Software Requirements Specification

for

Aircraft Classification System

Version 1.0

Prepared by

Group Name: Classifiers

Ritika Buchupalli	Se22ucse225	Se22ucse225@mahindrauniversity.edu.in
Sreeja pamu	Se22ucse257	Se22ucse257@mahindrauniversity.edu.in
Sathwika Alla	Se22ucse311	Se22ucse311@mahindrauniversity.edu.in
V.V. Lakshmi Vyshali	Se22ucse280	Se22ucse280@mahindrauniversity.edu.in
Rohan Gunda	Se22ucse312	Se22ucse312@mahindrauniversity.edu.in
T Karthik Reddy	Se22ucse324	Se22ucse324@mahindrauniversity.edu.in
Yellapragada Pranav	Se22ucse301	Se22ucse301@mahindrauniversity.edu.in
Merajuddin Mohammed	Se22ucse307	Se22ucse307@mahindrauniversity.edu.in

Instructor: *vijay rao*

Course: *Software Engineering*

Lab Section: CSE3-CSE4

Teaching Assistant: *Sravanthi Acchugatla*

Date: *10-03-2025*

CONTENTS II

REVISIONS II

Revisions

NULL

1 Introduction

1.1 Document Purpose

This document serves as the **Software Requirements Specification (SRS)** for the **Aircraft Classification System**. The software is designed to classify aircraft into four categories: **Civilian, Military, UCAV, and Unknown** using **YOLOv8, PyTorch, and OpenCV**. The primary objective of this document is to clearly define the software requirements, functionality, design constraints, and expected performance to ensure a smooth development process.

This SRS focuses solely on **image-based classification**, providing a structured reference for developers, project managers, and stakeholders. While the current version does not include **country identification or video processing**, future versions will integrate these advanced capabilities. The system is designed to be scalable, allowing for iterative improvements as additional features are developed.

1.2 Product Scope

The **Aircraft Classification System** is an **AI-driven software solution** that utilizes **computer vision and deep learning** to classify aircraft from images. The system is primarily aimed at applications in **defense and security**, where timely classification of aircraft can provide crucial intelligence for threat identification and surveillance. Additionally, this system can serve as a foundational tool for **academic research and educational projects** in artificial intelligence, aviation, and security studies.

The system will be trained on a **28GB dataset** and optimized for **real-time aircraft classification**. Key features include:

- **Detect and classify aircraft into four categories:** Civilian, Military, UCAV, Unknown.
- **High-efficiency training using a supercomputer with GPU acceleration.**
- **Deployment via an API to integrate with external applications.**
- **Scalability to support larger datasets and real-time inference.**

Additional Technologies and Future Enhancements

The system currently uses YOLOv8, PyTorch, OpenCV, Roboflow, and CVAT, but future enhancements may require additional technologies. TensorFlow and Keras could be explored for model comparison, while FastAPI and Flask can improve API deployment. To enhance scalability, Docker will enable efficient cloud-based containerization and orchestration. Faster model inference across different hardware platforms can be achieved using ONNX Runtime and TensorRT, while Graph Neural Networks (GNNs) may help analyze relationships between multiple aircraft in a scene.

System Impact and Benefits

This system automates real-time threat analysis, reducing response time for identifying potential aerial threats and integrating AI-based classification with radar detection to improve defense surveillance and aviation monitoring. It ensures faster and more reliable classification, minimizes human error, and provides real-time identification beyond traditional radar methods. Additionally, the system is scalable, allowing integration with multi-source tracking systems, including radar, satellites, and drone footage, to enhance national security operations.

1.3 Intended Audience and Document Overview

This document is intended for a diverse audience, each with a unique role in ensuring the successful development, deployment, and use of the system:

- **Defense Engineers:** Responsible for designing, implementing, and optimizing the system. They will use this document to understand the system architecture, functional requirements, and software dependencies.
- **Defense Researchers:** The system can serve as a reference for furthur academic studies and advancements in aircraft classification and defense applications.
- **Security & Defense Personnel:** Will utilize the system for real-time aircraft identification to support national security, threat detection, and border surveillance operations.
- **Instructors & Reviewers:** For educational evaluation and assessment, ensuring that the system meets industry standards and aligns with modern AI methodologies.

