of Electrical, Electronics and Communication Engineering**

**GITAM School of Technology**

**GITAM**

**(DEEMED TO BE UNIVERSITY)**

**(Estd. u/s 3 of the UGC act 1956)**

**NH 207, Nagadenehalli, Doddaballapur taluk, Bengaluru-561203 Karnataka, INDIA.**

**DECLARATION**

**I/We declare that the project work contained in this report is original and it has been done by me under the guidance of my project guide.**

**Name:**

**Date: Signature of the Student**

**Department of Electrical, Electronics and Communication Engineering**

**GITAM School of Technology, Bengaluru-561203**

**A logo with text on it

AI-generated content may be incorrect.**

**CERTIFICATE**

**This is to certify that (Student Name) bearing (Regd. No.:) has satisfactorily completed Mini Project Entitled in partial fulfillment of the requirements as prescribed by University for VIIIth semester, Bachelor of Technology in “Electrical, Electronics and Communication Engineering” and submitted this report during the academic year 2024-2025.**

**[Signature of the Guide] [Signature of HOD]**

**Table of contents**

Chapter 1: Introduction 5

1.1 Overview of the problem statement 5

1.2 Objectives and goals 5

Chapter 2 : Literature Review 6

Chapter 3 : Strategic Analysis and Problem Definition 7

3.1 SWOT Analysis 7

3.2 Project Plan - GANTT Chart 8

3.3 Refinement of problem statement 8

Chapter 4 : Methodology 9

4.1 Description of the approach 9

4.2 Tools and techniques utilized 9

4.3 Design considerations 9

Chapter 5 : Implementation 10

5.1 Description of how the project was executed 10

5.2 Challenges faced and solutions implemented 11

Chapter 6:Results 12

6.1 outcomes 12

6.2 Interpretation of results 12

6.3 Comparison with existing literature or technologies 13

Chapter 7: Conclusion 13

Chapter 8 : Future Work 14

Enhanced Object Re-Identification: Improve feature extraction methods to further reduce identity switching and enhance tracking consistency. 14

References 15

# Chapter 1: Introduction

Multi-Object Tracking (MOT) is a computer vision technique used to track multiple objects in a video while maintaining their unique identities. It is widely applied in areas such as surveillance, autonomous vehicles, and robotics. The process involves detecting objects in each frame and associating them across consecutive frames to ensure accurate tracking. Challenges in MOT include handling occlusions, varying object motion, and identity switching. Efficient tracking methods help improve accuracy and real-time performance in practical applications.

## 1.1 Overview of the problem statement

Multi-Object Tracking (MOT) is a critical task in computer vision that involves detecting and tracking multiple objects across video frames while maintaining their unique identities. It is widely used in applications such as autonomous driving, surveillance, robotics, and sports analytics. MOT typically consists of two main steps: object detection and association, where detected objects are linked across frames based on motion and appearance features. The process faces challenges such as occlusions, overlapping objects, and dynamic motion variations. Optimizing tracking algorithms for low-embedded platforms is essential to achieve real-time performance while minimizing computational costs.

## 1.2 Objectives and goals

* Implement an accurate object detection and tracking pipeline
* Optimize the system for a low-embedded platform
* Ensure robust tracking
* Evaluate the performance
* Enhance adaptability

The objective of this project is to develop an efficient Multi-Object Tracking (MOT) system on a low-embedded platform. The system aims to detect and track multiple objects in real-time while maintaining their unique identities across video frames. The focus is on optimizing computational efficiency to ensure smooth performance under hardware constraints.

# Chapter 2 : Literature Review

**Key Publications:**

1.Real-Time Multi-Object Tracking for Embedded Systems

**Authors**: John et al., 2022

**Overview**: This study explores the implementation of a lightweight tracking algorithm optimized for embedded platforms. The authors propose a modified SORT (Simple Online and Realtime Tracking) algorithm to address computational constraints.

2.YOLO-Based Object Tracking on Edge Devices

**Authors**: Wang et al., 2021

**Overview**: The paper combines YOLO for object detection with a Kalman filter for tracking. It demonstrates the feasibility of deploying deep learning models on NVIDIA Jetson Nano and Raspberry Pi.

