Project Title: Aircraft Identification and Classification Using Machine Learning Techniques

1. Task Descriptions

Below is the list of major activities that are to be carried out under this project. Each activity has been described briefly in order to comprehend the workflow of the project.

1.1 Project Establishment and Literature Review

- Conduct an extensive literature review to understand past research on aircraft identification and classification.

- Review different machine learning and computer vision techniques applied in similar projects.

- Synthesize key findings and incorporate conclusions into project planning.

1.2 Tool Configuration and Environmental Setup

- Install necessary software libraries (PyTorch, TensorFlow, OpenCV, Scikit-learn).

- Establish the necessary development environment, i.e., Jupyter Notebook, Visual Studio Code, or PyCharm.

- Set up hardware resources, such as activating GPU acceleration if present.

1.3 Dataset Selection and Preprocessing

- Access a publicly available aircraft dataset, such as FGVC-Aircraft or other suitable sources.

- Apply data augmentation and preprocessing techniques, such as resizing, normalization, and enhancement.

- Split data into training, validation, and test sets.

1.4 Model Selection and Baseline Implementation

- Employ baseline methods such as HOG + SVM and simple CNN models.

- Evaluate early model performance to set a baseline.

- Identify key shortcomings within the base methodology to allow improvement.

1.5 Developing and Refining Advanced Models

- Use deeper models such as ResNet, MobileNet, and Transformer models.

- Utilize transfer learning to fine-tune pre-trained models.

- Hyperparameters should be tuned using grid search or other tuning techniques.

1.6 Model Evaluation and Error Analysis

- Assess model performance using accuracy, precision, recall, F1-score, and confusion matrices.

- Analyze failure cases, e.g., false positives and false negatives.

- Compare results across different models and justify the best-performing architecture.

1.7 System Deployment and Testing

- Develop a prototype system for field tests.

- Integrate the model in an interface (e.g., web application or desktop application) for detecting/classifying aircraft.

- Evaluate system strength on new data and videos.

1.8 Documentation and Progress Reports

- Maintain project records, such as results, conclusions, and methodology.

- Prepare and submit progress reports as needed.

1.9 Preparation for Final Thesis Composition and Presentation

- Compile the final project report in line with university requirements.

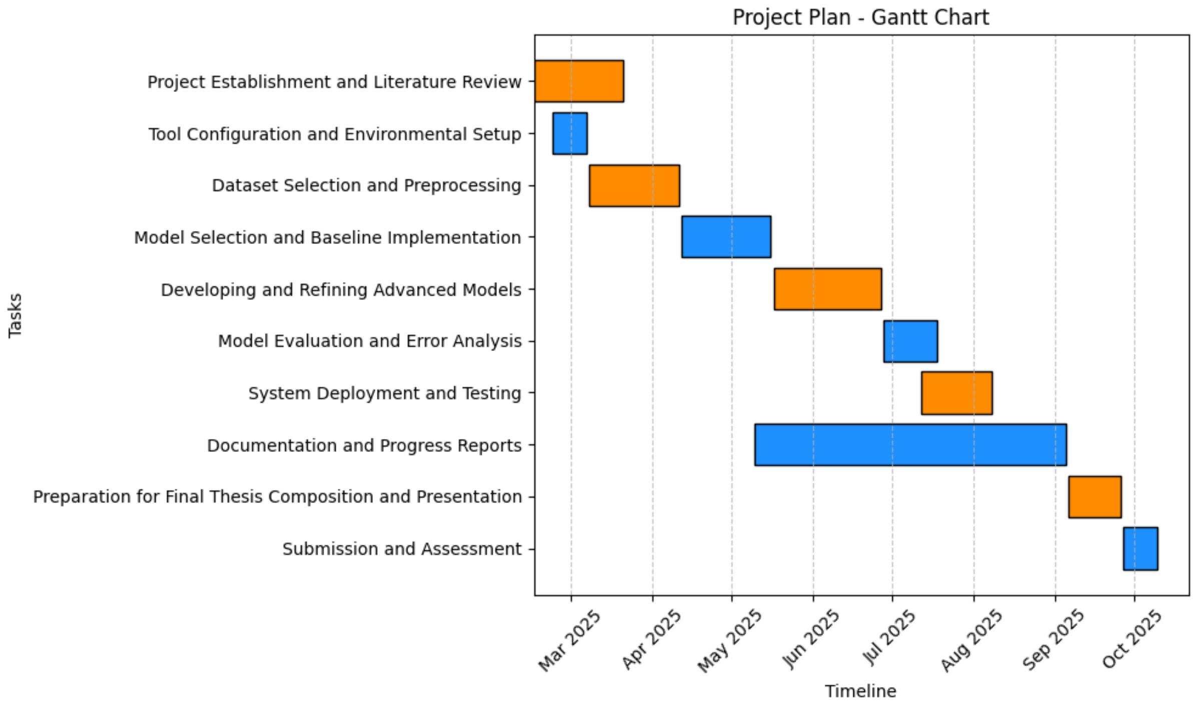
- Develop visual aids and materials for the final presentation.

1.10 Submission and Assessment

- Present the final project and deliverables.

- Take part in the evaluation process and respond to criticisms.

2. Gantt Chart



3. Project Requirements

3.1 Hardware Specifications

- Computer: A powerful desktop or laptop computer with a minimum of 16GB of RAM and a modern multi-core processor.

- GPU: A dedicated GPU such as NVIDIA RTX 3060 or higher for deep learning model training.

- Storage: At least 500GB of SSD storage for data, models, and logs.

- Cloud Computing: Access to cloud facilities such as Google Colab Pro or AWS for additional computational power if needed.

3.2 Computational Resources

- Local run on macOS with MPS support for PyTorch.

- Optionally using university high-performance computing (HPC) clusters as required.

- Cloud-based GPU acceleration from Google Colab/AWS/Google Cloud AI if required for large-scale training.

3.3 Software Resources

- Operating System: macOS/Linux/Windows
- Programming Languages: Python (mainly), Bash (employed for automation)
- Libraries and Frameworks:
    - PyTorch (to implement the deep learning model)
    - TensorFlow (alternative deep learning framework)
    - OpenCV (for image processing)
    - Scikit-learn (for classical machine learning techniques)
    - Matplotlib & Seaborn (for plots)
    - Torchvision (for preprocessing and dataset manipulation)
- Development Tools:
    - Jupyter Notebook, Visual Studio Code, PyCharm
    - Git for version control.
    - Docker (optional for environmental consistency)

3.4 Data Requirements

- Dataset:
    - FGVC-Aircraft dataset or custom annotated aircraft dataset.
    - Background-only images to train a presence detection model.

- Data Preparation:
    - Normalization, resizing, and augmentation to enhance generalization.
    - Splitting into training, validation, and test sets for rigorous evaluation.
    - Collection of Data (if necessary): Apply web scraping techniques or procure public repositories to collect more data.

- Guarantee ethical collection and data privacy regulation compliance.

3.5 Potential Challenges and Mitigation Strategies

- Insufficiency of Data: Augment the existing datasets or use transfer learning from pre-trained models.
- Computational Limitations: Utilize university HPC facilities or cloud training.
- Model Overfitting: Employ regularization methods, including dropout and data augmentation techniques.
- Class Imbalance: Employ data resampling methods such as SMOTE or utilize weighted loss functions.
- Real-world Testing: Get additional real-world images/videos to test model strength.