Document Structure:

- **Section 1:** Introduction, including purpose, scope, and intended audience.
- **Section 2:** System overview, architecture, and core components.
- **Section 3:** Specific functional and non-functional requirements.
- **Section 4:** Performance benchmarks, security considerations, and system scalability.
- **Appendices:** Glossary, data dictionary, references, and project logs.

This document describes the **entire system**, including its functional components and internal workflows. It serves as a comprehensive guide, outlining **data flow, model architecture, security considerations, and implementation constraints**. Future iterations of this document will include enhancements for **country identification, real-time video classification, and further AI optimizations**.

1.4 Definitions, Acronyms and Abbreviations

This section defines key **terms, acronyms, and abbreviations** used throughout this document, ensuring clarity and consistency.

Term/Acronym	Definition
AI	Artificial Intelligence, the simulation of human intelligence in machines.
API	Application Programming Interface, allowing different software to communicate.
CNN	Convolutional Neural Network, a type of deep learning model used in image processing.
CVAT	Computer Vision Annotation Tool, used for labeling images for machine learning.
GPU	Graphics Processing Unit, specialized hardware used to accelerate AI computations.
mAP	Mean Average Precision, a metric used to evaluate object detection models.
ONNX	Open Neural Network Exchange, a format for optimizing deep learning models across different platforms.
OpenCV	Open-source Computer Vision library, used for image processing.
PyTorch	A deep learning framework for building and training AI models.
Roboflow	A cloud-based platform for managing and deploying AI-based models.

TensorRT	NVIDIA's deep learning inference optimizer for high-performance AI applications.
TensorFlow	An alternative deep learning framework to PyTorch.
UAV	Unmanned Aerial Vehicle, commonly referred to as a drone.
YOLOv8	You Only Look Once version 8, an object detection model used for real-time aircraft classification.

1.5 Document Conventions

This document follows **IEEE SRS formatting** with the following conventions:

- **Font Style:** Arial, size 11.
- **Headings:** Bold and capitalized.
- **Code Snippets:** Monospace font for readability.
- **References:** IEEE citation format.
- **Diagrams:** High-resolution PNG/SVG for system illustrations.

1.6 References and Acknowledgments

This document is based on various standards and resources, including:

datasets

https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset

https://www.kaggle.com/datasets/rhammell/planesnet/data

https://github.com/dilsadunsal/HRPlanesv2-Data-Set?utm_source=chatgpt.com

https://www.airliners.net/

https://universe.roboflow.com/azra-savasan/vortex-6vbvh/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true

https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset

https://www.airliners.net/search?photoCategory=9

https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset

https://www.kaggle.com/datasets/balajikartheek/drone-type-classification

https://universe.roboflow.com/drones-hsv8d/drones-vs-birds-classification/browse?queryText=&pageSize=50&startingIndex=50&browseQuery=true

https://www.google.com/search?q=unmanned+conbat+arial+bechicles&sca_esv=d492661d9463033a&rlz=1C1CHBD_enKW907KW907&udm=2&biw=1422&bih=621&sxsrf=AHTn8zpULBRVRxXX6UlAaChVTb1kk2mTmA%3A1741356239204&ei=z_zKZ4KKDMGT4-EP5M7s0As&ved=0ahUKEwjCrcLAkfiLAxXByTgGHWQnG7oQ4dUDCBE&uact=5&oq=unmanned+conbat+arial+bechicles&gs_lp=EgNpbWciH3VubWFubmVkIGNvbmJhdCBhcmlhbCBiZWNoaWNsZXNIkz9QAFjqPXACeACQAQCYAX-gAZMZqgEFMjAuMTK4AQPIAQD4AQGYAg2gAsEMqAIKwgIKECMYJxjJAhjqAsICBxAjGCcYyQLCAggQABiABBixA8ICCxAAGIAEGLEDGIMBwgIKEAAYgAQYQxiKBcICDRAAGIAEGLEDGEMYigXCAgUQABiABMICBBAAGB6YAyiSBwQxLjEyoAfeTQ&sclient=img

