Project Title	**Motion Tracking in Drone Videos using FAST and ORB**
Skills take away From This Project	**Python, Computer Vision, OpenCV, Streamlit,**
Domain	**Security / Wildlife Monitoring / Sports**

Problem Statement:

Drone videos are increasingly used in various applications, such as surveillance, traffic monitoring, and wildlife observation. These videos often capture dynamic scenes with multiple moving objects, such as vehicles, people, or animals. Manually tracking these objects is time-consuming, labor-intensive, and prone to human error. Additionally, the complexity of drone footage—such as varying lighting conditions, occlusions, and fast-moving objects—makes it challenging to achieve accurate and real-time tracking.

This project aims to **automate the tracking of moving objects in drone videos** using computer vision techniques. By leveraging **FAST (Features from Accelerated Segment Test)** for keypoint detection and **ORB (Oriented FAST and Rotated BRIEF)** for descriptor matching, the system will detect and track objects in real-time. The goal is to provide a robust, efficient, and scalable solution that can handle the challenges of drone footage, enabling applications such as surveillance, traffic analysis, and wildlife monitoring. The system will also visualize motion tracks, making it easier for users to analyze object movements and patterns.

Business Use Cases:

1. Surveillance and Security

- **Use Case**: Monitor moving objects (e.g., vehicles, people) in restricted or sensitive areas.
- **Benefit**: Enhances security by providing real-time tracking of suspicious activities.

2. Traffic Monitoring and Management

- **Use Case**: Track vehicles in urban or highway environments to analyze traffic flow and detect congestion.
- **Benefit**: Improves traffic management and reduces congestion by providing real-time data.

3. Wildlife Monitoring

- **Use Case**: Observe and track animal movements in natural habitats or conservation areas.
- **Benefit**: Supports wildlife research and conservation efforts by providing detailed movement data.

4. Sports and Event Coverage

- **Use Case**: Track players, athletes, or participants during sports events, festivals, or large gatherings.
- **Benefit**: Enhances event coverage and provides valuable insights for performance analysis.

5. Disaster Management and Search & Rescue

- **Use Case**: Track moving objects (e.g., people, vehicles) during disaster response or search and rescue operations.
- **Benefit**: Improves efficiency in locating and rescuing individuals in critical situations.

6. Agriculture and Farming

- **Use Case**: Monitor the movement of livestock or farm equipment in large agricultural fields.
- **Benefit**: Enhances farm management and productivity by providing real-time tracking data

7. Construction and Infrastructure Monitoring

- **Use Case**: Track the movement of machinery, vehicles, or workers on construction sites.
- **Benefit**: Improves safety and efficiency in construction projects.

8. Retail and Crowd Analysis

- **Use Case**: Track customer movement in retail stores or crowded areas for behavior analysis.

- **Benefit**: Provides insights for optimizing store layouts and improving customer experience.

9. Autonomous Drone Applications

- **Use Case**: Enable drones to track and follow moving objects autonomously (e.g., delivery drones tracking a moving vehicle).
- **Benefit**: Enhances the capabilities of autonomous drones for delivery, surveillance, and more.

10. Military and Defense

- **Use Case**: Track enemy movements or vehicles in military operations using drone footage.
- **Benefit**: Provides real-time intelligence for strategic decision-making.

Approach:

1. Data Collection and Preprocessing

- **Step**: Collect drone-shot videos with moving objects (e.g., vehicles, people, animals).
- **Action**:
  - Extract frames from the video at a consistent frame rate.
  - Convert frames to grayscale for efficient keypoint detection.

2. Keypoint Detection

- **Step**: Detect keypoints in each frame using the **FAST (Features from Accelerated Segment Test)**algorithm.
- **Action**:
  - Identify corners and distinctive features in the frames.
  - Use FAST for its speed and efficiency in real-time applications.

3. Descriptor Extraction

- **Step**: Compute descriptors for the detected keypoints using the **ORB (Oriented FAST and Rotated BRIEF)** algorithm.
- **Action**:
  - Generate binary descriptors for each keypoint.
  - Use ORB for its robustness to rotation and scale changes.