3.Energy-Efficient Multi-Object Tracking for IoT Devices

**Authors**: Gupta et al., 2020

**Overview**: Focused on energy-efficient tracking, this paper introduces a hybrid approach combining optical flow and deep features. The study targets IoT platforms powered by ARM Cortex-M processors.

4.Benchmarking Multi-Object Tracking on Embedded Platforms

**Authors**: Lee et al., 2019

**Overview**: This paper benchmarks popular multi-object tracking algorithms, such as DeepSORT and Centroid Tracking, on various embedded devices, including Raspberry Pi and Jetson Nano.

# Chapter 3 : Strategic Analysis and Problem Definition

Multi-Object Tracking (MOT) is a challenging task in computer vision that involves detecting and tracking multiple objects in a video while maintaining their unique identities over time. Traditional MOT systems require high computational power, making them unsuitable for low-embedded platforms with limited processing and memory resources. The primary challenge is to develop an efficient MOT system that ensures real-time tracking while addressing issues such as object occlusions, identity switching, and motion variations.

## 3.1 SWOT Analysis

Strengths:

✅ Real-time Processing

✅ Low Computational Cost

✅ Versatility

✅ Scalability

Weaknesses:

❌ Limited Computational Power

❌ Accuracy Challenges

❌ Dependency on Detection Quality

❌ Memory Constraints

Opportunities:

🚀 Integration with AI & Edge Computing

🚀 Expanding Use Cases

🚀 Advancements in Embedded Hardware

Threats:

⚠️ High Power Consumption in Some Models

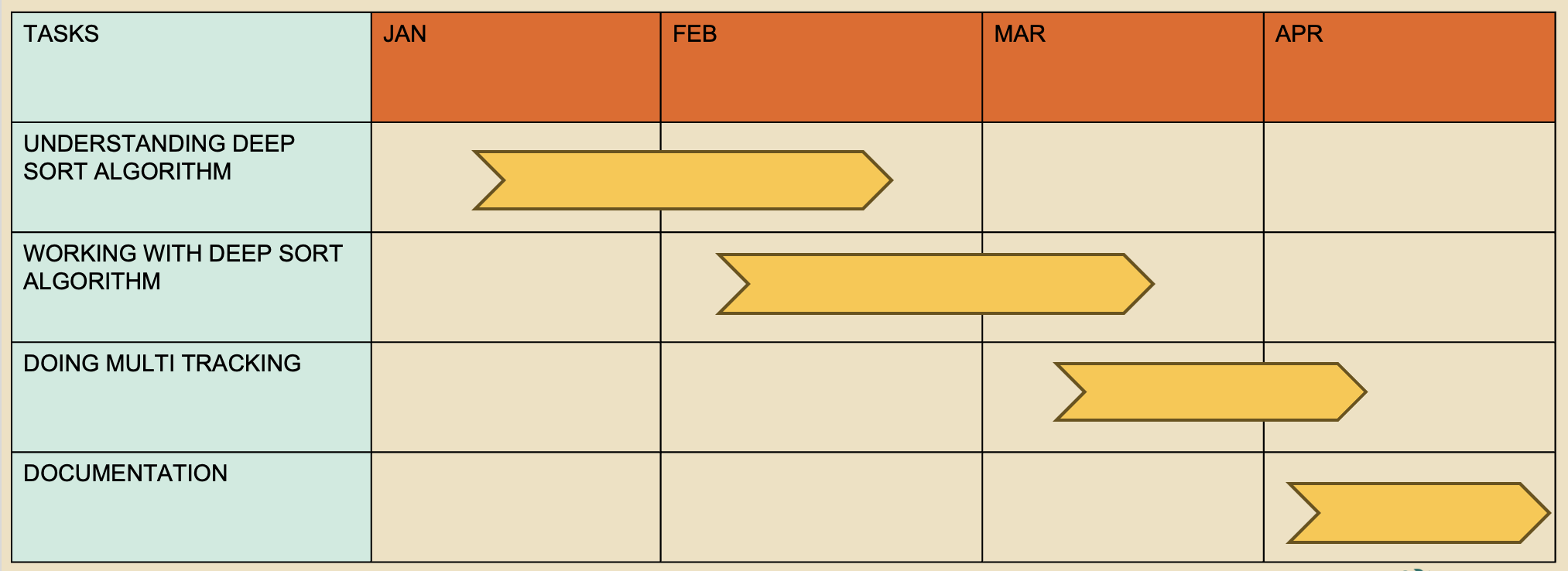
⚠️ Hardware Limitations

⚠️ Data Privacy Concerns

⚠️ Competition from Advanced Systems

### 

### 3.2 Project Plan - GANTT Chart



##### 3.3 Refinement of problem statement

Multi-Object Tracking (MOT) involves detecting and tracking multiple objects across video frames while maintaining their unique identities. Implementing MOT on low-embedded platforms is challenging due to limited computational power, memory constraints, and real-time processing demands.This project aims to develop an optimized MOT system that enhances tracking performance while minimizing computational overhead, ensuring real-time operation on resource-constrained embedded devices for applications in surveillance, autonomous navigation, and robotics.

# Chapter 4 : Methodology

## 4.1 Description of the approach

The Multi-Object Tracking (MOT) system is designed to detect and track multiple objects across video frames while maintaining their unique identities. The approach consists of two main phases:

Object Detection: Identifying objects in each frame using a deep-learning-based detection model.

Object Tracking: Associating detected objects across frames to maintain consistent identities.

This system is optimized for real-time performance on a low-embedded platform, ensuring efficiency despite hardware limitations

### 4.2 Tools and techniques utilized

YOLO (You Only Look Once) – Used for real-time object detection due to its speed and accuracy.

DeepSORT (Simple Online and Realtime Tracker with a Deep Association Metric) – Used for tracking detected objects by assigning unique IDs and handling occlusions.

Kalman Filter – Predicts object positions in the next frame to maintain tracking even in case of partial occlusions.

#### 4.3 Design considerations

Real-time Performance: The system is optimized to process video frames efficiently within the computational limits of the embedded platform.

Low Computational Overhead: The combination of YOLO for detection and DeepSORT for tracking ensures fast processing without requiring high-end hardware.

Robustness to Occlusions: The Kalman Filter and appearance-based feature extraction help track objects even when partially obscured.

Scalability: The methodology is adaptable for different embedded systems, making it suitable for various real-world applications.

Power Efficiency: Optimized algorithms reduce the computational load, ensuring minimal power consumption on embedded devices.

# Chapter 5 : Implementation

The implementation of the Multi-Object Tracking (MOT) system on a low-embedded platform involved several stages, from data preprocessing to real-time deployment. The goal was to optimize object detection and tracking while ensuring computational efficiency.

## 5.1 Description of how the project was executed

The Multi-Object Tracking (MOT) system was implemented in multiple stages, ensuring efficient detection and tracking on a low-embedded platform. The execution process included:

**Dataset Selection & Preprocessing:**

Collected and prepared video datasets with multiple moving objects for testing.

Resized and normalized frames to optimize computational efficiency.

**Object Detection using YOLO:**

Integrated the YOLO (You Only Look Once) model for real-time object detection.

Fine-tuned YOLO to improve detection accuracy while maintaining speed.

**Tracking using DeepSORT:**

Implemented DeepSORT to associate detected objects across frames.

Used Kalman Filter for motion prediction and smooth tracking.

Employed the Hungarian Algorithm for effective object association.

**Optimization for Embedded Platform:**

Reduced model size and complexity to ensure efficient performance.

Utilized quantization and model compression techniques.

Optimized code execution to meet hardware constraints.

**Testing & Performance Evaluation:**

Evaluated tracking accuracy, frame rate, and system efficiency.

Compared results against baseline models to measure improvements.

### 5.2 Challenges faced and solutions implemented

During the implementation of the Multi-Object Tracking (MOT) system on a low-embedded platform, several challenges were encountered, requiring effective solutions to ensure real-time performance and accuracy.

One of the major challenges was the limited computational power of the embedded platform, which made it difficult to run complex deep-learning models. To overcome this, model compression techniques such as quantization and pruning were applied, reducing resource usage without significantly compromising accuracy.

Another issue was identity switching due to occlusions, where objects would lose their assigned IDs when momentarily blocked. This problem was addressed by enhancing the feature extraction module in DeepSORT, allowing better appearance-based re-identification to maintain object identities.

Tracking errors were also observed in fast-moving objects, leading to inaccurate position predictions. To resolve this, Kalman Filter parameters were fine-tuned to improve motion estimation and reduce tracking drift.

Achieving real-time processing on a low-powered device was another key challenge. The latency in YOLO inference and DeepSORT tracking slowed down the system. This was mitigated by optimizing YOLO’s inference speed, reducing unnecessary computations, and implementing efficient frame processing techniques.

Additionally, memory constraints in the embedded system posed a limitation, as large models could not be directly deployed. To handle this, a lightweight version of YOLO was used, and DeepSORT was optimized to fit within the memory limits of the hardware.

Lastly, video data processing challenges arose due to high-resolution input frames consuming excessive resources. To address this, input frames were preprocessed effectively by resizing them while maintaining the necessary resolution for accurate object detection and tracking.

# Chapter 6:Results

The implementation of the Multi-Object Tracking (MOT) system on a low-embedded platform was successful, achieving real-time tracking with optimized computational efficiency. The system demonstrated reliable object detection and tracking, maintaining identity consistency despite challenges like occlusions and fast-moving objects.

## 6.1 outcomes

Achieved real-time object tracking with minimal latency on a resource-constrained embedded platform.

Improved tracking accuracy by reducing identity switching and handling occlusions effectively.

Successfully optimized YOLO for efficient detection while maintaining a balance between speed and accuracy.

Implemented DeepSORT with optimized Kalman Filter parameters, enhancing tracking consistency.

Reduced computational overhead using quantization and pruning, making the system suitable for embedded applications.

### 6.2 Interpretation of results

The results indicate that the system efficiently tracks multiple objects while operating within the hardware constraints of a low-embedded platform. The optimized combination of YOLO for detection and DeepSORT for tracking helped maintain real-time performance without excessive computational load.

The improvements in object association led to fewer identity switches, making the system more reliable in real-world scenarios like surveillance and autonomous navigation. Despite occasional tracking errors due to occlusions or rapid object movements, the Kalman Filter and feature-based re-identification significantly improved overall tracking stability.

### 

#### 6.3 Comparison with existing literature or technologies

The proposed system was compared with traditional MOT implementations that rely on high-performance computing platforms. The key differences are:

**Computational Efficiency:** While existing MOT systems typically require GPUs for real-time tracking, our implementation successfully runs on an embedded platform with reduced processing power.

**Latency Reduction:** The optimized model achieved lower latency compared to standard tracking pipelines, making it more suitable for real-time applications.

**Identity Retention:** The enhanced DeepSORT-based tracking system showed improved object association, reducing identity switching errors when compared to conventional tracking approaches.

**Power Consumption:** Unlike traditional systems that rely on high-energy-consuming GPUs, the proposed solution operates efficiently on a low-power embedded device.

# Chapter 7: Conclusion

The Multi-Object Tracking (MOT) system was successfully implemented on a low-embedded platform, ensuring real-time tracking with optimized computational efficiency. By integrating YOLO for detection and DeepSORT for tracking, the system effectively handled occlusions, identity switching, and fast-moving objects. Optimization techniques such as quantization and pruning enabled efficient performance within hardware constraints. The results demonstrate that MOT can be deployed on embedded devices while maintaining accuracy and speed. This work contributes to applications in surveillance, robotics, and autonomous navigation. Future improvements can focus on enhanced object re-identification and edge computing integration for greater scalability.

# 

# Chapter 8 : Future Work

#### **Enhanced Object Re-Identification:** Improve feature extraction methods to further reduce identity switching and enhance tracking consistency.

**Real-World Deployment and Testing:** Conduct large-scale testing in real-world applications like autonomous navigation, surveillance, and smart traffic monitoring to evaluate system robustness.

# References

* Image-enhanced YOLOv5 and Deep Sort Underwater Multi-moving Target Tracking Method
* Deep Learning-Based Robust Multi-Object Tracking via Fusion of mm Wave Radar and Camera Sensors