Reference papers

https://www.researchgate.net/publication/280625683_AIRCRAFT_CLASSIFICATION_USING_IMAGE_PROCESSING_TECHNIQUES_AND_ARTIFICIAL_NEURAL_NETWORKS

https://ieeexplore.ieee.org/document/10374700

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/1306.5151

https://youtu.be/pFiGSrRtaU4?si=ISRkHDKAVa8a5QAd

https://github.com/Hamed-Aghapanah/yolo8-based-aircraft-detection

2 Overall Description

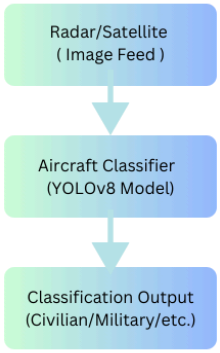
2.1 Product Overview

The **Aircraft Classification System** is a new AI-based tool designed to **work with radar technology** to improve **aviation security and defense monitoring**. Radar systems currently detect aircraft but do not classify them. This system fills that gap by **analyzing images from radar and satellite feeds** to classify aircraft into **Civilian, Military, UCAV, or Unknown**.

The system will be used by **defense personnel, researchers, and aviation authorities** to improve situational awareness and reduce manual classification errors.

How It Works

- **Input:** Images and video from radar and satellite systems.
- **Processing:** Deep learning model (YOLOv8) analyzes images in real-time.
- **Output:** Classified aircraft category (Civilian, Military, UCAV, or Unknown).
- **Usage:** Helps security forces assess threats faster and more accurately.



2.2 Product Functionality

The system performs the following tasks:

- **Classifies aircraft in real-time** from radar and satellite images.
- **Integrates with defense networks** for security monitoring.
- **Provides API access** for external system communication.
- **Generates automated reports** for documentation and analysis.
- **Supports future expansion** for satellite-based classification.

2.3 Design and Implementation Constraints

The development of the Aircraft Classification System is subject to the following constraints:

- **Hardware Requirements:**
  - Requires **high-performance GPUs** for training and inference.
  - Must support **real-time image classification** with low latency.
  - Needs integration with **existing radar systems**.
- **Software Requirements:**
  - Built using **YOLOv8, PyTorch, OpenCV, and Roboflow**.
  - For easy and faster deployments we use **Proton Drive** and **Azure SQL**.
  - Requires **secure API connections** with radar and defense networks.
  - Uses **ONNX Runtime** for optimized model inference.
  - Compliance with **UML modeling** and **COMET software design methodology**.
- **Security Considerations:**
  - Must ensure **classified military data security** with encryption.
  - Requires **user authentication and role-based access control**.
  - Uses **TLS 1.3 and AES-256 encryption** for secure communication.
- **Operational Requirements:**
  - Must function in **low-bandwidth environments and other harsh environments**.
  - Needs **high uptime and reliability** for defense applications.

2.4 Assumptions and Dependencies

The system's performance depends on several external factors:

Assumptions:

- The system assumes **consistent, high-quality image feeds** from radar and satellites.
- Defense networks will provide **secure API access for classified data exchange**.
- The dataset used for training is **accurate and labeled correctly**.
- Hardware used for inference **meets minimum GPU standards**.
- Future **AI improvements will enhance classification accuracy**.

Dependencies:

- **Radar Networks:** The system relies on **radar images as primary input**.
- **Cloud/On-Premise Computing:** Requires **supercomputers or cloud-based GPU servers**.
- **Satellite Integration (Future Versions):** Future updates will add **satellite image classification**.
- **Military Security Compliance:** Must follow **military regulations**.

These assumptions and dependencies play a key role in the system's functionality and development roadmap.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The system will provide a **stand-alone interface** for users to interact with the aircraft classification system. Users will have the following options for submitting and reviewing classification results:

- **Radar-based input:** The system will automatically process images from radar feeds.
- **Drag-and-drop option:** Users will have the ability to manually upload images for classification.
- **Standalone software interface:** The system will operate as a self-contained application, without reliance on web-based dashboards.
- **User roles:**
    - **Admin** – Manages user access and system settings.
    - **Researchers** – Access classification results for AI model improvements.
    - **Officers (Air Force/Military)** – Utilize classification for security assessments.
    - **Aviation Administrators** – Monitor classifications for regulatory compliance.
    - **Defence Engineers** – Analyze system performance and optimize AI models.
- **Error handling:** If an image cannot be classified with confidence, it will be labeled as **Unknown**.
- **History Log:** The system will maintain a **recent history log** of classifications for review.
- **User Feedback:** Users will have the ability to provide feedback on classification accuracy.
- **Language Support:** The system will support **English only** in its initial version.

3.1.2 Hardware Interfaces

The system will integrate with multiple hardware components to support real-time classification and secure data processing.