4. Keypoint Matching

- **Step**: Match keypoints between consecutive frames to track object movement.

- **Action**:
    - Use **Brute-Force Matching** or **FLANN (Fast Library for Approximate Nearest Neighbors)** to find corresponding keypoints.
    - Filter matches using a distance threshold to ensure accuracy.

5. Motion Tracking

- **Step**: Track the movement of objects using **Optical Flow** (e.g., Lucas-Kanade method).
- **Action**:
    - Estimate the motion vectors of keypoints between frames.
    - Use optical flow to handle small movements and improve tracking accuracy.

6. Visualization

- **Step**: Visualize the motion tracks on the video frames.
- **Action**:
    - Draw lines or trajectories connecting the tracked keypoints.
    - Highlight the path of moving objects for easy interpretation.

7. Streamlit Integration

- **Step**: Build a user-friendly web application for uploading videos and viewing results.
- **Action**:
    - Use **Streamlit** to create an interactive interface.
    - Allow users to upload drone videos, view motion tracks, and download processed videos.

8. Deployment in Streamlit

- **Step**: Deploy the motion tracking system as a web application using Streamlit.
- **Action**:
    - **Create a Streamlit App**:
        - Build a Python script (app.py) to handle video uploads, processing, and visualization.
    - **Upload Video**:
        - Use st.file_uploader to allow users to upload drone videos.
    - **Process Video**:
        - Apply the FAST, ORB, and Optical Flow algorithms to track moving objects.
    - **Display Results**:

        - Show the original video and the processed video with motion tracks side by side.
    - **Download Processed Video**:
        - Provide an option to download the processed video with motion tracks.
    - **Deploy the App**:
        - Use Streamlit Sharing, Heroku, or any cloud platform to deploy the app.

9. Evaluation and Optimization

- **Step**: Evaluate the system's performance and optimize for real-time use.
- **Action**:
    - Measure tracking accuracy, processing speed, and robustness.
    - Optimize algorithms for faster and more accurate tracking.

Streamlit Integration:

To ensure user-friendly access to the model, a Streamlit-based web application is developed with the following features:

The project will include a **Streamlit-based web application** to make the system accessible and user-friendly. The application will allow users to:

- Upload images for Segmentation of MRI.
- View the original input with detected objects highlighted.
- See segmented result
- Download the results for further analysis.

Exploratory Data Analysis (EDA):

Before training the model, thorough analysis is conducted on the dataset:

1. Video Analysis

- **Objective**: Understand the characteristics of the drone videos.
- **Steps**:
    - Analyze video resolution, frame rate, and duration.
    - Check for variations in lighting, background, and object movement.
- **Insights**:
    - Determine if the videos are suitable for real-time processing.
    - Identify potential challenges (e.g., low lighting, fast-moving objects).

2. Keypoint Distribution

- **Objective**: Study the distribution of keypoints detected by FAST.
- **Steps**:
  - Detect keypoints in a sample of frames using FAST.
  - Plot the spatial distribution of keypoints.
- **Insights**:
  - Identify regions with high keypoint density (e.g., moving objects, textured areas).
  - Ensure keypoints are evenly distributed across frames.

3. Object Size and Speed

- **Objective**: Analyze the size and speed of moving objects.
- **Steps**:
  - Measure the size (in pixels) of objects in the frames.
  - Estimate the speed of objects based on their movement between frames.
- **Insights**:
  - Determine if the system can handle small or fast-moving objects.
  - Identify objects that may be difficult to track (e.g., too small or too fast).

4. Background Complexity

- **Objective**: Evaluate the complexity of the background in drone videos.
- **Steps**:
  - Analyze the texture and motion in the background (e.g., trees, buildings).
  - Check for occlusions or distractions (e.g., shadows, reflections).
- **Insights**:
  - Identify potential challenges for keypoint detection and tracking.
  - Determine if background subtraction or filtering is needed.

5. Class Distribution

- **Objective**: Analyze the distribution of moving objects in the videos.
- **Steps**:
  - Count the number of objects in each frame.
  - Categorize objects by type (e.g., vehicles, people, animals).
- **Insights**:
  - Ensure the dataset includes a variety of objects for robust tracking.
  - Identify any class imbalance (e.g., too many vehicles, few animals).

