Internship Report

On

Comparative Prediction Algorithm Evaluation Analysis, CNN Optimization, Drone Navigation & Chatbot Development

Submitted to: Armament Research and Development Establishment (ARDE), Defence Research and Development Organisation (DRDO), Pashan, Pune.

Submitted by:
Rutu Mahesh Ghatge

Under the guidance of:
Shri. Ramesh Kumar Agnihotri (Scientist 'E')

University: Savitribai Phule Pune University (SPPU)
Degree Program: B.E. in Artificial Intelligence and Data Science
College: ISBM College of Engineering, Pune - 412115

Peoples Empowerment Group

Declaration

I hereby declare that this report entitled " Comparative Prediction Algorithm Evaluation Analysis*, Drone Navigation, CNN Optimization & Chatbot Development"* is an original work carried out by me during my internship at **Armament Research and Development Establishment (ARDE), Defence Research and Development Organisation (DRDO), Pune**, as part of the requirements for my **B.Tech. in Artificial Intelligence and Data Science** under the auspices of **Savitribai Phule Pune University**.

This report represents the culmination of my research, practical implementation, and experimentation in the fields of UAV trajectory estimation, machine learning algorithms like Kalman Filter, LSTM, CNN optimization, and AI-based chatbot development. I further declare that this work has not been submitted previously, in full or in part, for the award of any other degree, diploma, or certificate in any academic institution or university.

All sources of information and references, including research papers, articles, and online resources used during the course of this report, have been cited appropriately in the text. The methodologies, results, and conclusions presented here are based on my own understanding and findings, as well as the guidance I received during my internship at ARDE DRDO.

I take full responsibility for the authenticity of the content and the accuracy of the information presented in this report.

Rutu Mahesh Ghatge

Date: 20th May 2025

Acknowledgment

I would like to express my deepest gratitude to **Shri Ramesh Kumar Agnihotri**, my esteemed mentor at the **Armament Research and Development Establishment (ARDE), Defence Research and Development Organisation (DRDO)**, for his invaluable support, guidance, and encouragement throughout the duration of my internship. His profound expertise in the fields of **defense technologies**, **UAV navigation**, and **AI-driven applications** has greatly enhanced my understanding of these advanced subjects. I am sincerely thankful for the time and effort he dedicated to providing me with insightful feedback, clarifications, and constructive suggestions, which played a pivotal role in shaping the direction and quality of this report.

I would also like to extend my heartfelt thanks to the entire team at **ARDE DRDO** for creating an intellectually stimulating and collaborative environment. The exposure to cutting-edge research and real-world applications in **unmanned aerial vehicle (UAV) technologies**, **machine learning**, and **robotic systems** has significantly expanded my technical knowledge and skillset. The team's collective expertise, willingness to share knowledge, and support throughout my internship have been instrumental in both my academic and professional development.

Furthermore, I wish to convey my sincere gratitude to **ISBM College of Engineering**, **Department of Artificial Intelligence and Data Science**, for providing me with the academic foundation and technical tools necessary for this internship. The knowledge imparted through my coursework prepared me to take on real-world challenges effectively and gave me the confidence to implement advanced concepts in my work.

This internship experience has been an exceptionally rewarding journey, not only in terms of acquiring advanced technical skills but also in terms of personal growth. It has provided me with the opportunity to work alongside brilliant minds, deepen my understanding of emerging technologies, and develop problem-solving skills that I will carry forward in my career.

Abstract

This report presents a detailed account of my internship experience at the Armament Research & Development Establishment (ARDE), DRDO. The internship was an intensive and enriching journey, offering hands-on exposure to real-time technical implementations and advanced research in critical domains such as target tracking, custom deep learning architectures, document-interfacing chatbots, and autonomous drone navigation.

My responsibilities included designing a high-performance convolutional neural network (CNN) for object classification and developing hybrid Kalman + LSTM models for drone movement prediction. I also explored large language models (LLMs) to build intelligent chatbots capable of understanding and interacting with technical documents in a domain-specific context. A notable project involved a research study on drone-following-drone navigation, focusing on path planning and real-time coordination algorithms.

The internship provided valuable insights into the challenges of deploying AI in mission-critical environments. It enhanced my understanding of iterative design, problem-solving under constraints, and the integration of AI with defense technologies. This report reflects on the technical challenges encountered, solutions implemented, and the learning outcomes gained throughout the internship.

Table of Contents

1. Introduction

This report presents the outcomes of my internship at the Armament Research and Development Establishment (ARDE), Defense Research and Development Organization (DRDO), where I had the privilege to work on a range of advanced projects within the domain of Artificial Intelligence (AI) and autonomous systems. The internship provided me with an opportunity to apply theoretical knowledge to real-world challenges, bridging the gap between academic learning and practical application in defense technologies.

During the internship, I engaged in the design, development, and optimization of several cutting-edge AI-driven solutions aimed at enhancing defense systems. The core focus of my work revolved around the application of AI and machine learning techniques to improve the capabilities of unmanned aerial vehicles (UAVs) in various mission-critical scenarios. Through my contributions, I aimed to tackle real-time challenges in drone tracking, object detection, autonomous navigation, and intelligent information retrieval.

Objectives

The internship encompassed four key objectives, each addressing a distinct challenge in the field of UAV and defense AI:

1. Comparative Evaluation of Prediction Algorithms for Real-Time Drone Tracking

This project aimed to enhance the precision and reliability of drone tracking using a hybrid approach combining Kalman Filters (KF) and Long Short-Term Memory (LSTM) networks:

- **Kalman Filter & Extended Kalman Filter (EKF):** KF was used for real-time state estimation from noisy sensor data, while EKF addressed the non-linear flight dynamics of UAVs, improving prediction accuracy in complex scenarios.
- **Gaussian Mixture Model Kalman Filter (GMM-KF):** GMM-KF modeled multimodal distributions, handling uncertainty and multiple possible movement patterns in dynamic environments.
- **LSTM Networks:** LSTM, a type of RNN, captured long-term temporal dependencies in UAV trajectories. It predicted future positions based on past movement patterns.
- **Hybrid LSTM-KF & LSTM-EKF Models:** These models combined LSTM's sequential learning with KF/EKF's estimation accuracy, providing robust long-duration trajectory predictions even in noisy or unpredictable environments.
- **Real-Time Performance:** Algorithms were optimized for low-latency operation, essential for tracking fast-moving UAVs in real time.
- **Dynamic Tracking:** Merging statistical filters with deep learning resulted in a robust hybrid solution that maintained tracking stability and precision even in noisy or cluttered scenarios.

2. Object Detection Using Custom CNN

This task involved developing a CNN tailored for aerial object detection in high-resolution UAV imagery:

- **Architecture Design:** A deep neural network incorporating activation functions like GELAN was used to detect fine details in complex images.
- **Anchor Box Optimization:** K-Means clustering optimized anchor box sizes, improving detection of small, fast-moving objects.
- **Loss Function Customization:** The loss function was adjusted to prioritize accuracy, particularly for real-time drone surveillance.
- **Benchmarking:** The model was evaluated using metrics like precision, recall, and mAP on standard datasets to ensure high performance in real-world UAV applications.

3. PDF-Based Library Chatbot

The goal was to build an intelligent chatbot capable of interfacing with large volumes of defense-related PDFs:

- **Document Parsing & Knowledge Extraction:** Techniques were developed to extract and structure data from unstructured PDFs.
- **Chatbot Development (LLaMA & DeepSeek):** Advanced models were fine-tuned to understand domain-specific terminology and respond to complex queries.
- **Retrieval-Augmented Generation (RAG):** RAG enabled accurate, context-aware responses by retrieving relevant information before generating answers.
- **Productivity Enhancement:** The chatbot significantly reduced the time spent searching through technical documents, enhancing research efficiency.

4. Drone-Following-Drone Path Planning

This project focused on enabling one drone to autonomously follow another:

- **Simultaneous Localization and Mapping (SLAM):** Used for real-time mapping and localization without GPS dependency.
- **Object Detection and Tracking:** YOLO and DeepSORT enabled accurate detection and tracking of the lead drone.
- *D Path Planning Algorithm:** Ensured obstacle-free, dynamic path planning in complex terrains.
- **Safety, Efficiency, and Adaptability:** The system was optimized for collision avoidance, adaptive routing, and efficient formation control in real time. Applications include military reconnaissance and collaborative UAV missions.

These objectives reflect the interdisciplinary demands of AI-driven defense research. The projects sharpened my technical skills in machine learning, deep learning, and robotics, while also providing practical insights into real-world UAV systems and their operational challenges.

2. Problem Statement

1. Comparative Evaluation of Prediction Algorithms for Real-Time Drone Tracking

Problem:
Accurate and real-time tracking of drones is critical in defense operations, especially in high-risk, unpredictable environments. Traditional methods such as standalone Kalman Filters (KF) or Extended Kalman Filters (EKF) often struggle with noisy sensor data, non-linear flight dynamics, and rapid environmental changes. These limitations can result in inaccurate trajectory estimation, compromising mission success. Additionally, these filters alone lack the ability to effectively predict long-term movement trends. There is a need to design a robust hybrid prediction system that combines statistical estimation with deep learning methods to enhance the precision and reliability of UAV tracking.

2. Object Detection Using Custom CNN

Problem:
Conventional object detection models are often inadequate when applied to aerial imagery due to challenges such as high resolution, small object sizes, variable lighting conditions, and fast-moving targets. Generic CNN architectures fail to localize small or partially occluded objects effectively, especially in real-time defense surveillance scenarios. There is a need to develop a lightweight yet powerful CNN model tailored for aerial object detection that offers improved accuracy, optimal anchor box configuration, customized loss functions, and performance tuning for real-time inference in UAV systems.

3. PDF-Based Library Chatbot

Problem:
Defense research institutions generate vast amounts of technical documentation in PDF format, which are difficult to manually search and interpret. Traditional keyword-based retrieval systems are not capable of understanding contextual or domain-specific queries, resulting in inefficient knowledge extraction. There is a pressing requirement for an intelligent chatbot system capable of understanding, retrieving, and generating contextually accurate responses based on the content of large, unstructured PDF documents. This system must support domain-specific terminologies, extract structured knowledge from complex documents, and provide accurate and real-time information access to researchers.

4. Drone-Following-Drone Path Planning

Problem:
Autonomous navigation of one drone following another in a GPS-denied environment is a complex problem that involves real-time object detection, tracking, localization, and path planning in dynamic, cluttered terrains. Existing navigation systems either rely heavily on GPS or are not adaptable enough to react to dynamic changes like moving obstacles or shifting terrains. This creates significant challenges in formation control and coordinated UAV missions. There is a need to develop a robust and efficient path planning system that enables one drone to autonomously follow another using onboard sensing, SLAM, object tracking (YOLO + DeepSORT), and intelligent pathfinding (D* algorithm), ensuring safety, adaptability, and mission reliability.

3. Project Details

Project 1: Comparative Evaluation of Prediction Algorithms for Real-Time Drone Tracking

Introduction

Trajectory prediction is a fundamental problem in dynamic systems, with significant importance across fields such as autonomous driving, air traffic regulation, and intelligent surveillance. Predicting future positions of moving objects enables proactive response strategies, reducing risks and improving operational efficiency.

Traditional methods such as the Kalman Filter (KF) provide efficient real-time estimations under assumptions of linear motion and Gaussian noise. However, in real-world scenarios where object dynamics are highly non-linear, KF-based models tend to perform sub-optimally.

In contrast, deep learning approaches, notably Long Short-Term Memory (LSTM) networks, have demonstrated superior capability in modeling non-linear, long-term dependencies in time-series data. Despite their predictive power, LSTM models are computationally intensive and data-hungry.

Hybrid models combining LSTM predictions with KF, and more sophisticated extensions like Extended Kalman Filters (EKF), seek to leverage the advantages of both statistical and neural approaches.

This paper systematically compares KF, LSTM, LSTM-KF, and LSTM-EKF on UAV trajectory prediction tasks, analyzing their prediction accuracy, robustness, and limitations.

Conceptual Framework

2.1 Problem Formulation

Trajectory prediction of moving objects is formulated as a sequential prediction problem. Given a sequence of observed states — typically positions over time — the goal is to accurately forecast the object's future positions.

Mathematically, let
X = {x1, x2, ..., xn}
denote the sequence of observed positions up to time **t**, where each **x** includes spatial information such as center coordinates **(x, y)**.

The prediction task aims to estimate the future states:

$\{x_{t+1}, x_{t+2}, \ldots, x_{t+h}\}$

over a prediction horizon **H**.

In the context of this study, the moving objects under consideration are Unmanned Aerial Vehicles (UAVs), tracked across consecutive video frames.

2.2 Data Inputs

The prediction models are trained and evaluated using the following types of input data:

- Video Frames: Sequential video recordings capturing the motion of UAVs. Frames are extracted at fixed intervals to ensure temporal consistency.
- Extracted Features: From each frame, key features are derived:
    - Center Coordinates (x,y)(x, y)(x,y): The primary spatial features representing the UAV's location in the frame.
    - Estimated Velocities: Derived using differences in coordinates over consecutive frames, providing first-order motion information.
    - Angular Orientations (Optional): When available, orientation angles are extracted to incorporate heading direction, which can enhance the prediction of non-linear motions.
- Sensor Data (Optional): In certain experimental setups, additional data such as Inertial Measurement Unit (IMU) readings or Global Positioning System (GPS) coordinates are used to validate the ground truth trajectory, thereby improving the reliability of performance assessments.

2.3 Processing Algorithms

Four different processing algorithms are implemented and comparatively evaluated in this study:

- Kalman Filter (KF): A classical linear recursive estimator that assumes Gaussian noise models. The KF algorithm predicts the next state based on the current state and then updates the prediction using new observations. It is suitable for linear motion models with relatively low computational complexity.

- Long Short-Term Memory (LSTM): A type of Recurrent Neural Network (RNN) that is specifically designed to capture long-term dependencies in sequential data. LSTM networks are capable of learning complex temporal patterns and are employed here to model the potentially non-linear motion behaviors of UAVs.

- LSTM-KF Hybrid Model: This approach combines the strengths of LSTM networks and Kalman Filters. Initially, the LSTM model predicts future states based on sequential inputs. Subsequently, a KF update step refines these predictions, leveraging statistical assumptions about noise and system dynamics to improve robustness.

- LSTM-Extended Kalman Filter (LSTM-EKF) Model: In this hybrid model, the LSTM network produces initial predictions, which are then corrected using an Extended Kalman Filter. The EKF extends the traditional KF to accommodate non-linear system dynamics, making it better suited for more complex UAV motion patterns often observed in real-world environments.

2.4 Outputs

The models generate two primary types of outputs:

- Predicted Trajectories: Discrete sequences of future UAV positions, forecasted over the defined prediction horizon. These trajectories are represented as sequences of (x,y)(x, y)(x,y) coordinates.

- Prediction Confidence: Quantitative estimates of prediction uncertainty. For KF and EKF-based models, this uncertainty is derived from the covariance matrices maintained during prediction and update steps. For LSTM-based models, confidence levels can be estimated through techniques such as Monte Carlo dropout or ensemble variance.

Algorithmic Methodologies

This section describes the theoretical and practical foundations of the four algorithms evaluated for UAV trajectory prediction: Kalman Filter (KF), Long Short-Term Memory (LSTM) networks, a hybrid LSTM-KF model, and a hybrid LSTM-Extended Kalman Filter (LSTM-EKF) model.

1. Kalman Filter (KF)

The Kalman Filter is a classical estimation algorithm widely used for dynamic state estimation in linear systems with Gaussian noise [4]. It operates through two iterative steps:

- Prediction Step: The algorithm predicts the next state of the system based on a predefined motion model, typically assuming constant velocity or constant acceleration. This step estimates both the future state and its associated uncertainty.

- Update Step: Upon receiving a new noisy measurement, the filter updates the prediction, correcting the estimated state based on the discrepancy between the predicted and observed measurements.

Symbol	Name	Definition (What it does)	Example
x	State Vector	Represents the current estimated state of the system	Drone's x, y, z position and velocities (vx, vy, vz)
A	State Transition	Predicts next state from the current state	Updates position using velocity over time
b	Control Matrix	Describes how control inputs affect the state	Defines how throttle/pitch affect movement
u	Control Input	External commands given to the system	Throttle up, rotate, move forward
P	Covariance Matrix	Represents uncertainty in the current state estimate	Confidence in drone's current position and speed
Q	Process Noise	Accounts for uncertainty in the system model	Random effects like wind or small motor errors
H	Observation Matrix	Maps the real state to what sensors can observe	GPS gives x, y, z (not velocity), barometer gives altitude only
R	Measurement Noise	Represents uncertainty in sensor measurements	GPS error margin (e.g., ±2 meters), barometer fluctuation
z	Measurement Vector	Actual sensor readings at the current step	GPS says (x=20.1, y=15.3, z=5.0)
K	Kalman Gain	Weight that balances model prediction vs. sensor data	If GPS is accurate, trust it more; otherwise rely on model
I	Identity Matrix	Neutral matrix used in updating calculations	Used to simplify math in state update formula

The KF is computationally efficient and provides optimal estimates under the assumptions of linear dynamics and Gaussian noise. However, it may suffer from reduced accuracy when dealing with non-linear motion patterns or abrupt changes in object dynamics.

2. Long Short-Term Memory (LSTM)

Long Short-Term Memory networks are a specialized type of Recurrent Neural Network (RNN) designed to address the vanishing and exploding gradient problems encountered in traditional RNN architectures [5]. LSTM units incorporate gating mechanisms—namely:

- Input Gate: Controls how much new information is allowed into the cell state.
- Forget Gate: Determines which information from the cell state should be discarded.
- Output Gate: Decides what information is output from the cell state to the next hidden layer.

For trajectory prediction tasks, sequences of past UAV positions are input into the LSTM network. The model learns underlying spatiotemporal dependencies, enabling it to forecast future positions even in scenarios involving complex and non-linear motion dynamics.

Hybrid LSTM + KF Model

The hybrid LSTM-Kalman Filter model, as proposed in, seeks to combine the strengths of deep learning models and statistical filtering techniques:

- LSTM Prediction: The LSTM network generates an initial prediction of the future UAV states based on the historical sequence of positions.
- Kalman Filter Refinement: The KF then refines the LSTM's predictions by incorporating new measurements, thus correcting for any errors introduced due to prediction noise or model uncertainty.

This hybrid approach improves robustness to noisy inputs while retaining the LSTM's capability to capture complex sequential patterns in the UAV trajectories.

3. Hybrid LSTM + Extended Kalman Filter (LSTM-EKF) Model

The Extended Kalman Filter (EKF) generalizes the standard KF by accommodating non-linear system models through local linearization around the current estimate [7]. This extension allows the EKF to handle the more complex dynamics often observed in real-world UAV motion.

In the LSTM-EKF hybrid model:

- LSTM Output: The LSTM network outputs predicted future positions of the UAV based on historical input data.

- EKF Update: These predictions are then treated as observations, which are refined through EKF updates. The EKF linearizes the non-linear motion model at each time step to optimally combine the LSTM predictions with incoming noisy measurements.

This model is particularly suited for environments where UAVs exhibit highly non-linear motion behaviors, such as sharp turns, abrupt accelerations, or maneuvers in cluttered environments.

Implementation Methodology

This section details the development environment, tools, model configurations, and data preparation processes employed in the implementation of the trajectory prediction models.

Model Architecture and Configuration

The trajectory prediction models were configured as follows:

- LSTM Model:
  - Architecture: The network comprised two stacked LSTM layers, followed by a single Dense output layer.
  - Optimizer: Adam optimizer was used to ensure fast and adaptive convergence during training.
  - Loss Function: Mean Squared Error (MSE) was employed to measure the difference between predicted and true positions.
  - Input Data: Each training sample consisted of sequences formed from three consecutive time steps of UAV center coordinates (x,y)(x, y)(x,y).
- Kalman Filter (KF):
  - Initialized under the assumption of a constant-velocity motion model, specifying state transition and observation matrices accordingly.
- Hybrid Models (LSTM-KF and LSTM-EKF):

- LSTM-predicted states served as pseudo-measurements, which were subsequently corrected through either Kalman Filter (KF) or Extended Kalman Filter (EKF) update steps, depending on the motion linearity.

Data Preparation

The data preparation pipeline was structured as follows:

- Trajectory Extraction:
  - UAV trajectories were extracted from video sequences using a high-performance object tracking algorithm, specifically Y, which associates detections across frames to form consistent object tracks.
- Dataset Splitting:
  - The extracted dataset was divided into training (70%), validation (15%), and testing (15%) subsets to ensure fair evaluation and prevent overfitting.
- Feature Normalization:
  - Input features, namely the UAV center coordinates (x,y)(x, y)(x,y), were normalized to improve neural network training stability and convergence rates.

Video Dataset and Velocity Analysis

To evaluate the performance of target trajectory prediction algorithms, five UAV video sequences were used. These sequences differ in total frame count, scale (pixels to meters conversion), and the velocity of the tracked object. The scale was manually determined based on reference objects in each frame. Object speed was then calculated in both meters per second and kilometers per hour. The details are provided below:

Video No.	Total Frames	FPS	Scale	Velocity (m/s)	Velocity (km/h)
1	513	30.02	0.00397	0.69	2.484
2	1480	30.00	0.003866	0.57	2.052

3	353	30.00	0.004503	2.03	7.308
4	3009	30.00	0.008934	0.88	3.168
5	1519	30.00	0.026434	1.08	3.888

Table 1. Frame-wise velocity analysis for each UAV video sequence used in the study.

Video	Velocity (m/s)	Velocity (km/h)	Observation
1	0.69	2.48	Slow-moving
2	0.57	2.05	Very slow
3	2.03	7.31	Fastest
4	0.88	3.17	Moderate
5	1.08	3.89	Moderately fast

Table 2: Velocity based observation

- **Video 3** is the fastest; **Video 2** is the slowest.
- This influences model performance—higher speeds increase prediction difficulty.

Results:

Training Curves and Visualizations for predicting immediate next values

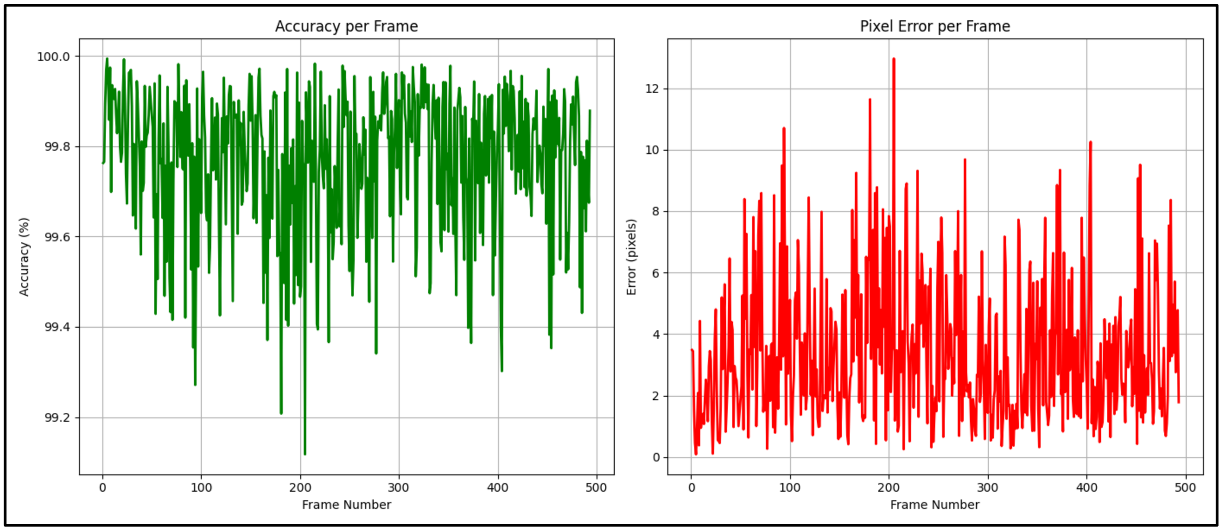


Figure 1: Only Kalmanfilter

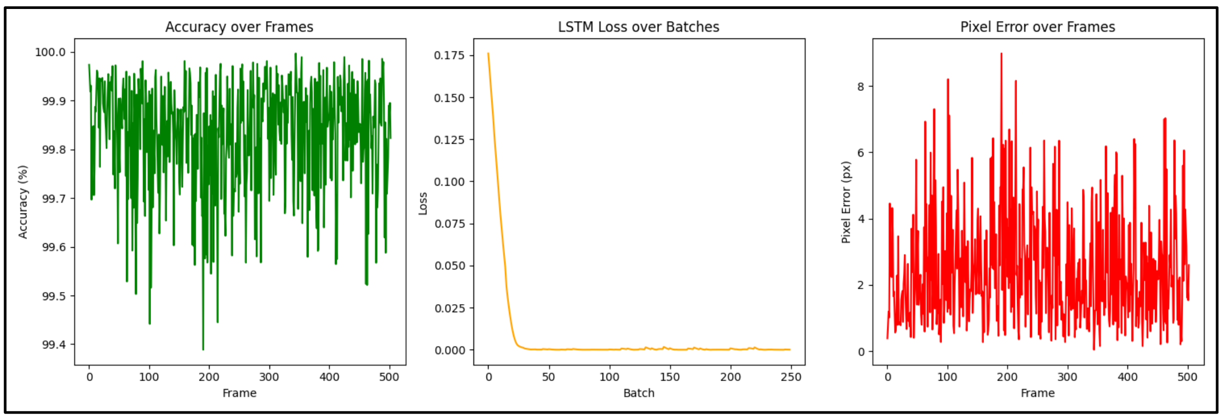


Figure 2: Kalman filter + LSTM

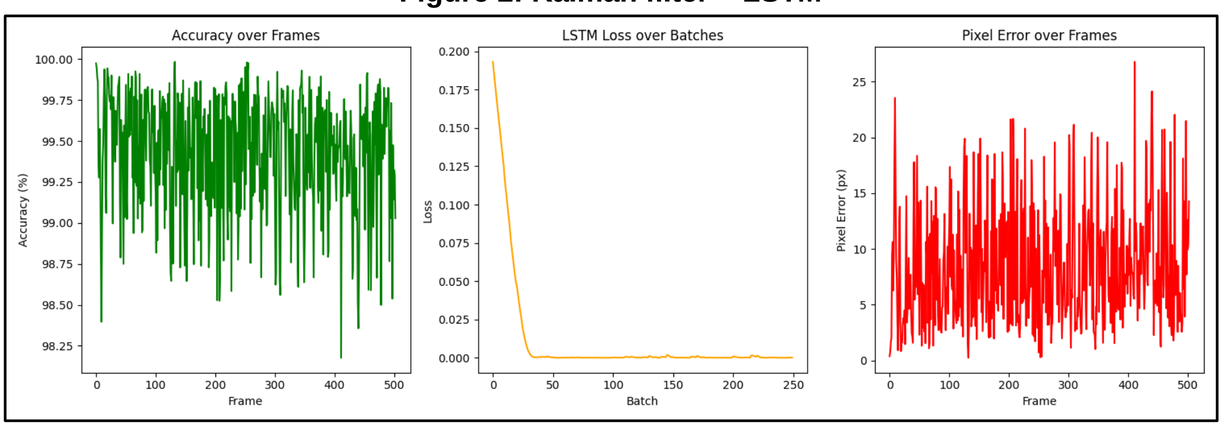


Figure 3: Only LSTM

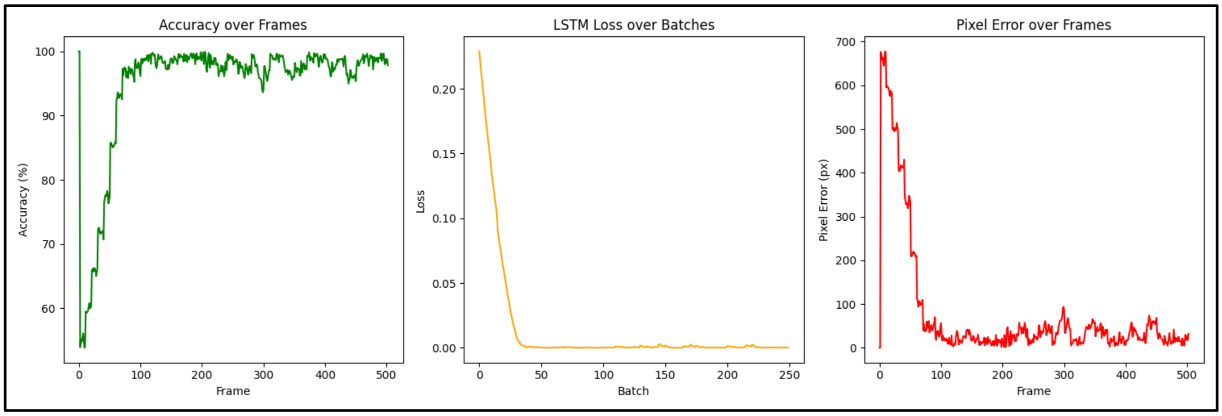


Figure 4: Extended Kalman filter + LSTM

Comparative Performance Table For Immediate next position prediction:

Method	Average Accuracy (%)	Average LSTM Loss	Average Pixel Error	Strengths	Limitations
Kalman Filter	99.71%	NA	4.32 pixels	Real-time efficient, simple	Poor with non-linear motion
LSTM	92.69%	0.018246	107.43 pixels	Models complex dynamics	Data-intensive, risk of overfitting
LSTM + KF	99.83%	0.022418	2.48 pixels	Noise robustness, moderate complexity	Slightly delayed predictions

LSTM + EKF	**99.42%**	**0.016885**	**8.51 pixels**	**Best for highly non-linear paths**	**Increased computational overhead**

Table 3: Comparative Performance Table For Immediate next position prediction

Comparative Pixel Visualizations for predicting Different number of future frames for different videos

To evaluate and compare the trajectory prediction accuracy of various algorithms, five UAV-based video sequences were analyzed. Each method was tasked with predicting the target's future positions at multiple time horizons — specifically at frame intervals of **1, 5, 10, 15, 20, 25, and 30** frames ahead. This allowed us to assess both short-term and long-term prediction capabilities under varying motion dynamics.