- **Radar and Satellite Integration:**
    - Supports **military-grade air surveillance radars** (e.g., **AN/TPS-series** or equivalent).
    - Works with **civilian air traffic control radar systems**.
    - Future versions may include **over-the-horizon radar systems** for early detection.
- **Computing Hardware:**
    - **Supercomputers with GPU acceleration** will be used for training.
    - **Operational deployment** will require:
        - **High-performance GPU servers** (e.g., **NVIDIA A100/H100** or equivalent).
        - **Distributed computing infrastructure** for real-time processing.
        - **Sufficient RAM** for handling simultaneous classification tasks.
        - **Fast SSD storage** to cache images and load models efficiently.
- **Bandwidth Requirements:**

    - **Real-time processing:** Requires at least **50-100 Mbps** for high-resolution image transfers.
    - **Compressed images:** Lower bandwidth is required for lower resolution inputs.
- **Edge Computing Support:**
    - While not explicitly part of the initial release, future versions may support **edge deployment** for real-time classification in the field.
    - The system may use **ONNX Runtime or TensorRT** for optimized model inference on edge devices.
    - Model quantization techniques could be applied for deployment in resource-limited environments.
- **Secure Data Storage:**
    - The system will store **classified results** in **secure, encrypted databases**.
    - **Audit logs** will be maintained for all classifications to ensure compliance and traceability.
    - **Access controls** will be enforced based on user clearance levels.
    - The system will follow **strict military security policies** for data retention and handling.
- **Offline Mode Support:**
    - If network connectivity is lost, the system will:
        - Cache models locally to continue classification tasks.
        - Temporarily store classification results for synchronization once connectivity is restored.
        - Allow limited decision-making based on cached reference data.
        - Use optimized lightweight models to ensure continued operation.

These hardware interfaces will enable the system to function efficiently in various defense and aviation monitoring scenarios.

3.1.3 Software Interfaces

The system will integrate with various software components to ensure seamless classification and data exchange.

- **API Support:** No external APIs are required.
- **Data Output Formats:** Classification results will be provided in **JSON or CSV format**.
- **Integration with Radar Systems:** The system will interact with **multiple radar and satellite feeds** to fetch images.
- **External AI Models:** No pre-trained external AI models will be used as of now.
- **Cloud-Based AI Services:** The system will not use cloud-based AI services in its initial release.
- **Compatibility with Legacy Defense Systems:** The system will be designed to integrate with **existing military radar and surveillance infrastructure**.

These software interfaces will ensure that the system remains functional and adaptable to defense and aviation monitoring applications.

3.2 Functional Requirements

3.2.1 Real-Time Classification

The system shall classify aircraft in real-time with the following specifications:

- **Processing Speed:** Classification should occur in **<50ms per image**, but accuracy is prioritized over speed.
- **Handling Unclassified Images:** If an image cannot be classified confidently, it will be assigned to the **Unknown** category.
- **Multiple Aircraft Detection:** The system can classify multiple aircraft within a single image.
- **Confidence Score:** Each classification will include a confidence score to indicate reliability.
- **Preprocessing:** The system will apply preprocessing techniques to improve image quality but can also request a new image if needed.
- **Handling Partial or Obstructed Aircraft:** The system does not explicitly classify partially visible aircraft but will label them as **Unknown** and request a new image.

Handling Motion Blur in Video Feeds

The system will employ the following techniques to reduce the impact of motion blur:

- **Frame Selection Algorithms:** Identifies and processes the clearest frames.
- **Training Data Augmentation:** Includes motion-blurred examples to improve model robustness.
- **Temporal Averaging:** Analyzes multiple frames to enhance classification reliability.

Handling Conflicting Classification Results

If classification results conflict with previous classifications, the system will:

- **Implement Confidence Scoring:** Weighs the reliability of each classification.
- **Maintain Classification History:** Detects unusual classification changes.
- **Flag Anomalous Changes:** Alerts users for human review in case of unexpected classification shifts.
- **Apply Decision Fusion Algorithms:** Uses multiple classification sources to improve accuracy.

Visual Markers for Detected Aircraft

The system will include visual markers to assist in aircraft identification:

- **Bounding Boxes:** Displayed around detected aircraft.

- **Color Coding:** Different colors for each classification category (Civilian, Military, UCAV, Unknown).
- **Confidence Scores:** Shown visually for each detection.

These functionalities will ensure accurate, efficient, and transparent aircraft classification for defense and aviation monitoring applications.

3.2.2 Classification Report Generation

The system will generate classification reports within its standalone interface for authorized users.

- **Report Formats:** Classification reports will be available in **CSV or JSON format**.
- **Automated Reports:** The system does not generate automated daily reports.
- **Historical Data Access:** Certain past classification history will be accessible, but only to **authorized personnel**.
- **Inclusion of Confidence Scores:** Reports will include confidence levels for each classification.
- **Security Standards:** The system follows **military security protection standards** to ensure data integrity and confidentiality.
- **Data Retention and Deletion:** Reports will not be deleted automatically but can be manually deleted by **admins**.
- **Data Sharing:** Reports can be shared, but only with **authorized personnel**.
- **Filtering Reports:** The system does not support automatic filtering of reports.
- **Exporting Reports:** Reports cannot be exported beyond the system's secure environment.

3.2.3 Radar Data Integration

The system integrates with radar data to improve classification accuracy.

- **Data Input Format:** The system processes **raw images** along with a structured **metadata JSON file**, which includes:
    - **Captured Location**
    - **Captured Time**
    - **Other relevant radar details**
- **Supported Data Types:** The system currently supports **image-based input**, with future support for **video**.
- **Timestamp Synchronization:** The system uses **UTC-synchronized timestamps** embedded in radar data for **temporal accuracy**.
- **Multi-Radar Support:** The system supports **multiple radar sources** to eliminate blind spots.
- **Fallback Mechanisms for Missing Data:** If radar data is incomplete, the system will:
    - Interpolate between known data points where applicable.
- **Noise Filtering:** The system applies **sophisticated filtering algorithms** to distinguish between:
    - **Actual aircraft**

    - **Radar noise**
    - **Weather effects**
    - **Other anomalies**
- **Processing 3D Radar Data:** The system supports **3D radar image processing** for better classification.
- **Communication Protocols:** Data transmission uses **secure MQTT or dedicated UDP streams**.
- **Manual Adjustments:** Users can manually adjust radar-detected aircraft classifications.
- **Data Format Handling:** The system only accepts **one standardized data format** for radar inputs.

These functionalities ensure accurate, secure, and efficient aircraft classification using radar integration.

3.2.4 Radar-Based Classification and Data Handling

The system incorporates radar-based classification and structured data handling:

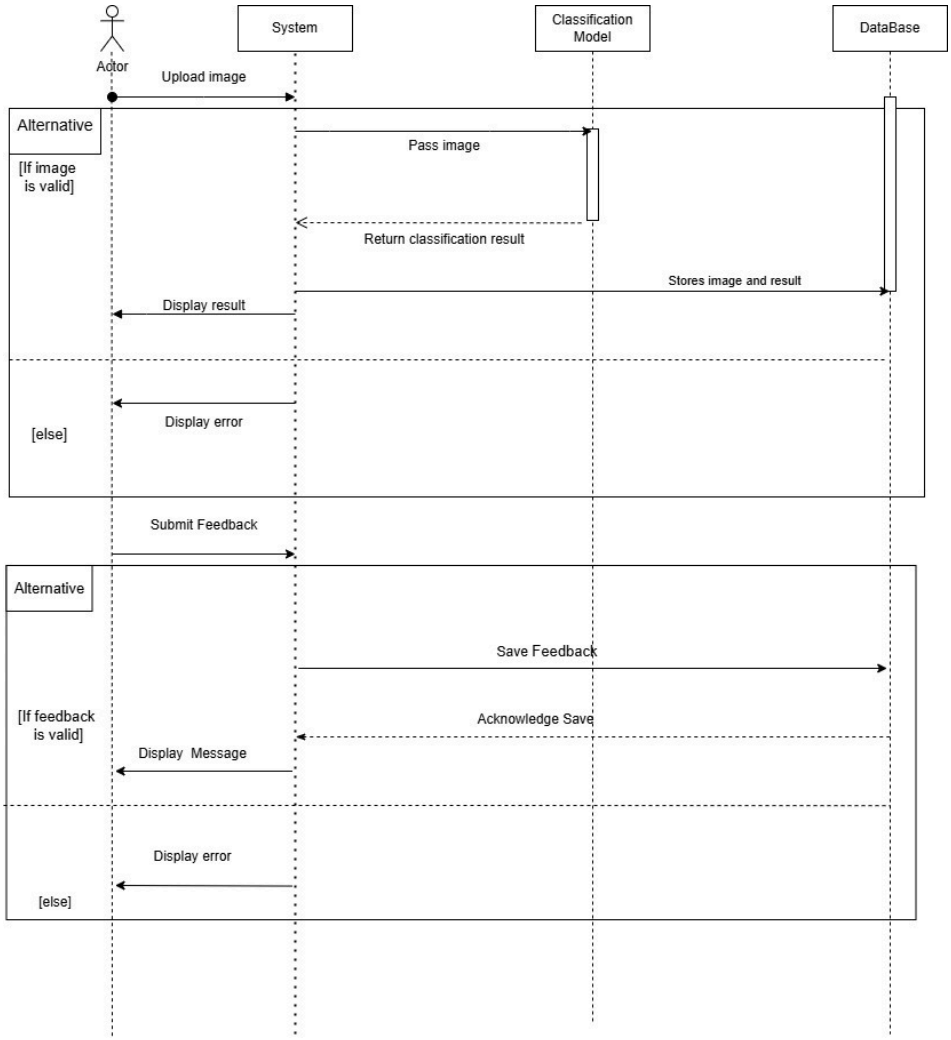
- **Data Types Processed:**
    - **Raw images** for classification.
    - **Structured metadata JSON file** containing:
        - **Captured location.**
        - **Captured time.**
        - **Other aircraft identification data.**
- **Image-Based Input First, Video Later:**
    - Initially, the system will process **image-based input**.
    - Future versions will support **video-based analysis**.
- **Timestamp Synchronization:**
    - The system will use **UTC-synchronized timestamps** embedded in radar data to ensure precise temporal tracking.
- **Multi-Radar Support:**
    - The system will accept input from multiple radars to **eliminate blind spots**.
- **Fallback Mechanisms for Missing Data:**
    - If radar data is missing or incomplete, the system will:
        - **Interpolate between known data points** where possible.
- **Radar Noise Filtering:**
    - The system will implement advanced filtering algorithms to **differentiate real aircraft from radar noise** caused by:
        - **Weather conditions.**
        - **Environmental interferences.**
        - **System anomalies.**
- **Processing 3D Radar Data:**
    - The system supports processing **3D radar images** for better aircraft classification.
- **Secure Communication Protocols:**

- The system will use **secure MQTT or dedicated UDP streams** for real-time radar data transmission.

- **Manual Classification Adjustments:**
  - Users will be able to manually correct or adjust radar-detected aircraft classifications.
- **Data Format Handling:**
  - The system will **only accept one standardized data format** to ensure uniformity in processing.

These radar-based functionalities will ensure accurate classification, enhanced security, and future adaptability.

3.3 Use Case Model



3.3.1 Defense & Military Surveillance (U1)

Author: Sreeja and ritika

Purpose:

To enhance threat detection and response by classifying aircraft detected by radar into Civilian, Military, UAV, or Unknown in real-time, allowing defense teams to assess potential threats and take action.

Requirements Traceability:

- R1: Real-time classification of aircraft
- R2: Integration with radar systems
- R3: Automated threat alerting

Priority: High

Preconditions:

- Radar system must provide input images
- System must be running and operational

Postconditions:

- Aircraft classified into one of the four categories
- Alerts triggered for unauthorized or military aircraft
- Classification results stored for further analysis

Actors:

- Military personnel
- Air defense operators
- Radar systems

Extends: None

Flow of Events:

1. **Basic Flow:**
    1. Radar captures an image of an aircraft
    2. Image is processed by the classification system
    3. Aircraft is categorized as Civilian, Military, UAV, or Unknown
    4. If unauthorized aircraft is detected, an alert is raised
    5. Classification data is stored for analysis

2. **Alternative Flow:**
    - If image quality is poor, the system requests a new image from the radar system
    - If classification confidence is low, the system labels it as Unknown
3. **Exceptions:**
    - Radar system failure: System waits for radar to resume operation
    - Processing failure: System logs an error and retries

Includes: None

Notes/Issues: None

3.3.2 Border Surveillance & Homeland Security (U2)

Author: vyshali,Rohan and karthik

Purpose:

To detect unknown aircraft and UCAVs (Unmanned Combat Aerial Vehicles) that enter restricted or no-fly zones, enabling security forces to track and respond in real-time.

Requirements Traceability:

- R4: Identification of unauthorized aircraft in no-fly zones
- R5: Multi-radar integration for blind spot elimination
- R6: Automated alert system for unidentified UCAVs

Priority: High

Preconditions:

- Radar system must provide real-time image feeds
- Security teams must have access to classification results

Postconditions:

- Unauthorized aircraft and UCAVs identified in no-fly zones
- Alerts triggered for unknown aerial activity
- Data stored for intelligence analysis

Actors:

- Border security agencies
- Defense intelligence units
- Radar systems

Extends: None

Flow of Events:

1. **Basic Flow:**
    1. Radar captures an image in a restricted airspace
    2. Image is processed by the classification system
    3. Aircraft classified as Civilian, Military, UAV, or Unknown
    4. If an Unknown or UCAV is detected, alert is triggered
    5. Data is logged for further review
2. **Alternative Flow:**
    - If radar feed is missing, system attempts to retrieve past data
    - If UCAV is detected but its origin is uncertain, security teams are notified
3. **Exceptions:**
    - False positive detection: Security team manually verifies classification
    - Radar signal loss: System continues processing available data

Includes: None

Notes/Issues: None

3.3.3 AI Research & Future Expansion with Satellite Monitoring (U3)

Author: Sathwika, Meraj and pranav

Purpose:

To enhance AI models for aircraft classification and support large-scale aircraft tracking using satellite imagery in future versions.

Requirements Traceability:

- R7: AI model training and improvement
- R8: Satellite integration for large-scale tracking
- R9: Fusion of radar and satellite data for classification

Priority: Medium

Preconditions:

- Dataset for AI training must be available
- Satellite image processing capability must be implemented

Postconditions:

- Improved AI models with enhanced classification accuracy
- Future versions capable of satellite-based classification
- Integration of radar and satellite data

Actors:

- AI researchers
- Defense technology developers
- Space agencies
- Intelligence surveillance teams

Extends: None

Flow of Events:

1. **Basic Flow:**
    1. AI researchers feed training dataset into the system
    2. Model is trained to classify aircraft accurately
    3. Performance metrics are analyzed
    4. Future integration with satellite imagery is planned
2. **Alternative Flow:**
    - If training data contains errors, system flags and requests review
    - If satellite image quality is poor, system prioritizes radar data
3. **Exceptions:**
    - Data corruption: System logs and requests a fresh dataset
    - Model training failure: System restarts with last known good state

Includes: None

Notes/Issues: None

4 Other Non-functional Requirements

4.1 Performance Requirements

The aircraft classification system should work quickly and accurately. It must classify each image in under 50 milliseconds to support real-time defense applications. The system should maintain an accuracy of at least 90%, even when multiple aircraft are present in the same image.

The software will run on GPUs to speed up processing and should handle at least 10 images per second. When analyzing videos, it should select the best frames to avoid unnecessary processing. The system should also be able to handle challenges like motion blur, low lighting, and partially visible aircraft to improve reliability.

4.2 Safety and Security Requirements

Since this system deals with sensitive defense data, security is a top priority. All stored data, including images and classification results, must be encrypted to prevent unauthorized access. The system should use secure connections when transferring data and ensure that only authorized personnel can view or modify results.

To prevent misclassification, human analysts should have the option to review and correct classifications if needed. The system should also work offline in case of network issues and store results locally until the connection is restored. It must follow military security rules for data protection and compliance.

4.3 Software Quality Attributes

4.3.1 Reliability

The system should be available at all times, with minimal downtime to ensure continuous operation in defense applications. To prevent disruptions, it should have backup mechanisms such as redundant GPU servers that take over if the primary system fails. Regular updates will include model retraining with new aircraft images to maintain high accuracy. For example, if the model starts misclassifying aircraft due to changes in real-world conditions, new training data should be incorporated to improve reliability.

4.3.2 Scalability

The system should handle increasing amounts of data without performance issues. It must process larger datasets, more aircraft images, and additional classification requests efficiently. For instance, if the system is expanded to classify aircraft from satellite and drone footage, it should scale up using distributed computing without slowing down. The architecture should also allow easy integration of new aircraft categories such as stealth aircraft by retraining the model without major code modifications.

4.3.3 Maintainability

Updating the system should be simple and efficient, allowing for quick improvements without major changes. The software will be modular and well-documented, ensuring that developers can easily modify or upgrade components. For example, if a new AI model (e.g., YOLOv9) offers better accuracy, it should be possible to replace the classification model without rewriting the entire system. The use of Docker containers will help maintain compatibility across different deployment environments.

4.3.4 Usability

The system should have a clear and intuitive interface that displays aircraft classifications, confidence scores, and bounding boxes to highlight detected objects. Users, such as defense

analysts and security personnel, should be able to navigate the interface without extra training. For example, if an aircraft is classified as "Unknown," the system should provide an option for manual review, allowing experts to verify and update the classification.

This ensures the system is reliable, scalable, maintainable, and user-friendly, making it effective for real-world defense applications.

5 Other Requirements

*<This section is **Optional**.>*
NULL

Appendix A – Data Dictionary:

1. Constants:

- **IMAGE_SIZE:** Input image resolution (default 640x640 for YOLOv8).
- **CONFIDENCE_THRESHOLD:** Minimum confidence score for detecting an aircraft.
- **NMS_THRESHOLD:** Non-Maximum Suppression threshold to filter overlapping detections.
- **CLASS_NAMES:** Categories for classification—Civilian, Military, Drone, Unknown.

2. State Variables:

- **Model_State:** Indicates if the YOLOv8 model is loading, running, or idle.
- **Detection_Results:** Stores detected aircraft types and their confidence scores.
- **Bounding_Boxes:** Coordinates of detected aircraft in the image.
- **Training_Epoch:** Tracks the current epoch during model training.

3. Inputs:

- **Input_Image:** Raw ground-based aircraft images for classification.
- **Pretrained_Weights:** YOLOv8 model weights used for transfer learning.
- **Threshold_Values:** Adjustable parameters like confidence and NMS thresholds.
- **Annotation_Labels:** Manually labeled data for supervised learning.

4. Outputs:

- **Detected_Objects:** Classified aircraft with bounding boxes and confidence scores.
- **Processed_Image:** Image with annotations highlighting detected aircraft.
- **Training_Logs:** Metrics like loss, accuracy, and epoch progression.
- **Performance_Report:** Precision, recall, and F1-score of the classification model.

Appendix B - Group Log

Minutes of Meeting

Project: *AirCraft Classification*
Date: *8-2-25 and 9-2-25*
Time: *[11:30 AM] – [9:30 PM]*
Location: *Mahindra University*

---

1. Objective of the Meeting:

- *To discuss and finalize the structure and content for the* ***Software Requirements Specification (SRS)*** *document.*

---

2. Key Discussion Points:

2.1 SRS Template Overview:

- *Reviewed the IEEE SRS template provided.*
- *Agreed to follow sections: Introduction, Overall Description, Specific Requirements, Non-functional Requirements, and Appendices.*

2.2 Scope and Purpose:

- *Defined the scope of the project as: The Aircraft Classification System is an AI-based solution that uses computer vision and deep learning to classify aircraft from images. It is designed for defense, security, and academic research applications, providing timely intelligence and supporting AI projects.*

2.3 Functional Requirements:

• Identified key functionalities:

o Real-Time Classification: Detects multiple aircraft in <50ms with confidence scoring.

o Unclassified & Obstructed Aircraft: Labels unclear cases as "Unknown" and requests new images.

o Motion Blur Handling: Uses frame selection, data augmentation, and temporal averaging.

o Conflicting Results: Applies confidence scoring, anomaly detection, and decision fusion.

o Visual Markers: Uses bounding boxes, color coding, and confidence displays.

2.4 Non-functional Requirements:

- *Discussed performance, security, and reliability requirements.*

- *Agreed to include requirements for usability and scalability.*

2.5 Roles and Responsibilities:

- ***Introduction & Scope:*** *[Meraj,Pranav]*
- ***Functional Requirements:*** *[Sreeja , Ritika]*
- ***Non-functional Requirements:*** *[Karthik,Sathwika]*
- ***Diagrams and Appendices:*** *[Vyshali,Rohan]*

---

3. Decisions Made:

- *Use **PyTorch** for development.*
- *Follow the **IEEE format** strictly for the SRS.*

Meeting Adjourned at: *[9 PM - 9/3/25]*
Minutes Prepared by: *[Meraj]*