6. Lighting Conditions

- **Objective**: Assess the impact of lighting on video quality.
- **Steps**:
  - Analyze brightness and contrast levels in the frames.
  - Check for variations in lighting (e.g., shadows, glare).
- **Insights**:
  - Determine if preprocessing (e.g., histogram equalization) is needed.
  - Identify frames with poor lighting that may affect tracking accuracy.

7. Motion Patterns

- **Objective**: Study the movement patterns of objects.
- **Steps**:
  - Track the trajectory of objects across frames.
  - Analyze the direction and speed of movement.
- **Insights**:
  - Identify common motion patterns (e.g., linear, circular).
  - Detect anomalies or unexpected movements.

8. Noise and Artifacts

- **Objective**: Identify noise or artifacts in the videos.
- **Steps**:
  - Check for compression artifacts, blur, or distortion.
  - Analyze the impact of noise on keypoint detection.
- **Insights**:
  - Determine if denoising or filtering is required.
  - Ensure the system can handle low-quality videos.

Questions to be Answered (OpenCV Computer Vision):

1. How can businesses automate the tracking of moving objects in drone videos for surveillance and security?
2. Can the system improve traffic monitoring and management by tracking vehicles in real-time?
3. How can retailers use motion tracking to analyze customer movement and optimize store layouts?
4. Can the system enhance construction site safety by tracking machinery and workers?
5. How can the system support autonomous drone applications, such as delivery or inspection?

6. How can the system help monitor wildlife movements for conservation and research purposes?
7. Can the system assist in disaster management by tracking people or vehicles during search and rescue operations?
8. How can the system improve public safety by monitoring crowded areas or events?
9. Can the system be used to analyze traffic patterns and reduce congestion in urban areas?
10. How can the system support law enforcement in tracking suspicious activities or vehicles?
11. How can the system be used in sports analytics to track players or athletes during games?
12. Can the system enhance agricultural monitoring by tracking livestock or farm equipment?
13. How can the system support military and defense operations by tracking enemy movements?
14. Can the system be used in filmmaking or media production to track moving objects for special effects?
15. How can the system assist in environmental monitoring, such as tracking animal migrations or deforestation activities?

Project Evaluation Metrics:

Object tracking in **drone videos** using **FAST (Features from Accelerated Segment Test) and ORB (Oriented FAST and Rotated BRIEF)** requires a combination of **accuracy, robustness, and real-time efficiency**. Below are the key evaluation metrics for assessing the performance of the tracking system.

1. Accuracy Metrics

These metrics measure how well the tracker identifies and follows the object over time.

Intersection over Union (IoU):

- Measures how much the predicted bounding box overlaps with the ground truth.
- Higher IoU values indicate better tracking accuracy.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Data:

https://drive.google.com/file/d/1UiU-GxPrQQgouLaBVd7R9hOtMwka11gM/view?usp=sharing

Dataset Explanation:

The dataset consists of **drone-shot videos** capturing dynamic scenes with moving objects such as vehicles, people, or animals. The goal is to use this dataset to develop and evaluate a motion tracking system that can detect and track these objects in real-time.

Preprocessing Steps

1. **Frame Extraction**:
    a. Extract frames from the videos at a consistent frame rate.
2. **Grayscale Conversion**:
    a. Convert frames to grayscale for efficient keypoint detection.
3. **Resizing**:
    a. Resize frames to a standard resolution (e.g., 640x480) for consistent processing.
4. **Annotation (Optional)**:
    a. Manually annotate ground truth tracks for evaluation (e.g., bounding boxes or motion paths).

Challenges in the Dataset

1. **Dynamic Backgrounds**:
    a. Moving trees, water, or clouds can interfere with object tracking.
2. **Occlusions**:
    a. Objects may be temporarily hidden by other objects or structures.
3. **Lighting Variations**:
    a. Changes in lighting (e.g., shadows, glare) can affect tracking accuracy.
4. **Small or Fast-Moving Objects**:
    a. Small objects or objects moving at high speeds may be difficult to track.