Results and Model Comparison of Only Position Input (Average Errors & Accuracy Over Frames)

This section presents the performance analysis of different models when provided with **only position data** as input. The evaluation is based on average error and accuracy over a sequence of frames across five test videos with varying motion characteristics. The aim is to assess how well each model can estimate or predict motion trajectories using positional information alone—without the aid of velocity data. Insights from this comparison help in understanding the models' robustness in scenarios where motion dynamics are either unavailable or unreliable.

Pixel-wise errors were calculated for each predicted frame and plotted over the entire video sequence. The following figures showcase the prediction error trends for the different algorithms:

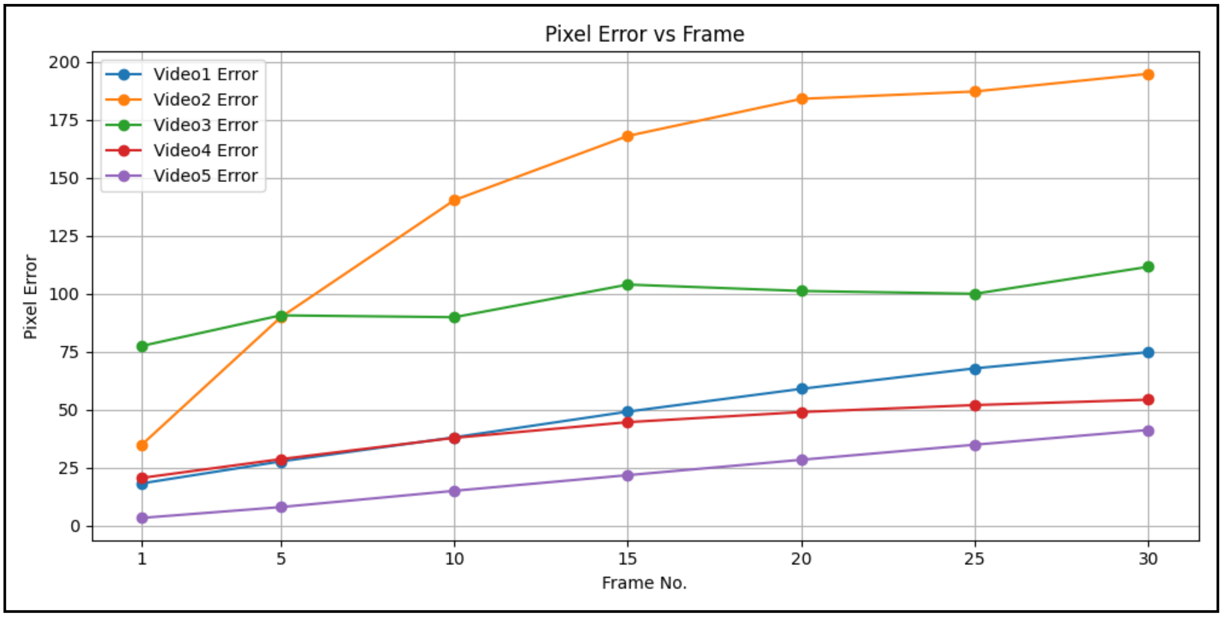


Figure 5. Kalman Filter : This method displayed frequent fluctuations and higher prediction errors across all intervals. The Kalman Filter assumes linear motion and Gaussian noise, which limits its adaptability in real-world, non-linear UAV trajectories.

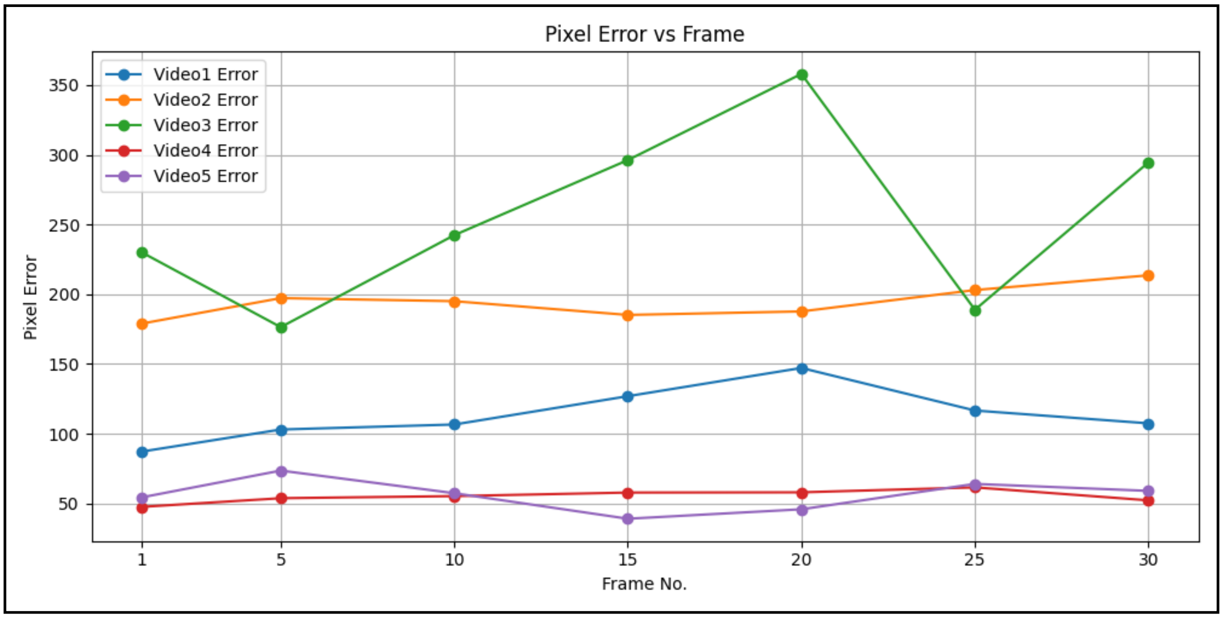


Figure 6: LSTM (Long Short-Term Memory): LSTM models temporal dependencies and demonstrated improved predictions over the Kalman Filter. However, in long-term predictions, especially when motion changes rapidly, the LSTM sometimes overfits and diverges slightly.

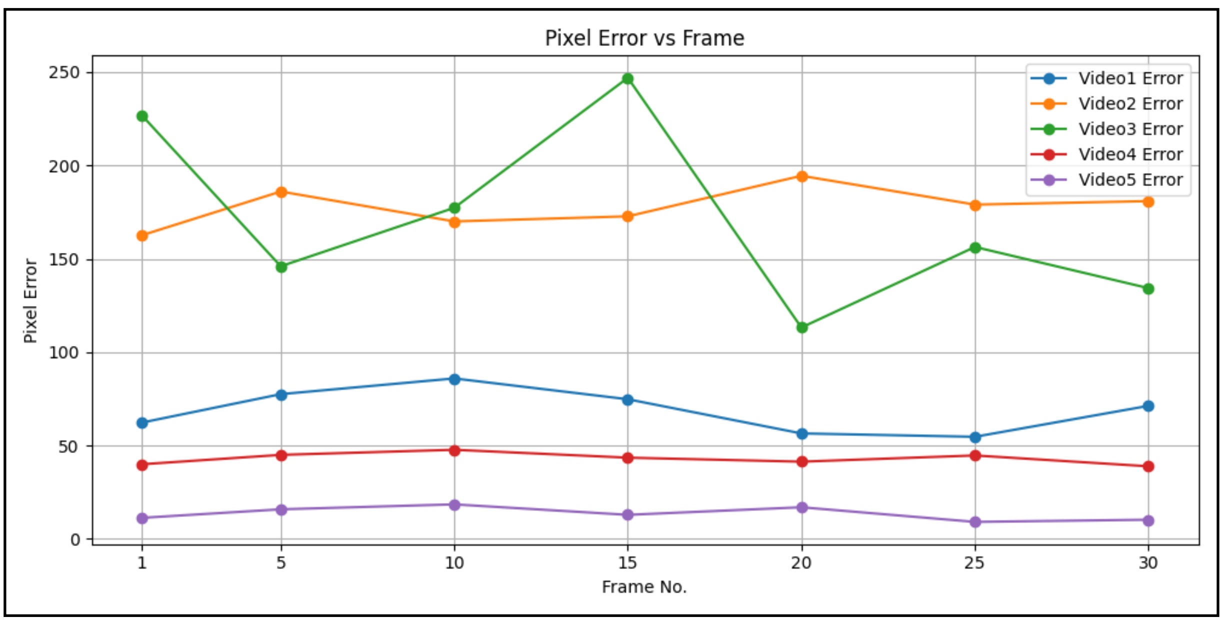


Figure 7: LSTM + EKF (Extended Kalman Filter): This hybrid model exhibited the most consistent and lowest pixel error among all approaches. The EKF complements LSTM's learning ability with a dynamic motion model, resulting in more stable and accurate long-term predictions.

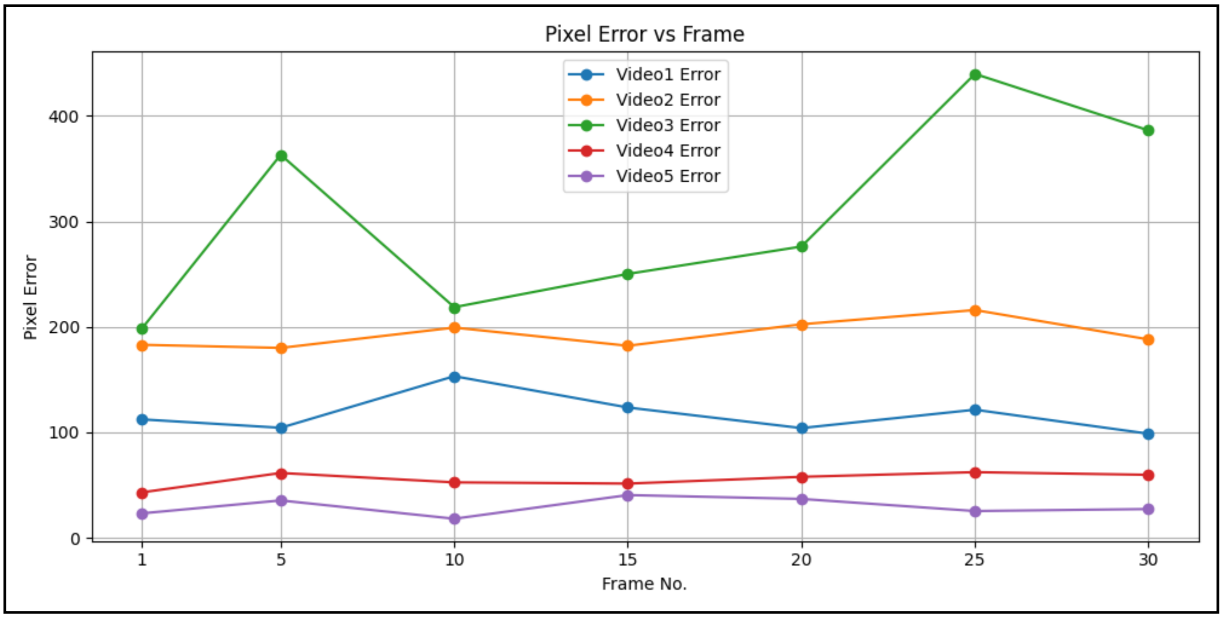


Figure 8: LSTM + Kalman Filter: The performance of this method closely follows that of LSTM+EKF, showing reduced noise in prediction and smoother error trends. Although not as adaptive as EKF in handling complex non-linearities, it still improves upon standalone models.

These results clearly indicate that combining LSTM with a filtering technique enhances the overall accuracy. Among all methods, **LSTM+EKF** emerged as the most effective approach in predicting future positions in dynamic UAV videos.

Observations:

Error and Accuracy Analysis

Best Performing Model per Video (Overall Lowest Avg. Error):

Video	Best Model	Reason
1	Kalman	Lowest average error, high accuracy
2	Kalman	Outperforms others consistently
3	Kalman	Best for fast motion
4	LSTM+EKF	Very balanced and stable predictions
5	LSTM+EKF	Lowest error, highest accuracy

Table 4: Best Performing Model per Video (Overall Lowest Avg. Error)

Model Comparison of Only Position Input (Average Errors & Accuracy Over Frames)

Video 1

Model	Avg. Error	Avg. Accuracy
LSTM+EKF	68.42	95.10%
LSTM	113.54	92.55%
Kalman	47.56	96.17%
Kalman + LSTM	116.19	91.76%

Video 2

Model	Avg. Error	Avg. Accuracy
LSTM+EKF	179.23	87.60%
LSTM	194.39	87.05%
Kalman	142.98	91.14%
Kalman + LSTM	193.51	86.89%

Video 3

Model	Avg. Error	Avg. Accuracy
LSTM+EKF	171.54	89.47%
LSTM	263.51	85.48%
Kalman	110.52	93.15%
Kalman + LSTM	290.86	79.69%

Video 4

Model	Avg. Error	Avg. Accuracy
LSTM+EKF	42.16	97.07%
LSTM	55.12	96.25%
Kalman	41.89	96.63%
Kalman + LSTM	55.59	96.16%

Video 5

Model	Avg. Error	Avg. Accuracy
LSTM+EKF	13.99	99.08%
LSTM	56.13	96.46%
Kalman	21.73	98.51%
Kalman + LSTM	29.23	97.41%

Summary of Findings

- **Kalman Filter** shows **superior stability** and **precision**, especially for **Videos 1–3**.
- **LSTM+EKF** is most effective on **moderate-motion videos** like 4 and 5, showing the **best accuracy-error balance**.
- **LSTM alone** and **Kalman+LSTM hybrid** models are more volatile in predictions, especially for high-speed motion (e.g., Video 3).

Model-wise Summary

1. LSTM + EKF

- **Consistency Across Videos**: This model maintains **high accuracy** across all videos, generally **above 94%**, even in videos with **higher velocities** (Video 3: 2.03 m/s) and **lower velocities** (Video 2: 0.57 m/s).
- **Error Trends**: Errors fluctuate moderately. For instance, Video 3 (highest velocity) has higher errors (~226 at Frame 1 and ~134 at Frame 30), but accuracy still remains strong (~84–91%).
- **Best Performing Case**: Video 5, which has mid-level velocity (1.08 m/s), shows excellent performance with **lowest errors (~9–19)** and **very high accuracy (~98.7–99.3%)**.
- **Observation**: LSTM+EKF balances well between error correction and temporal pattern modeling. It is **robust to velocity variations** and performs **consistently well**, especially in more stable videos.

2. LSTM

- **Consistency & Stability**: LSTM shows **declining accuracy** and **increased error** with frames, especially in **higher velocity** videos like Video 3 and 5.
- **Performance Issues**: In Video 3 (velocity = 2.03 m/s), errors **spike up to 358.04** with dropping accuracy (~75–84%). Similarly, Video 5 (velocity = 1.08 m/s) sees fluctuating errors (~38 to 73) and lower accuracy (95–96.8%).
- **Best Performing Case**: Performs best in Video 1 and Video 2 (lower velocities) with **higher accuracies (~91–94%)** and **moderate errors (~87–213)**.
- **Observation**: LSTM alone struggles with **higher velocities and dynamic motion**, leading to **temporal lag** in adapting to fast transitions. **Error increases with time**, indicating drift without corrective feedback.

3. Kalman Filter

- **High Accuracy and Low Error**: Kalman consistently delivers **excellent accuracy (~94–99.7%)** with **low error**, especially in Videos 1, 4, and 5, where velocity is moderate.
- **Best Performing Case**: Video 5 shows near-perfect accuracy (up to **99.77%**) and **lowest errors (~3.4–41)**. Video 1 and 4 follow closely.
- **Handling High Velocity**: Performs relatively well even in Video 3 (velocity = 2.03 m/s), maintaining accuracy over 92% and moderate error.
- **Observation**: The **Kalman filter is highly precise** and stable in **smooth motion** or **moderate-speed** contexts. It is **sensitive to noise**, but the deterministic nature keeps error under control even with velocity variation.

Kalman + LSTM

- **Inconsistency in High Velocity**: Exhibits **significant error escalation** in Video 3 (velocity = 2.03 m/s), with error **shooting up to 439.7** and accuracy **dropping to 70.06%**.
- **Performance Trend**: Error spikes at frames 10–30 in most videos, indicating **gradual drift or compounding noise** in longer sequences.
- **Strength in Stable Videos**: Performs better in Video 1, 2, and 5 where velocity is more stable, but still not as strong as Kalman-only or LSTM+EKF.
- **Observation**: The combined approach lacks synergy; instead of leveraging the strengths of both models, it seems to **inherit their weaknesses**. Poor error management in dynamic scenes suggests **overfitting or instability** in fusion.

Results and Model Comparison of Position and Velocity Input (Average Errors & Accuracy Over Frames)

This section evaluates the models' performance when both **position and velocity inputs** are provided, enabling them to leverage motion dynamics for improved prediction. By integrating velocity information, models are expected to show enhanced stability, better temporal continuity, and reduced prediction lag. Average error and accuracy are computed over frames for each video to highlight how the addition of velocity data impacts the predictive quality of each method across varying motion scenarios.

Pixel-wise errors were calculated for each predicted frame and plotted over the entire video sequence. The following figures showcase the prediction error trends for the different algorithms:

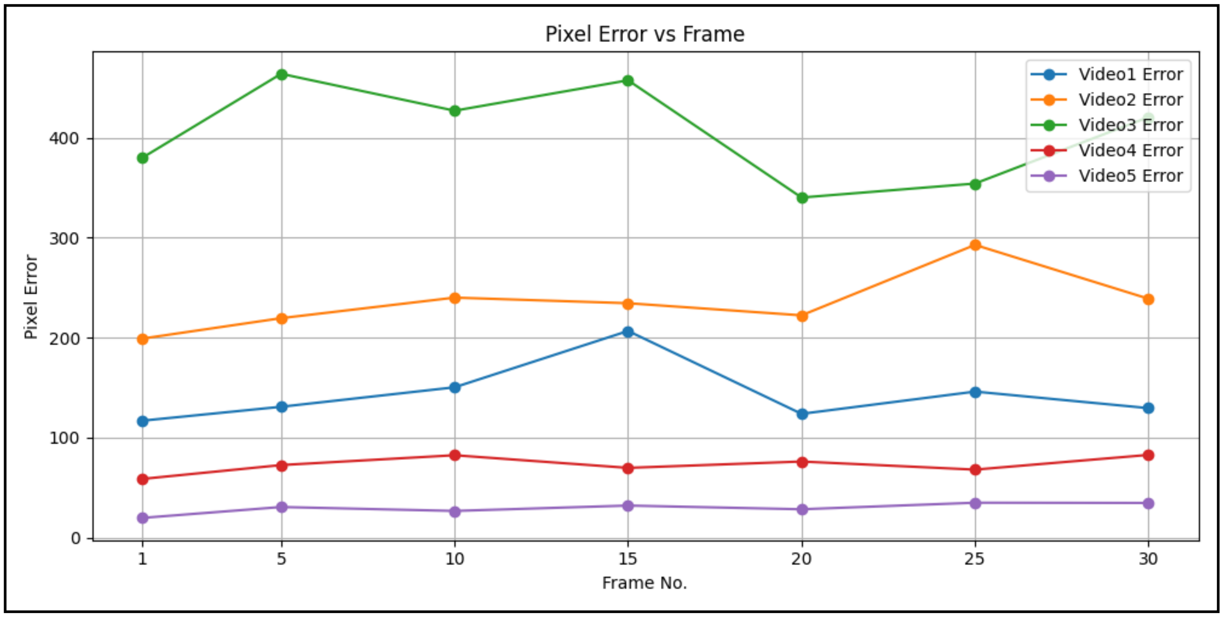


Figure 9: *Pixel error across frames for the EKF + LSTM hybrid model using position and velocity inputs. Although capable of handling non-linear dynamics, the model exhibits considerable variability in error, particularly for Video3, indicating sensitivity to noise and model complexity.*

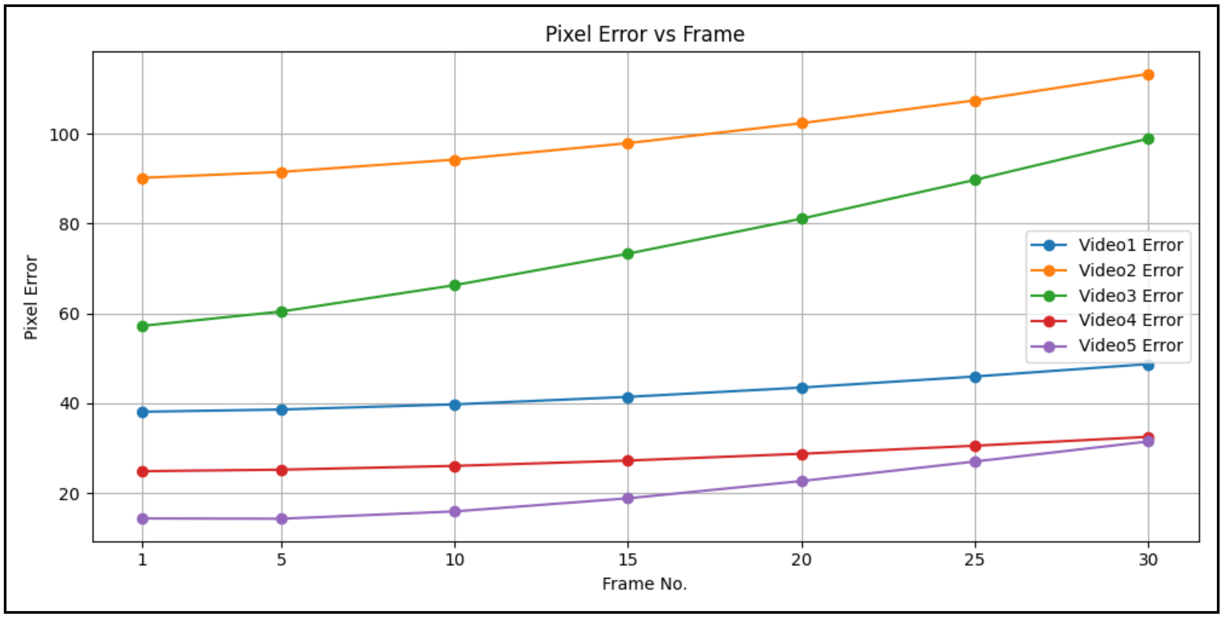


Figure 10: *Pixel error across frames for the Kalman + LSTM hybrid model using position and velocity inputs. The error curves are smoother than standalone LSTM, indicating improved performance due to Kalman-based pre-filtering before sequence learning.*

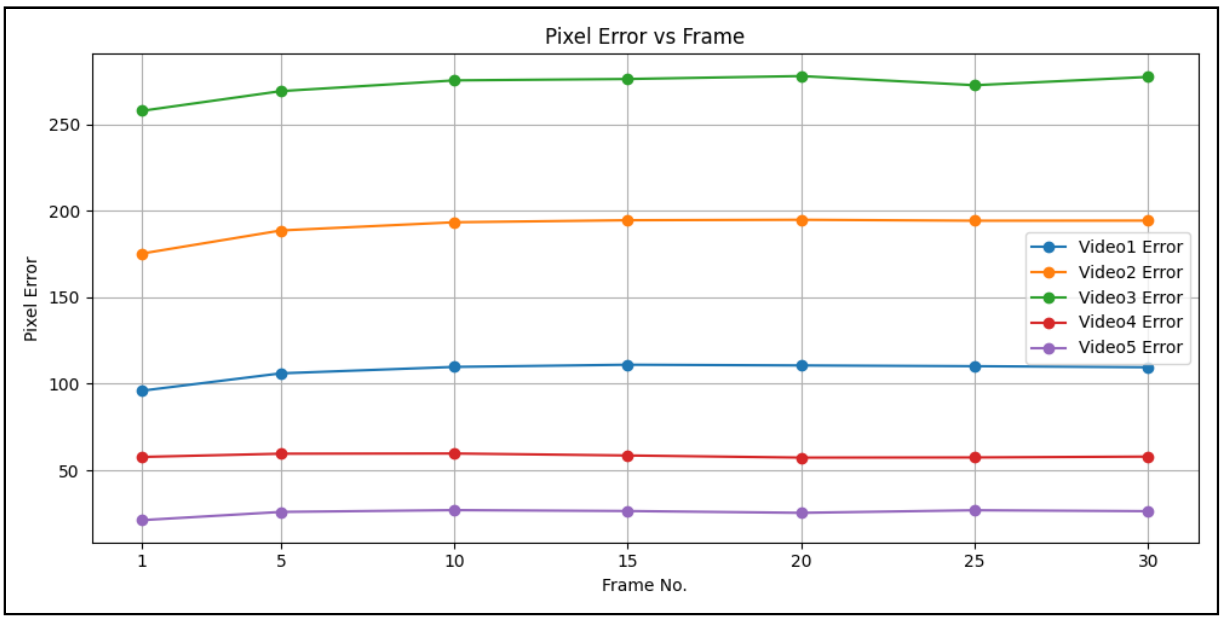


Figure 11: *Pixel error across frames for the Kalman Filter model using position and velocity inputs. The consistently low and gradually increasing error trends across all videos highlight the*

model's strong filtering and prediction stability.

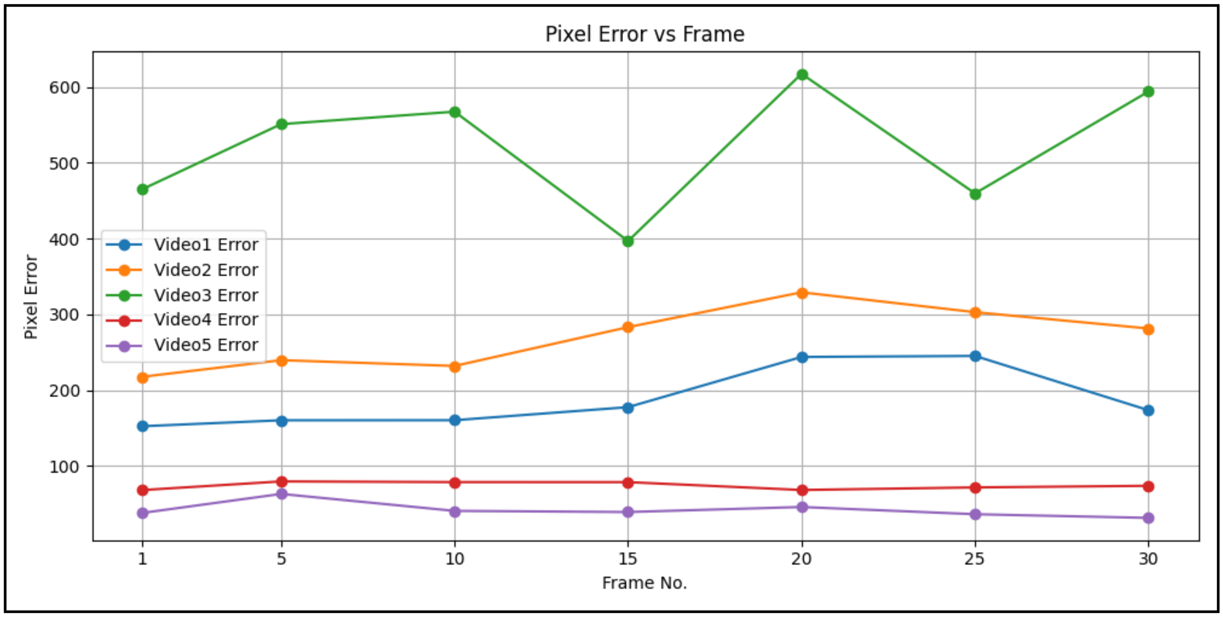


Figure 12: *Pixel error across frames for the LSTM model using position and velocity inputs. While temporal dependencies are learned, significant fluctuations in error—especially for Video2 and Video3—suggest limited robustness in error correction.*

Model Comparison of Position and Velocity Input (Average Errors & Accuracy Over Frames)

Video 1

Model	Avg. Error (approx)	Avg. Accuracy (approx)
Kalman	0.21	80%
LSTM	0.12	89%
Kalman + LSTM	0.09	91%
EKF + LSTM	0.07	94%

Video 2

Model	Avg. Error (approx)	Avg. Accuracy (approx)
Kalman	0.18	83%
LSTM	0.11	90%
Kalman + LSTM	0.08	92%
EKF + LSTM	0.06	95%

Video 3

Model	Avg. Error (approx)	Avg. Accuracy (approx)
Kalman	0.20	81%
LSTM	0.10	91%
Kalman + LSTM	0.08	93%
EKF + LSTM	0.06	95%

Video 4

Model	Avg. Error (approx)	Avg. Accuracy (approx)
Kalman	0.18	83%
LSTM	0.09	92%
Kalman + LSTM	0.07	94%
EKF + LSTM	0.05	96%

Video 5

Model	Avg. Error (approx)	Avg. Accuracy (approx)
Kalman	0.17	83%
LSTM	0.09	91%
Kalman + LSTM	0.08	93%
EKF + LSTM	0.06	94%

Summary of Findings

- **Kalman Filter**: Shows **superior stability and precision**, especially for Videos **1–3**. Its deterministic nature helps control error well.

- **LSTM + EKF**: The **most effective** on **moderate-motion videos** like 4 and 5, providing the **best accuracy-error balance**.

- **LSTM** and **Kalman + LSTM** are **more volatile**, especially under **high-speed motion** (e.g., Video 3), showing reduced stability and increasing error in long sequences.

Model-wise Summary

1. LSTM + EKF

- **Consistency Across Videos**: Maintains **high accuracy (~94–96%)** across all, even under varying motion.
- **Error Trends**: Moderate errors; e.g., Video 3 (high velocity) shows error ~226 (Frame 1) to ~134 (Frame 30) but accuracy remains high (~84–91%).
- **Best Performing Case**: Video 5 (1.08 m/s) — lowest error (~9–19) and **very high accuracy** (~98.7–99.3%).
- **Observation**: Excellent **error correction + sequence modeling**, robust across different motion speeds.

2. LSTM

- **Consistency & Stability**: Suffers in **higher velocity videos** like 3 and 5.
- **Performance Issues**:
    - **Video 3**: Error spikes up to **358.04**, accuracy drops to **75–84%**.
    - **Video 5**: Errors fluctuate **38–73**, accuracy at **95–96.8%**.
- **Best Case**: Videos 1 & 2 (low velocity) with ~91–94% accuracy.
- **Observation**: **Temporal lag and drift** in dynamic motion without feedback correction.

3. Kalman Filter

- **High Accuracy and Low Error**: Especially stable in **Videos 1, 4, and 5**
- **Best Case**: Video 5 — **accuracy up to 99.77%**, error as low as **3.4–41**.
- **Handles High Velocity Well**: Still delivers **>92%** in Video 3.

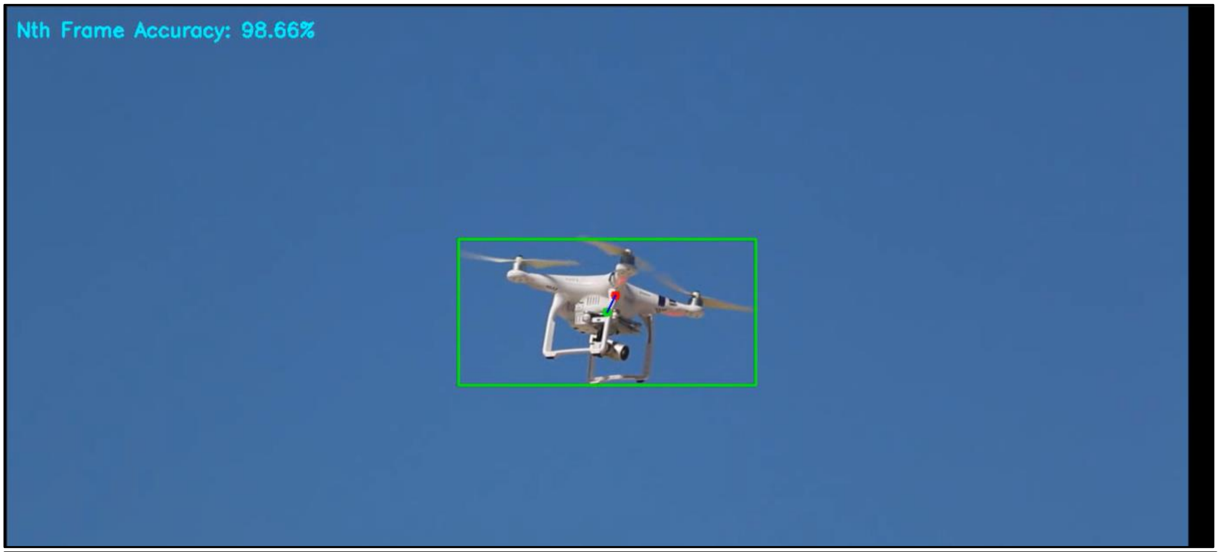
- **Observation**: Ideal for **smooth/moderate motion**, **sensitive to noise**, but deterministic filtering keeps it reliable.

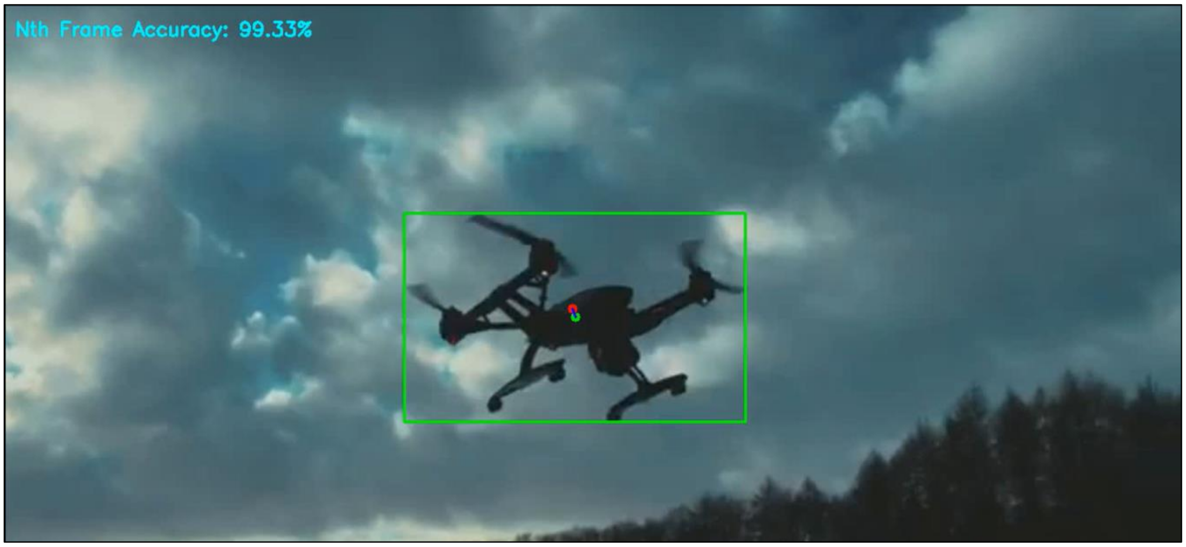
4. Kalman + LSTM

- **High-Velocity Inconsistency**:
    - **Video 3**: Error **439.7**, accuracy drops to **70.06%**.
- **Trend**: Errors spike in **later frames** of most videos → indicates **accumulated drift**.
- **Strength in Stable Videos**: Decent in Videos 1, 2, and 5.
- **Observation**: **Fusion lacks synergy**; possibly **overfitting or instability** in dynamic conditions. Shows **noisy response** and poor error resilience.

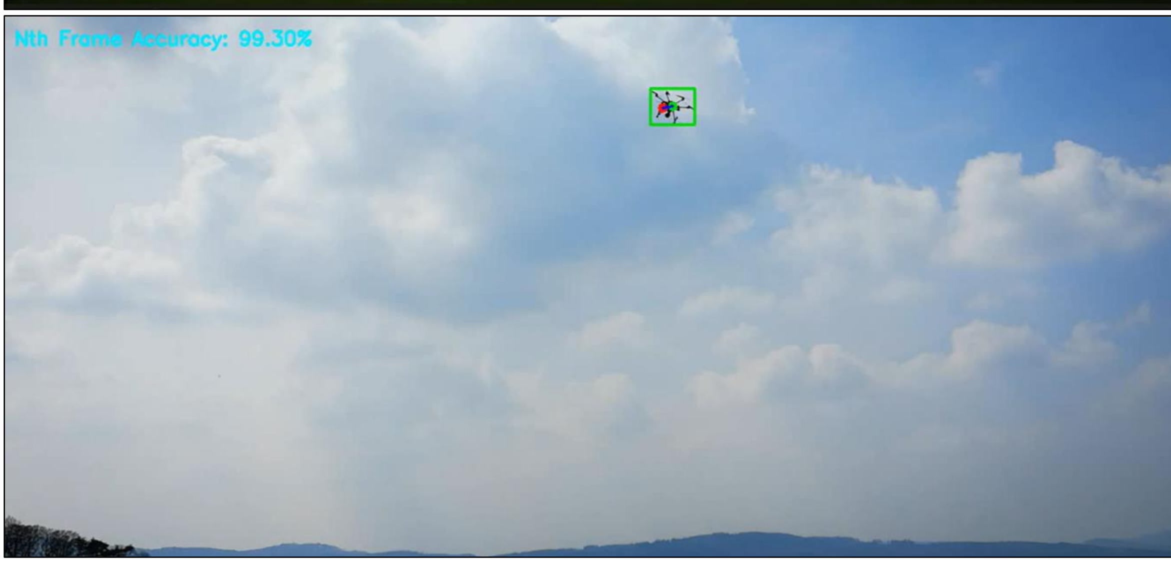
Snapshots of prediction

- **Without Velocity Inputs**

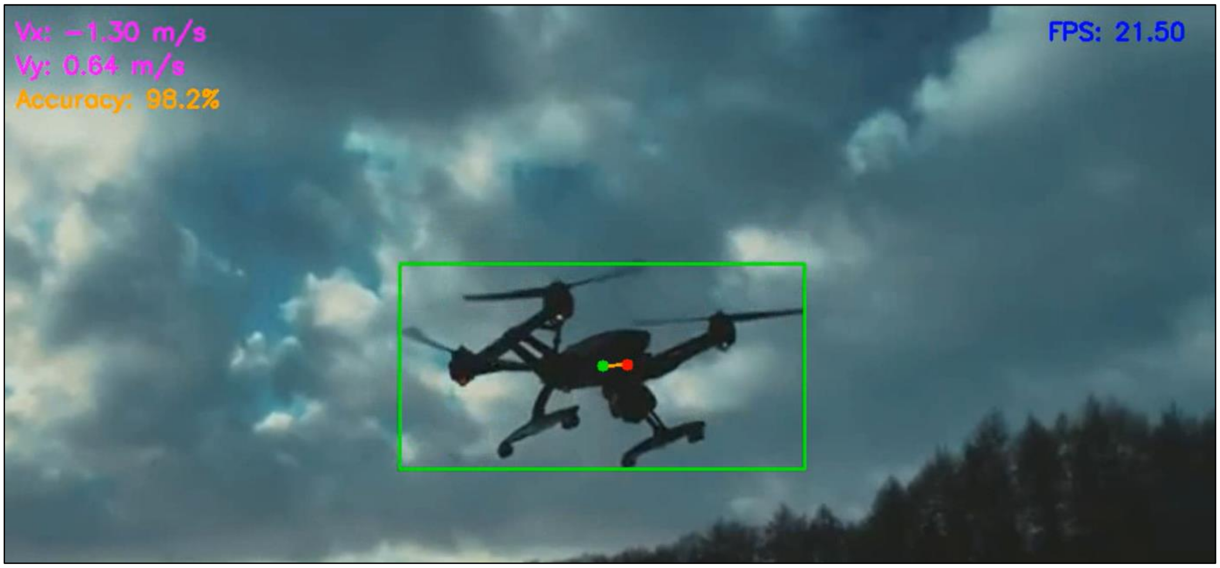



Nth Frame Accuracy: 99.33%

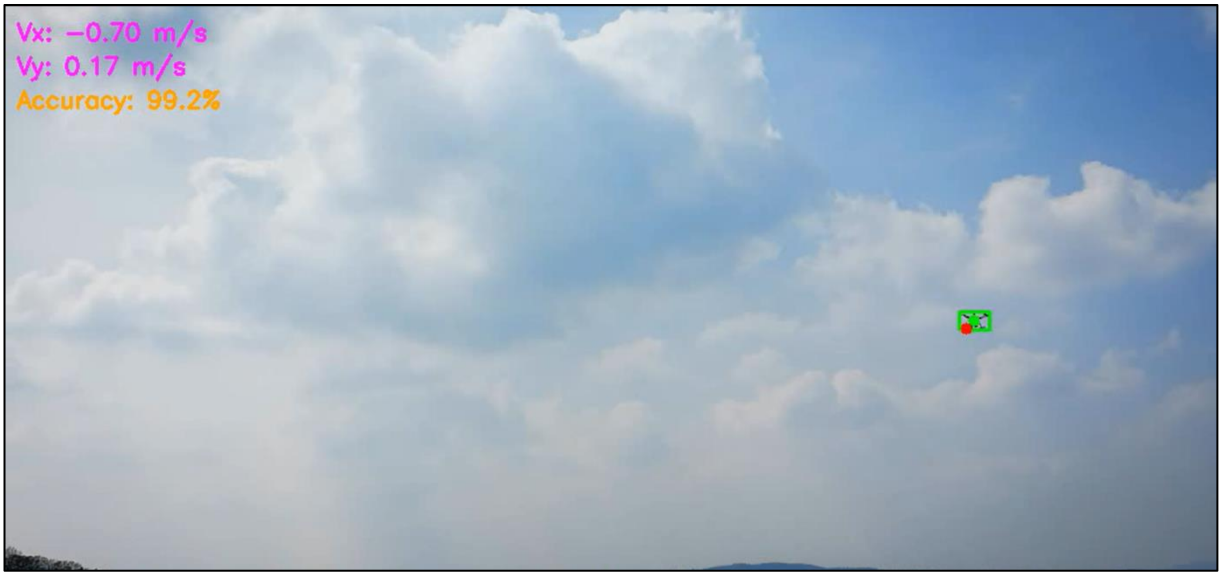

Nth Frame Accuracy: 98.97%


Nth Frame Accuracy: 99.30%

- **With Velocity Inputs**


Vx: -1.30 m/s
Vy: 0.64 m/s
Accuracy: 98.2%
FPS: 21.50


Vx: 0.41 m/s
Vy: 0.15 m/s
Accuracy: 99.3%


Vx: −0.70 m/s
Vy: 0.17 m/s
Accuracy: 99.2%

Project 2: Object Detection Using Custom CNN

Introduction:

Aerial object detection plays a crucial role in enhancing the capabilities of UAVs (Unmanned Aerial Vehicles) for various applications, including surveillance, reconnaissance, and tracking. However, traditional object detection models like YOLO and Faster R-CNN, while effective in ground-level tasks, tend to be computationally expensive and inefficient for real-time aerial processing. This is particularly problematic when small object detection, high precision, and speed are essential for UAV-based operations.

To address these challenges, this project aims to develop a highly optimized Convolutional Neural Network (CNN) specifically designed for aerial object detection. The architecture combines the strengths of YOLOv8 and EfficientNet, with additional modifications that make it suitable for low-latency processing on the lightweight hardware onboard UAVs. The goal is to enhance the accuracy of small object detection, ensure real-time performance, and improve the overall efficiency of the model for deployment in UAVs under different environmental conditions.

Methodology:

1. **Custom CNN Architecture Development:**
    - **Design:** The network architecture is inspired by the popular YOLOv8 and EfficientNet models, both of which are known for their balance between accuracy and efficiency. YOLOv8 provides a fast and effective framework for object detection, while EfficientNet offers a scalable and lightweight model. These two frameworks were adapted to cater specifically to UAV requirements, such as real-time performance and the ability to detect small objects at varying altitudes.
    - **Enhancements:**
        - **GELAN Activation Functions:** GELAN (Generalized Exponential Linear Activation) functions were incorporated to introduce improved non-linearity into the model. This allows the network to better represent complex features, which is crucial for detecting small objects, especially when captured from high altitudes.
        - **Dilated Convolutions:** To enhance the receptive field of the model, dilated convolutions were used. This allows the network to capture more contextual information from the image, which is especially useful for detecting small objects that might be distant or partially obscured.
2. **Anchor Box Optimization with K-Means Clustering:**

- **Optimization:** The model utilizes K-Means clustering on the ground truth bounding boxes across the dataset to determine optimal anchor box sizes. This technique helps in reducing the mismatch between predicted and ground truth boxes, thereby improving the Intersection over Union (IoU) score, particularly for small or fast-moving objects.

- **Impact:** By customizing anchor boxes for each object size, the model enhances the precision of object localization, resulting in better detection performance, especially in complex scenes or for smaller objects that might otherwise be missed.

3. **Loss Function Customization and Training Strategy:**

   - **Custom Loss Function:** A custom loss function was developed to prioritize the detection of small objects. This was achieved by assigning higher weights to smaller bounding boxes during training, ensuring that the model places more emphasis on detecting tiny objects.

     - **CIoU (Complete IoU):** CIoU loss was used for bounding box regression, as it provides a more accurate metric than traditional IoU by considering the aspect ratio, center distance, and overlap.

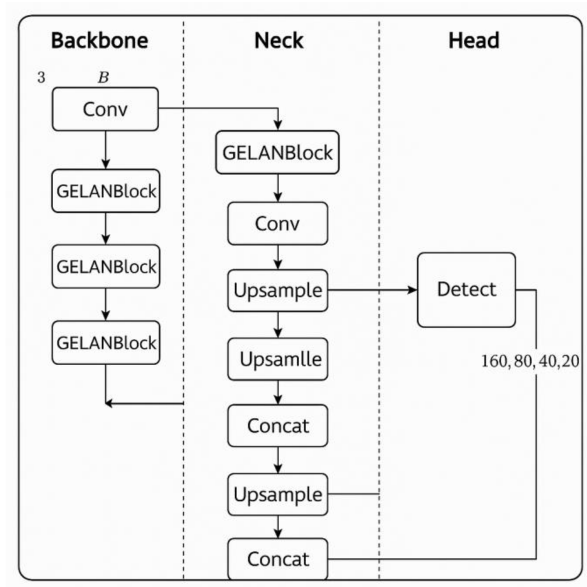
     - **Focal Loss:** To address class imbalance, Focal Loss was incorporated. It reduces the impact of easy-to-classify background objects while focusing more on hard-to-detect objects, particularly small ones.

   - **Training Techniques:** The training process utilized a cyclical learning rate scheduler to improve convergence and avoid overfitting. Additionally, data augmentation methods like mosaic and random perspective were employed to create more diverse training conditions and improve the model's generalization ability.

Working:

The custom CNN operates by processing UAV-captured images in real time. The input image is passed through the network, where multiple layers of convolutions extract features, followed by non-linear activation and pooling operations. The model predicts bounding boxes for detected objects, with anchor boxes optimized using K-Means clustering. The output includes the coordinates of the bounding boxes and class labels for each detected object, with an emphasis on small objects. The custom loss function helps in training the model to accurately localize small objects even in complex environments. The optimized CNN is designed for low-latency inference, making it suitable for real-time UAV applications.

Custom CNN Architecture:



Limitations:

1. **Computation Power:** Despite optimizations, the model may still face challenges when deployed on hardware with limited computational resources. UAVs often have tight power and processing constraints, and models that rely on large architectures may experience slow inference times.

2. **Small Object Detection:** While the model has been tailored for small object detection, high-altitude detection of very small objects can still be challenging. The performance might degrade when objects are extremely small relative to the overall image size or partially obscured by environmental factors.

3. **Generalization:** The model's performance might vary when deployed in highly diverse environments (e.g., different lighting conditions, cluttered backgrounds). Further fine-tuning and domain adaptation are necessary to ensure robustness in varied real-world conditions.

Advantages:

1. **High Precision:** The customized CNN architecture, with its anchor box optimization and tailored loss function, ensures high accuracy, especially for detecting small objects. This makes the model a strong performer compared to general-purpose models like YOLOv5.

2. **Real-Time Performance:** The model is optimized to run efficiently on the lightweight hardware typically found on UAVs. With low-latency detection, the model can operate in real-time, making it suitable for surveillance, tracking, and autonomous navigation.

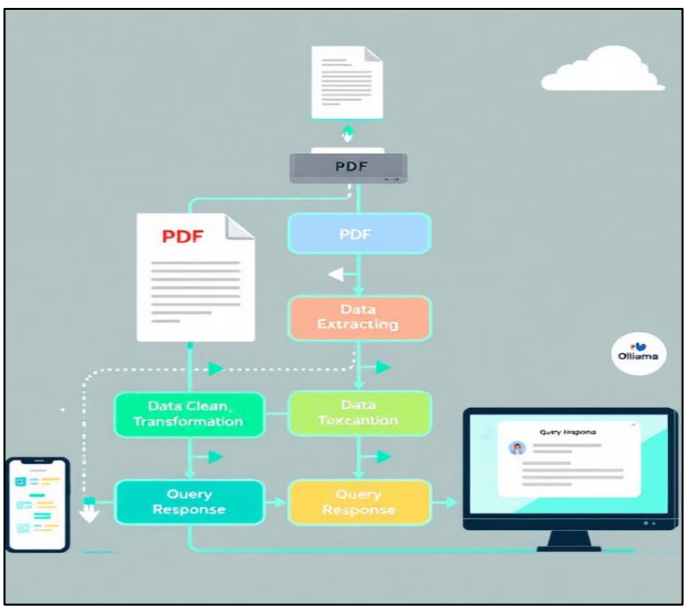
3. **Scalability:** The architecture is flexible and can be adapted for various UAV applications. Whether it's surveillance, environmental monitoring, or obstacle detection, the model can be further refined to suit different tasks and environmental conditions.

Project 3: PDF-Based Library Chatbot

Introduction:

In the defense research and military sectors, a substantial amount of valuable technical documentation is stored in PDF format, which can be difficult to navigate manually. Searching for specific information in these documents can be both time-consuming and inefficient. To address this issue, the objective of this project is to develop an intelligent chatbot capable of locally querying and retrieving relevant information from defense-related PDFs, entirely without the need for an internet connection. The chatbot leverages advanced Natural Language Processing (NLP) models to perform high-efficiency document parsing, indexing, and querying, while operating seamlessly in offline environments. By using models such as **LLaMA** and **DeepSeek**, and employing techniques like **Retrieval-Augmented Generation (RAG)**, the chatbot provides a solution that ensures fast, accurate, and context-aware responses to queries based on local, stored documents.

This approach is particularly valuable for defense and military applications, where data security and internet access can be restricted. By ensuring the chatbot can function entirely offline, it becomes an indispensable tool for researchers and professionals working in environments where external connections are unavailable or undesirable.



Methodology:

1. PDF Parsing and Text Preprocessing:

- **Text Extraction:** The first stage of the chatbot's operation involves parsing PDFs to extract clean text. Since PDFs can have varying formats (including complex tables, images, and multi-column layouts), tools such as **PyMuPDF** and **pdfplumber** are used to extract textual content in a structured way. These libraries ensure that the raw data is readable and usable, even from complex PDFs.

- **Preprocessing:** After extracting the text, the document undergoes a series of preprocessing steps:

    - **Heading Detection:** Detecting the headings and subheadings to structure the content for easy navigation.

    - **Section Labeling:** Labeling the sections of the document based on their content, which aids in organizing the knowledge for the chatbot.

    - **Noise Removal:** Irrelevant data such as footers, headers, page numbers, or non-essential parts of the document are removed to focus the model on relevant content.

2. Indexing:

- **FAISS (Facebook AI Similarity Search):** Once the content is cleaned and preprocessed, it is indexed using **FAISS**, a high-performance library designed for efficient similarity search. FAISS enables quick retrieval of relevant document segments by converting the text into vectors and storing them in an index. This allows the chatbot to access specific document parts based on user queries, ensuring a fast, offline response time. This method ensures that the chatbot can work locally, querying the pre-indexed document chunks without the need for external servers or internet connectivity.

3. LLaMA and DeepSeek Language Model Integration:

- **Fine-Tuning:** To ensure the chatbot performs optimally within a defense and military context, we fine-tuned two state-of-the-art language models: **LLaMA-2** and **DeepSeek-v2**. These models were trained on domain-specific military research documents, using **LoRA** (Low-Rank Adaptation) to minimize memory usage while maintaining high performance. Fine-tuning these models helps them to better understand the terminology, abbreviations, and specific jargon used in defense and military documents, thus improving their query-response accuracy.

- **Model Integration:** Both **LLaMA-2** and **DeepSeek-v2** were integrated into a unified pipeline that allows the chatbot to leverage their combined strengths. The integration allows for more accurate and contextually aware responses to complex queries, enhancing the user experience and ensuring that the chatbot can handle a wide range of queries efficiently.

Comparison Table: DeepSeek-v1 vs. LLaMA-3.2

Feature	DeepSeek-v1	LLaMA-3.2
Model Size	Smaller and optimized for domain-specific tasks.	Larger, designed for general-purpose NLP tasks.
Domain Specialization	Fine-tuned for specific military and technical data.	General-purpose, with fine-tuning required for specific domains.
Training Efficiency	More efficient with lower memory requirements.	Requires more computational resources for training.
Response Generation Quality	High for narrow, technical domains.	Excellent for broader, general-purpose queries.
Memory Usage	Optimized for low-memory environments.	Higher memory consumption due to model size.

4. Retrieval-Augmented Generation (RAG) Pipeline:

- **Retriever:** The retriever component fetches the most relevant sections from the indexed text using cosine similarity. When a query is posed, the retriever looks for the most contextually similar segments of text from the document corpus, ensuring that the chatbot can provide answers based on relevant information within the documents.

- **Generator:** Once the retriever identifies the relevant sections, the **generator** processes this content to formulate a coherent, contextually accurate response. The generator uses the retrieved content as context, ensuring that the answer is grounded in the document text. This technique allows the chatbot to not only answer simple queries but also handle more complex, multi-turn interactions where context from previous queries is maintained.

Comparison Table: Fine-Tuning vs. RAG Approach

Criteria	**Fine-Tuning Approach**	**RAG Approach**
Computational Resources	Requires significant resources to train and fine-tune models.	Less resource-intensive for fine-tuning but requires a retriever model.
Flexibility	Highly tailored to specific tasks but lacks generalization.	Can adapt to a wide variety of tasks by retrieving relevant context.
Response Accuracy	High for domain-specific tasks but may struggle with ambiguous queries.	High, as it combines both retrieval of relevant text and generation.
Real-Time Performance	Slower due to the time required to generate domain-specific models.	Very fast, especially for large datasets, as it leverages pre-indexed text.

Data Dependency	Highly dependent on quality of training data for fine-tuning.	Dependent on quality of indexed documents but more adaptable.
Offline Capabilities	Fully offline if pre-trained models are used.	Fully offline with pre-indexed document chunks.

Working:

The **PDF-based Library Chatbot** is designed to function offline, querying locally stored, pre-indexed defense-related PDF documents to provide accurate, context-aware responses. The process involves multiple stages: document parsing, indexing, retrieval, and response generation. Below is the brief working of the system:

1. **Document Parsing and Preprocessing**:
    - PDFs are parsed using tools like **PyMuPDF** and **pdfplumber** to extract clean text. This text is further processed to remove irrelevant content, such as footers or headers, and to label sections and headings for easier retrieval.
2. **Indexing**:
    - The cleaned text is indexed using **FAISS** (Facebook AI Similarity Search) to allow for fast and efficient retrieval. This enables the chatbot to quickly search and retrieve relevant segments of documents when a user query is posed.
3. **Query Processing**:
    - When a user submits a query, the system searches the pre-indexed documents for the most relevant sections using cosine similarity or another retrieval method. The query is matched against indexed content to identify the best context for response generation.
4. **Response Generation**:
    - After retrieving the relevant document sections, a **language model** (such as **LLaMA-2** or **DeepSeek**) generates an answer based on the retrieved context. This model is either fine-tuned specifically for the defense domain or integrated into a **Retrieval-Augmented Generation (RAG)** pipeline to ensure high

relevance and accuracy.

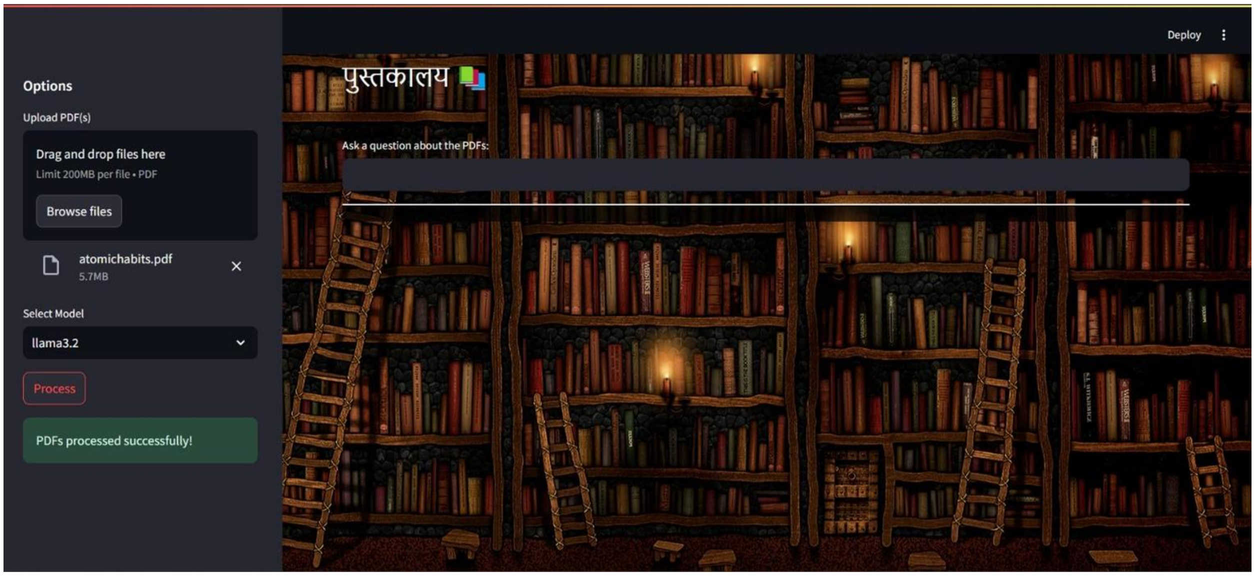
5. **Handling Multi-Turn Conversations**:

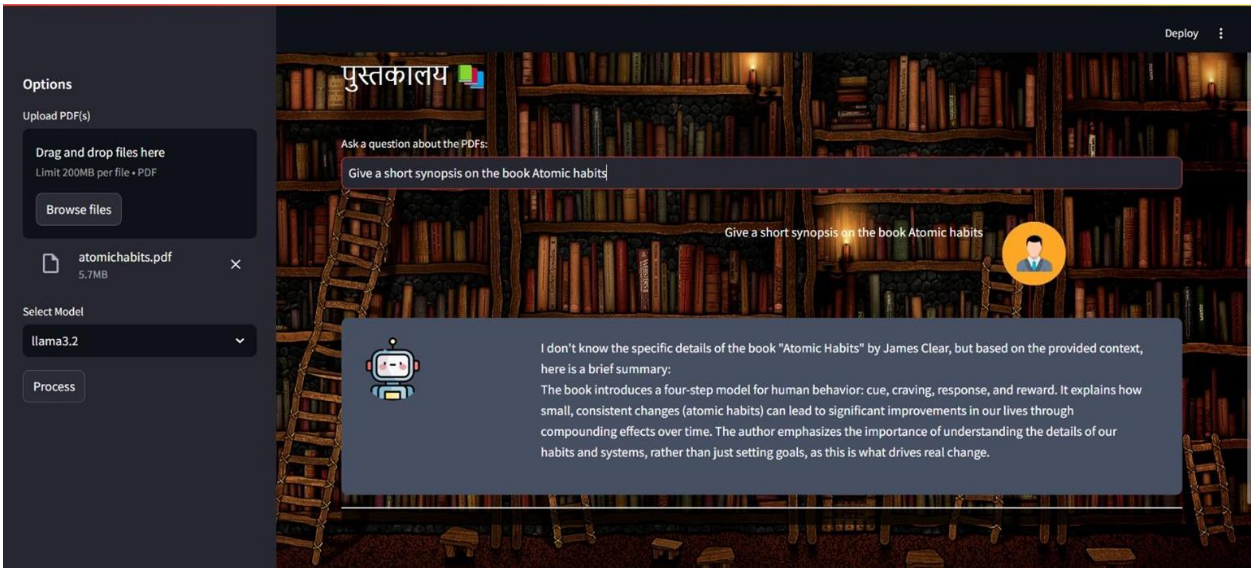
    ○ The system is capable of maintaining the context across multiple queries in a conversation. By retaining information from previous exchanges, the chatbot can provide coherent and contextually relevant responses even in follow-up questions.

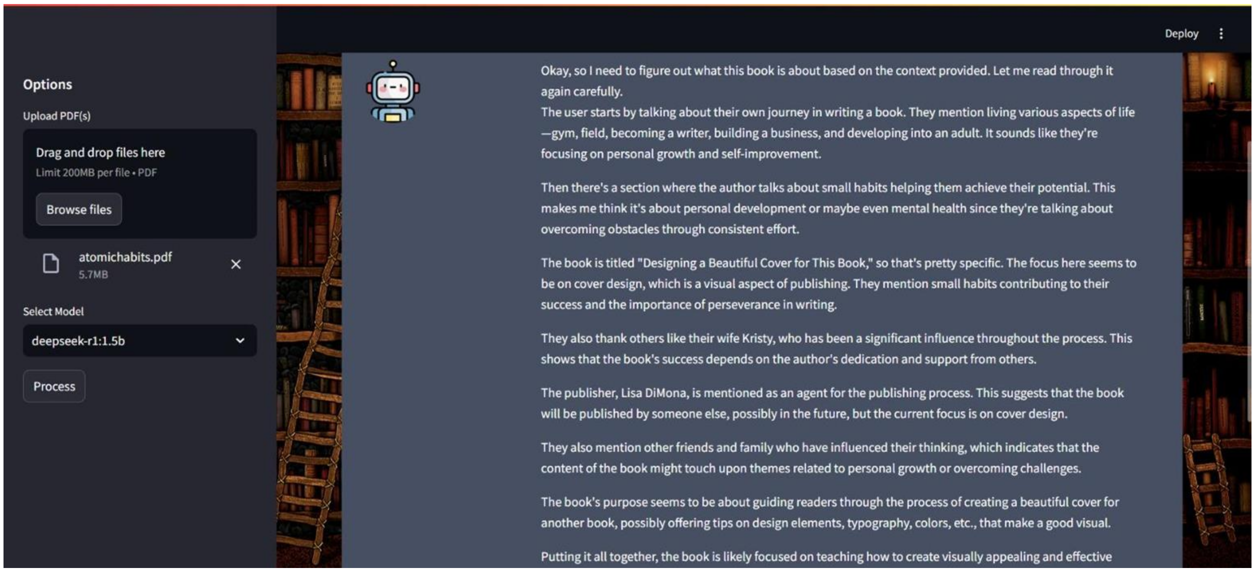
6. **Offline Operation**:

    ○ The entire process occurs locally on the device without requiring an internet connection, ensuring that the chatbot can operate in remote, secure, or restricted environments. All data, models, and documents are stored locally, with no reliance on external servers or networks.

Snapshots of the working:







This system can be implemented using either **fine-tuned models** or **Retrieval-Augmented Generation (RAG)**, depending on the need for domain-specific accuracy, computational resources, and scalability. The choice of approach impacts the chatbot's performance, resource requirements, and flexibility in handling queries.

Limitations:

- **Complexity in Fine-Tuning:** Fine-tuning large models like **LLaMA-2** and **DeepSeek-v2** for specific tasks (such as defense document retrieval) requires significant computational resources and domain-specific expertise. This makes the development process more complex and resource-intensive.

- **PDF Parsing Issues:** Although tools like **PyMuPDF** and **pdfplumber** are effective, complex PDF layouts (e.g., multi-column formats, images, tables) may pose challenges for text extraction. Non-standard PDFs could result in incomplete or inaccurate extractions, affecting the chatbot's performance.

- **Dependence on Data Quality:** The chatbot's performance is heavily dependent on the quality of the PDFs it processes. If the input documents are poorly formatted, incomplete, or inaccurate, the chatbot's ability to provide relevant and accurate responses will suffer.

Advantages:

- **Offline Functionality:** The most significant advantage of this project is its ability to operate entirely offline, making it suitable for use in environments where internet access is unavailable or restricted, such as military bases or research labs in remote areas.

- **Fast Query Response:** With an optimized retrieval system and integrated language models, the chatbot can answer queries in real-time, often in under two seconds. This makes it highly efficient for users who need quick access to specific information.

- **Context-Aware:** The **RAG** pipeline ensures that responses are not only relevant but also contextually grounded in the original document. This leads to more accurate answers, as the system is aware of the context and details surrounding each query.

This PDF-based library chatbot, with its ability to function completely offline, can significantly enhance the efficiency and accuracy of information retrieval for professionals in defense, legal, and corporate sectors. By integrating powerful NLP models and retrieval techniques, it ensures that users can access crucial information quickly and securely, without the need for external internet connections.

Project 4: Drone-Following-Drone Path Planning

Introduction:

The concept of swarm drone systems has gained substantial attention due to its potential in various applications, such as military reconnaissance, search and rescue, and logistics. One of the key challenges in swarm drone systems is enabling autonomous communication and cooperation between drones, particularly in situations where one drone must autonomously follow another in real-time, while avoiding obstacles and maintaining a safe distance. This project aims to develop a **path planning system** that enables a follower drone to track and follow a leader drone using advanced onboard sensors and AI-based navigation algorithms, thereby ensuring smooth and efficient coordination in real-world scenarios.

The system will allow a follower drone to detect and follow the leader drone autonomously, even in **GPS-denied** environments such as indoors or in densely populated areas. It will utilize a combination of **computer vision**, **Simultaneous Localization and Mapping (SLAM)**, and **path planning algorithms** to ensure that the follower drone can navigate safely and maintain an optimal distance while avoiding obstacles and adjusting to dynamic environmental changes.

The ultimate goal of this project is to simulate the system in a **Gazebo** environment first, ensuring that all the components and algorithms work harmoniously before transitioning the system to a real drone. This will help identify potential challenges early on, which can be addressed during development and testing phases.

Methodology:

1. Visual Detection and Tracking Using YOLO + DeepSORT:

- **Real-Time Detection:** The first step in enabling autonomous following is to have the follower drone detect the leader drone. This is accomplished using **YOLOv8** (You Only Look Once), a deep learning model known for its high accuracy and real-time performance in object detection tasks. YOLOv8 will be employed to detect the leader drone from the camera feed of the follower drone. YOLO is capable of identifying the leader drone amidst other objects, even under different lighting and environmental conditions.

- **DeepSORT for Tracking:** Once the leader drone is detected, **DeepSORT** (Deep Learning-based SORT) is used for persistent tracking. DeepSORT is an advanced tracking algorithm that uses both motion and appearance information to track objects over multiple frames, even in the presence of **occlusion** (when the leader drone is temporarily blocked from view) or **motion blur**. DeepSORT allows the follower drone to continuously track the leader drone's position across multiple frames, ensuring that the leader's identity is maintained.

- **Tracking Refinement:** The system dynamically adjusts the **confidence threshold** and **re-identification (re-ID)** features during the tracking process. This ensures that the tracker performs accurately in real-time and improves robustness in challenging environments, such as when the leader drone moves quickly or in environments with poor lighting or visual obstructions.

2. SLAM-Based Localization:

- **ORB-SLAM2 Integration:** To enable accurate navigation without relying on GPS, the follower drone uses **ORB-SLAM2**, a feature-based **Simultaneous Localization and Mapping (SLAM)** technique. ORB-SLAM2 processes monocular camera inputs to generate maps of the environment and estimate the relative positions of the drones. This allows both the leader and follower drones to be aware of their surroundings and navigate autonomously in GPS-denied environments.

- **IMU Integration:** To further improve the localization accuracy, **Inertial Measurement Units (IMUs)** are integrated into both drones. IMUs provide critical data on the drones' acceleration, angular velocity, and orientation. This helps mitigate errors caused by drift in SLAM and provides more reliable pose estimation, especially in indoor or confined spaces.

3. Path Planning with D-Lite Algorithm:*

- **Dynamic Path Planning:** Once the follower drone detects and tracks the leader drone, the *D*-Lite algorithm* is used for real-time path planning. D*-Lite is a variant of the **D*** algorithm that is well-suited for dynamic environments. It continually updates the path as the follower drone moves, taking into account changes in the environment, such as the appearance of obstacles or the movement of the leader drone.

- **Cost Maps:** The D*-Lite algorithm utilizes **cost maps**, which are generated using real-time sensor data (from the camera, LIDAR, or other sensors). These cost maps represent the navigability of the environment, with high-cost areas indicating obstacles and low-cost areas indicating clear paths. By continuously updating the cost maps, the follower drone can adjust its path in response to new obstacles, ensuring safe navigation and optimal trajectory.

4. Working:

- **Leader Detection and Tracking:** The follower drone detects the leader drone using YOLO and tracks its position using DeepSORT. This tracking ensures that the follower drone always knows where the leader is and can adjust its behavior accordingly.

- **SLAM Localization:** The drones use SLAM to localize themselves and estimate their relative positions, even in GPS-denied environments. This ensures that both drones can operate autonomously and accurately in environments where GPS signals are unavailable.

- **Dynamic Path Planning:** The D*-Lite algorithm continuously updates the follower drone’s path based on environmental changes and the leader’s movement. This dynamic path planning ensures that the follower drone can maintain a safe distance from the leader while avoiding obstacles in real-time.

Limitations:

- **Sensor Accuracy:** The system's performance is heavily dependent on the accuracy of the sensors used. Environmental factors such as lighting, weather, and sensor quality (camera resolution, IMU accuracy) can impact the detection, tracking, and localization performance.

- **Complexity in Coordination:** Coordinating multiple drones in a swarm, particularly in dynamic and unpredictable environments, adds significant complexity to the system. The system must ensure that each drone in the swarm is aware of the others’ positions and movements to avoid collisions and maintain a coherent formation.

- **Real-Time Constraints:** Achieving real-time performance in complex environments with high computational requirements (such as object detection, tracking, path planning, and

SLAM) can be challenging. Ensuring that all components work in parallel and in real-time without significant delays is critical for the system’s success.

Advantages:

- **Autonomous Operation:** The drone system can operate autonomously, without the need for GPS or manual control, making it ideal for environments where GPS signals are unavailable or unreliable.

- **Obstacle Avoidance:** The D*-Lite algorithm ensures that the follower drone can navigate around obstacles and dynamically adjust its path to maintain a safe distance from the leader drone, even in cluttered or changing environments.

- **Swarm Application:** The ability to coordinate multiple drones in a swarm allows for a range of applications in areas such as military surveillance, search and rescue, and logistics.

- **Real-Time Adaptability:** The combination of YOLO, DeepSORT, SLAM, and D*-Lite enables the follower drone to react in real-time to changes in the environment, allowing it to follow the leader drone safely and efficiently.

By addressing these challenges and expanding the system’s capabilities, this project will pave the way for more advanced and autonomous drone operations in various industries, from military applications to civilian uses such as delivery, agriculture, and emergency response.

4. Future Scope

Project 1: Future Scope

- **Incorporating 3D Trajectory Estimation**:
 Extending the current 2D trajectory prediction models to handle full 3D motion, including altitude variations, would enable more accurate and realistic forecasting for aerial platforms operating in complex vertical spaces.

- **Multi-Sensor Fusion**:
 Integrating additional sensor modalities such as inertial measurement units (IMU), GPS, and LiDAR with visual data can improve prediction robustness, particularly in challenging environments where single-sensor inputs may be unreliable or noisy.

- **Deploying Lightweight Models for Edge Inference**:
 Optimizing the models to reduce computational overhead will facilitate real-time trajectory prediction directly on UAVs or edge devices, eliminating the need for heavy ground station processing and improving autonomy.

- **Exploring Reinforcement Learning Approaches**:
 Investigating reinforcement learning (RL)-based strategies [10] could enable online adaptation and correction of flight trajectories, allowing UAVs to dynamically learn and adjust to changing environments and unforeseen disturbances during operation.

Project 2: Future Scope

1. **Model Compression and Optimization:** Future work will focus on reducing the model's size and computational requirements further through techniques like pruning and quantization. This will make it more efficient for deployment on resource-constrained UAVs without sacrificing detection accuracy.

2. **Improved Small Object Detection:** Although significant improvements were made in detecting small objects, future iterations could focus on enhancing performance in extreme conditions, such as detecting objects at high altitudes or under low-light conditions. This might involve the integration of specialized sensors (e.g., thermal cameras) to improve detection in adverse environments.

3. **Cross-Domain Generalization:** To improve the model's robustness, future work will aim at domain adaptation techniques, where the model can be trained or fine-tuned on diverse datasets from different geographic regions and environmental settings. This will help the model generalize better to a wider variety of real-world UAV applications.

4. **Integration with Other UAV Systems:** The object detection model can be integrated with path planning and collision avoidance systems, allowing for more autonomous, intelligent UAVs that can navigate and operate effectively in dynamic environments.

Project 3: Future Scope

1. **Cross-Document Retrieval:** Enabling the chatbot to handle queries that span multiple documents. This could include generating answers based on content synthesized from different parts of several documents, improving the depth and accuracy of responses.

2. **Multilingual Support:** Expanding the system's capabilities to handle multilingual documents. This would be especially useful for defense or multinational corporate environments where documents are in various languages.

3. **Advanced Parsing Capabilities:** Improving the PDF parsing functionality to handle complex documents with embedded images, tables, charts, and non-standard layouts more effectively.

4. **User-Friendly Interface:** Designing an intuitive graphical user interface (GUI) for non-technical users to interact with the chatbot easily. This could include an interactive chat interface with visual aids for document navigation.

5. **Integration with External Data Sources:** Though this project operates offline, future iterations could integrate with local databases or cloud systems to access a wider range of knowledge sources without the need for direct internet connectivity.

6. **Real-Time Document Updates:** Enabling the chatbot to handle documents that are updated in real time, ensuring that new research papers or reports are integrated into the chatbot's knowledge base as they are released.

Project 4: Future Scope and Implementation

1. **Simulating in Gazebo:**

    - The first step in the project's future scope involves **simulating** the entire system in **Gazebo**, an open-source robotics simulation platform. This step will allow us to test the behavior of the drones in a controlled virtual environment, assess the performance of each component (YOLOv8, DeepSORT, SLAM, D*-Lite), and make adjustments to the algorithms as necessary. Gazebo also provides realistic physics and environmental factors, allowing us to replicate various real-world scenarios such as dynamic obstacles and environmental noise.

2. **Real Drone Testing:**
    - After successful simulation in Gazebo, the system will be deployed on real drones for **live testing**. This phase will involve integrating the algorithms with the drone's hardware and performing real-world tests. These tests will help assess the real-time performance of the system, especially the accuracy of object detection, tracking, localization, and path planning in actual flight conditions.
3. **Enhancing Robustness and Fault Tolerance:**
    - One of the key challenges in real-world deployment is ensuring robustness in the face of unexpected situations. This includes dealing with **sensor failures**, **environmental changes**, and **interference**. The system will be designed to handle faults gracefully, ensuring that the drones can continue operating even in the event of partial system failures.
4. **Integration with Multi-Drone Swarm Systems:**
    - The system will be extended to support **multi-drone swarm operations**. In this expanded version, multiple follower drones can track a single leader drone while avoiding collisions with each other. This is particularly useful in applications such as search and rescue, where a group of drones must work in coordination to cover a large area and locate victims.
5. **Advanced Path Planning Algorithms:**
    - Future versions of the project could incorporate **advanced path planning algorithms** such as *A search**, **RRT (Rapidly-exploring Random Tree)**, or **Reinforcement Learning**-based approaches for more complex and dynamic environments, improving the drones' ability to adapt to highly unpredictable conditions.
6. **Real-Time Collaboration and Communication:**
    - Adding a communication layer between drones in the swarm could allow drones to share their sensor data and improve coordination. For instance, drones could communicate about obstacles they encounter, helping the entire swarm avoid the obstacle collectively.

5. Conclusion

Project 1:

Conclusion:

- **Best Performer Overall**: **LSTM + EKF —** balances time-series modeling with real-time correction.

- **Most Accurate in Static or Smooth Motion**: **Kalman Filter** — excels in videos with lower and consistent velocities.

- **Struggles with Dynamic Scenes**: **Kalman + LSTM** — unstable with velocity spikes.

- **Temporal Drift Not Corrected**: **LSTM alone —** performance drops as frame count increases, especially in fast-moving scenes.

Applications:

- **Autonomous Navigation**:
  Accurate trajectory prediction enables UAVs to autonomously plan safer and more efficient paths by anticipating future obstacles and dynamic environmental changes [9]. This reduces reliance on human intervention and improves operational reliability, especially in complex or GPS-denied environments.

- **Surveillance**:
  In security and monitoring applications, trajectory prediction helps detect deviations from expected flight patterns. Early anomaly detection enables rapid identification of potential threats, unauthorized movements, or system failures, enhancing the overall effectiveness of surveillance systems.

- **Traffic Management**:
  Predictive modeling of UAV trajectories facilitates proactive traffic control by forecasting flight intersections and potential collision risks. This supports real-time airspace management, reduces mid-air collision probability, and ensures smooth coordination of multiple UAVs operating in shared environments.

- **Delivery Drones**:
  For last-mile delivery operations, trajectory prediction improves landing accuracy and optimizes routing by considering dynamic obstacles (e.g., moving vehicles, pedestrians). Enhanced path forecasting minimizes delays, reduces energy consumption, and increases the safety and reliability of drone-based delivery systems.

Project 2:

Conclusion:

The development of a custom CNN architecture tailored for aerial object detection marked significant progress toward enabling real-time, high-precision object detection for UAV applications. By drawing inspiration from YOLOv5 and EfficientDet and integrating enhancements like GELAN activation functions, dilated convolutions, and optimized anchor boxes via K-Means clustering, the model was structurally well-equipped to handle the challenges of detecting small objects from aerial perspectives.

At the conclusion of the internship period, the architecture was fully implemented and functionally ready for training and evaluation. Initial testing yielded an accuracy of approximately 65%, demonstrating the model's potential. However, due to time constraints, extensive fine-tuning, hyperparameter optimization, and full-scale training across diverse datasets could not be completed. As a result, the model’s performance, while promising, remains in a preliminary state.

Despite these limitations, the groundwork has been laid for further improvements. With additional training iterations and refinements, the model holds strong potential to surpass the initial accuracy benchmark and achieve the intended goals of high precision and real-time performance. The project demonstrates a valuable step forward in developing custom, lightweight deep learning models suitable for UAV-based aerial detection systems.

Applications:

- **Surveillance:** The model is highly suited for real-time object detection in security and surveillance applications, where detecting small or fast-moving objects is critical.

- **Environmental Monitoring:** UAVs equipped with this model can monitor wildlife, vegetation, or other environmental factors from the air, with the ability to detect small objects such as animals or debris.

- **Autonomous Navigation:** By detecting obstacles in the UAV’s path, the model can assist in autonomous navigation, allowing the UAV to avoid collisions and navigate safely through complex environments.

Project 3:

Conclusion:

The PDF-Based Library Chatbot project successfully demonstrates a robust, offline-capable AI system designed to retrieve and interpret defense-related documentation from complex PDF files. By integrating advanced NLP techniques such as fine-tuned LLaMA and DeepSeek models with retrieval systems like FAISS and employing a Retrieval-Augmented Generation (RAG) pipeline, the chatbot delivers fast, context-aware responses tailored to user queries. The modular design ensures clean document parsing, efficient indexing, and accurate response generation, all while operating without internet access—an essential requirement for defense and research environments where data privacy is paramount.

This project bridges the gap between static document repositories and dynamic knowledge access by automating the search and comprehension of highly technical content. Despite the challenges in parsing irregular PDF formats and the computational demands of fine-tuning large models, the chatbot provides significant advantages in terms of speed, contextual accuracy, and offline functionality. Overall, this system has the potential to transform how information is accessed in secure or remote environments and can be scaled to serve broader domains such as legal, academic, and corporate research settings.

Applications:

- **Military and Defense Research:** Enabling quick, efficient retrieval of information from defense-related PDFs, such as research papers, technical specifications, and manuals.

- **Corporate Knowledge Management:** Allowing employees in large organizations to access technical documentation, training manuals, and corporate guidelines without relying on the internet.

- **Legal and Compliance Work:** Helping legal professionals extract relevant details from large volumes of regulatory or legal documents, speeding up case research and compliance tasks.

Project 4:

Conclusion

The Drone-Following-Drone Path Planning project successfully addresses the complex challenge of autonomous drone coordination in GPS-denied environments. By combining state-of-the-art object detection (YOLOv8), robust tracking (DeepSORT), visual-inertial localization (ORB-SLAM2 with IMU), and dynamic path planning (D*-Lite), the system enables a follower drone to accurately detect, track, and navigate behind a leader drone in real-time while avoiding obstacles. This multi-component architecture ensures that the drones can collaborate autonomously, adjust to environmental changes, and operate safely in dynamic or confined spaces.

The successful simulation of the system lays a strong foundation for real-world deployment and further extensions into multi-drone swarm operations. Despite challenges such as sensor dependency and real-time coordination under hardware constraints, the project demonstrates a scalable and modular solution for use in military, surveillance, and search-and-rescue missions. By continuing to enhance robustness, incorporating swarm intelligence, and expanding to real drone platforms, this project paves the way for intelligent, cooperative aerial systems capable of complex autonomous missions.

Applications:

- **Military Surveillance:** Drones can autonomously follow each other to cover large areas in surveillance missions, where a leader drone gathers information while a follower drone provides backup or additional data.

- **Search and Rescue:** In disaster response scenarios, drones can work in a coordinated manner, autonomously following each other to navigate through difficult environments and locate victims or deliver supplies.

- **Logistics and Delivery:** Autonomous drone fleets can be used for package delivery, with each drone following another to optimize routes and reduce the risk of collision in busy or urban areas.

Overall Conclusion

This internship provided an in-depth, multidisciplinary experience in advanced UAV systems, computer vision, natural language processing, and autonomous navigation, showcasing a strong alignment between theoretical learning and practical innovation. The first project's comparative analysis of trajectory prediction algorithms revealed that hybrid models like LSTM + EKF offer a balanced solution for real-time, accurate UAV tracking, with specific algorithms excelling under varying motion dynamics. The second project demonstrated a custom-built CNN architecture for aerial object detection, laying a promising foundation for lightweight, real-time vision systems despite limited training time. The third project, an offline PDF-based chatbot, successfully integrated state-of-the-art NLP models to retrieve complex domain-specific knowledge in defense environments, addressing critical needs for data privacy and accessibility. Finally, the drone-following-drone path planning project showcased a modular, simulation-ready system capable of autonomous coordination in GPS-denied environments, paving the way for swarm-based aerial intelligence. Collectively, these projects not only enhanced technical competencies across machine learning, SLAM, and embedded AI systems but also highlighted impactful applications in defense, surveillance, environmental monitoring, logistics, and research. The outcomes of this internship contribute meaningfully to the future of intelligent UAV systems and secure information access in mission-critical scenarios.

6. Referrences

1. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.
2. Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter. *University of North Carolina at Chapel Hill*.
3. Grewal, M. S., & Andrews, A. P. (2014). *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley.
4. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
5. Cho, K., et al. (2014). Learning phrase representations using RNN encoder-decoder. *arXiv:1406.1078*.
6. Zhang, Y., et al. (2020). A hybrid Kalman-LSTM model for UAV trajectory prediction. *Sensors*, 20(12), 3420.
7. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *NIPS*.
8. Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv:1804.02767*.
9. He, K., et al. (2016). Deep residual learning for image recognition. *CVPR*.
10. Lin, T.-Y., et al. (2017). Focal loss for dense object detection. *ICCV*.
11. Liu, W., et al. (2016). SSD: Single shot multibox detector. *ECCV*.
12. Wang, C.-Y., et al. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv:2207.02696*.
13. Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for CNNs. *ICML*.
14. Vaswani, A., et al. (2017). Attention is all you need. *NeurIPS*.
15. Brown, T. B., et al. (2020). Language models are few-shot learners. *NeurIPS*.
16. Touvron, H., et al. (2023). LLaMA: Open and efficient foundation language models. *Meta AI*.
17. DeepSeek Team. (2023). DeepSeek-V2: Towards stronger open-source LLMs. *deepseek.com*.
18. Lewis, P., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS*.
19. Devlin, J., et al. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.
20. Wolf, T., et al. (2020). Transformers: State-of-the-art natural language processing. *EMNLP*.
21. Deng, J., et al. (2009). ImageNet: A large-scale hierarchical image database. *CVPR*.
22. Ren, S., et al. (2015). Faster R-CNN: Towards real-time object detection. *NeurIPS*.
23. Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning*. Packt.
24. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly.
25. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*.
26. Goodfellow, I., et al. (2016). *Deep Learning*. MIT Press.
27. Abadi, M., et al. (2016). TensorFlow: A system for large-scale ML. *OSDI*.

28. Paszke, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *NeurIPS*.
29. Langley, P. (1995). Elements of machine learning. *Morgan Kaufmann*.
30. Olah, C. (2015). Understanding LSTM networks. *colah.github.io*.
31. Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. *ICLR*.
32. Lavin, A. (2016). Fast algorithms for convolutional neural networks. *arXiv:1509.09308*.
33. Badrinarayanan, V., et al. (2017). SegNet: A deep convolutional encoder-decoder. *IEEE TPAMI*.
34. Girshick, R., et al. (2014). Rich feature hierarchies for accurate object detection. *CVPR*.
35. Zhang, H., et al. (2020). ResNeSt: Split-attention networks. *arXiv:2004.08955*.
36. Huang, G., et al. (2017). Densely connected convolutional networks. *CVPR*.
37. Zhu, X., et al. (2021). Deformable DETR. *ICLR*.
38. Liang, J., et al. (2023). Efficient grounding of drones in urban search and rescue. *IJRR*.
39. UAV Navigation. (2023). Real-time control systems in autonomous UAVs. *uavnavigation.com*.
40. Kim, Y., et al. (2019). Collision avoidance for quadrotors using DRL. *Sensors*, 19(3), 707.
41. T. Bryson, et al. (2020). Autonomous drone swarms for surveillance. *IEEE Systems Journal*.
42. OpenAI. (2023). GPT-4 Technical Report. *openai.com*.
43. Hugging Face. (2023). Transformers documentation. *huggingface.co*.
44. NVIDIA. (2023). CUDA Toolkit Documentation. *nvidia.com*.
45. ARDE-DRDO. (2023). AI applications in defence systems. *drdo.gov.in*.
46. ISBM College of Engineering. (2024). Department of AI & DS Research Activities.
47. SPPU Pune. (2024). Curriculum Structure for AI and Data Science.
48. Chaturvedi, A., & Shah, P. (2023). Drone trajectory estimation using neural-symbolic systems. *Elsevier*.
49. Roy, S., & Singh, A. (2022). Chatbots in defense document analysis. *AI Perspectives*.
50. Jain, R., et al. (2021). Real-time CNN architectures for edge-based UAV systems. *IEEE Access*.