Evaluation Using the Dataset

1. **Tracking Accuracy**:
    a. Compare predicted motion tracks with ground truth annotations (if available).
2. **Robustness**:
    a. Test the system's performance under challenging conditions (e.g., occlusions, lighting changes).

3. **Real-Time Performance**:
    a. Measure the system's processing speed (FPS) on the dataset.

Results:

1. Tracking Accuracy

- **Dice Coefficient**: Achieved an average Dice score of **0.85**, indicating strong overlap between predicted and ground truth motion tracks.
- **IoU (Intersection over Union)**: Achieved an average IoU of **0.78**, demonstrating precise boundary detection for tracked objects.
- **Precision**: **88%** of predicted tracks were correct, minimizing false positives.
- **Recall**: **83%** of actual objects were successfully tracked, ensuring minimal false negatives.

2. Real-Time Performance

- **Processing Speed**: The system processes videos at **15–20 FPS** on a standard GPU, making it suitable for real-time applications.
- **Latency**: The average delay between input and output is **<100 ms**, ensuring responsive performance.

3. Robustness

- **Lighting Variations**: The system maintains a Dice score of **0.80+** under varying lighting conditions (e.g., shadows, glare).
- **Occlusions**: Achieves a tracking accuracy of **75%+** even when objects are partially occluded.
- **Fast-Moving Objects**: Successfully tracks objects moving at speeds of up to **30 km/h** in the video.

4. Visualization

- **Motion Tracks**: Overlaid motion tracks on the video frames, clearly showing the path of moving objects.
- **Heatmaps**: Generated heatmaps to highlight regions with high object activity.

Technical Tags:

1. **Python**: Primary programming language for implementation.

2. **OpenCV**: For image processing, keypoint detection, and motion tracking.
3. **FAST (Features from Accelerated Segment Test)**: For efficient keypoint detection.
4. **ORB (Oriented FAST and Rotated BRIEF)**: For robust descriptor extraction and matching.
5. **Streamlit**: For building the user-friendly web application.
6. **Optical Flow**: For tracking object motion between frames (e.g., Lucas-Kanade method).
7. **Keypoint Detection**: Identifying distinctive features in video frames.
8. **Descriptor Matching**: Matching keypoints across frames for object tracking.
9. **Video Processing**: Handling and processing video frames in real-time.
10. **Feature Extraction**: Extracting meaningful features from video frames.
11. **Real-Time Tracking**: Ensuring low-latency processing for live video feeds.
12. **Robustness**: Handling challenges like occlusions, lighting changes, and fast motion.

Project Deliverables:

1. Trained Motion Tracking System

- **Description**: A Python-based system using **FAST** for keypoint detection and **ORB** for descriptor matching to track moving objects in drone videos.
- **Format**: Python scripts and model files (e.g., .py, .pt).
- **Purpose**: Core functionality for detecting and tracking objects.
-

2. Streamlit Web Application

- **Description**: A user-friendly web interface for uploading drone videos, visualizing motion tracks, and downloading results.
- **Features**:
  - Upload drone videos.
  - Display original video and processed video with motion tracks.
  - Download processed videos and performance reports.
- **Purpose**: Makes the system accessible to non-technical users.
-

3. Documentation

- **Description**: A detailed README file and project report explaining the setup, usage, and results.
- **Contents**:
  - Project overview and objectives.
  - Dataset description and preprocessing steps.

    - Model architecture and training details.
    - Instructions for running the code and Streamlit app.
    - Results and performance metrics.
- **Purpose**: Ensures easy understanding and replication of the project.

4. Sample Outputs

- **Description**: A collection of sample drone videos with motion tracks overlaid.
- **Format**: Video files (e.g., MP4) or a PDF report with screenshots.
- **Purpose**: Demonstrates the system's capabilities and accuracy.

Timeline:

The project must be completed and submitted **within 10 days from the assigned Date.**

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Tuesday, Thursday, Saturday (5:00PM to 7:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.
Note: This form will Open on Saturday and Sunday Only on Every Week

Timing: Monday-Saturday (11:30PM to 12:30PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA