Military Aircraft Detection and Classification from Satellite Images Using YOLOv8 and CNN–Vision Transformer Hybrid Model

Sravya Matta
Department of Computer Science and Engineering
PES University
Bengaluru, India
mattasravya444@gmail.com

K. Prabhath Reddy
Department of Computer Science and Engineering
PES University
Bengaluru, India
prabhathbdl@gmail.com

Sai Keerthana K
Department of Computer Science and Engineering
PES University
Bengaluru, India
saikeerthanakalimisetty@gmail.com

Priyanka H
Department of Computer Science and Engineering
PES University
Bengaluru, India
drpriyankahsachin@gmail.com

***Abstract*—Satellite imaging and remote sensing technology have advanced rapidly, providing access to high-resolution aerial imagery useful in defense surveillance, disaster response, and aviation monitoring. However, manually identifying and classifying military aircraft from large-scale satellite data is slow, labor-intensive, and error-prone. This paper presents an automated deep learning-based system for detecting, recognizing, and counting military aircraft in satellite images. The system employs a two-stage cascade architecture combining YOLOv8n for object detection with a pretrained ResNet-50 or Vision Transformer (ViT-B/16) classifier for fine-grained aircraft recognition. YOLOv8n localizes aircraft with high accuracy, while the classifier distinguishes between 20 military aircraft categories. The model is trained and evaluated on the MAR20 dataset. A Flask-based web interface with ReactJS frontend enables users to upload images and visualize detection results. Experimental results demonstrate 92.3% mAP for detection and 90.5% classification accuracy, making it suitable for automated defense and surveillance applications.**

***Index Terms*—YOLOv8, Vision Transformer, ResNet-50, CNN, remote sensing, satellite imagery, aircraft detection, deep learning, transfer learning**

I. Introduction

Satellite and aerial imagery play an important role in modern defense, surveillance, and strategic monitoring. High-resolution images captured by satellites and unmanned aerial vehicles (UAVs) can reveal the presence, type, and distribution of military assets, including aircraft stationed at airbases or deployed in the field. However, manually inspecting these images to identify and classify aircraft is time-consuming and difficult to scale when dealing with large regions or continuous monitoring requirements.

Traditional image processing techniques are often sensitive to variations in lighting, resolution, viewing angle, and background clutter. These limitations reduce their reliability in real-world remote sensing environments. As the demand for near real-time situational awareness increases, there is a growing need for automated methods that can detect and classify military aircraft accurately and efficiently.

Recent advances in deep learning have led to significant progress in object detection and image classification. One-stage detectors such as YOLO [7] and SSD can localize objects quickly, while Convolutional Neural Networks (CNNs) such as ResNet [9] and Vision Transformers (ViTs) [8] have improved performance on fine-grained recognition tasks. Building on these developments, this work proposes an integrated two-stage cascade system that combines YOLOv8n for aircraft detection and a pretrained ResNet-50 or ViT-B/16 model for aircraft type classification.

The main contributions of this paper are:

- A two-stage cascade pipeline for detecting, classifying, and counting military aircraft in satellite images using YOLOv8n and ResNet-50/ViT-B/16.
- A crop-based training approach that extracts aircraft regions from YOLO-format annotations to train the classifier.
- Transfer learning from ImageNet-pretrained models with custom classification heads for fine-grained aircraft recognition across 20 categories.
- A practical web-based interface using Flask backend and ReactJS frontend for uploading satellite images and visualizing annotated detection results with fleet composition analysis.

The remainder of this paper is organized as follows: Section II reviews related work in aircraft detection and remote sensing. Section III describes the proposed methodology and model architecture. Section IV presents implementation de-

tails. Section V discusses experimental results, and Section VI concludes the paper.

II. Related Work

Deep learning-based methods have been widely explored for object detection and classification in aerial and satellite images. Several studies have specifically focused on aircraft detection due to its importance in defense and surveillance applications.

A significant contribution to this field is the MAR20 dataset introduced by Yu *et al.* [1], which provides more than 22,000 labeled aircraft instances across 20 military aircraft categories. The dataset includes both horizontal and rotated bounding boxes and has been used to evaluate advanced detectors such as Faster R-CNN, RetinaNet, ATSS, FCOS, and RoI Transformer under challenging conditions including occlusion, atmospheric distortion, and illumination variation.

Cheng *et al.* [2] proposed a system combining object detection with tracking to identify aircraft in UAV-based imagery, improving stability in continuous detection scenarios. Ji *et al.* [3] introduced a multi-angle feature extraction approach using multiple CNN models and majority voting, which enhanced detection performance when aircraft appeared at different orientations.

Wang *et al.* [4] investigated lightweight variants of YOLOv5 to achieve faster inference while maintaining reasonable accuracy, making such models suitable for real-time or resource-constrained deployments. Wu *et al.* [5] developed CGC-NET, which combines deep learning features with handcrafted image descriptors to reduce false positives and more accurately locate aircraft centers in complex backgrounds.

Hu *et al.* [6] presented GLF-Net, a network that fuses global and local feature representations for aircraft recognition in high-resolution satellite imagery. Their work showed that capturing both large-scale context and fine structural details is important for distinguishing visually similar aircraft types.

The YOLO family of detectors [7] has evolved significantly, with recent versions such as YOLOv8 bringing improvements in both speed and accuracy for object detection tasks. Vision Transformers [8] have demonstrated strong performance in image classification by modeling global dependencies through self-attention mechanisms, while deep residual networks such as ResNet [9] remain effective for feature extraction due to their skip connections.

YOLOv8 [10] offers improved feature extraction through an enhanced backbone architecture, making it well-suited for detecting small objects in satellite imagery.

A. Research Gaps

Despite significant progress in aircraft detection and classification, several research gaps remain in the existing literature:

1) **Single-stage limitations:** Most existing approaches [2], [4] rely on single-stage detection networks that perform both localization and classification simultaneously. This tightly coupled design limits classification accuracy for fine-grained recognition of visually similar aircraft types.
2) **Limited classifier architectures:** Prior works [3], [5] primarily use CNN-based classifiers without exploring the potential of Vision Transformers, which excel at capturing global context and long-range dependencies critical for distinguishing aircraft with similar silhouettes.
3) **Lack of hierarchical categorization:** Existing systems [1], [6] focus on identifying specific aircraft models but do not provide operational category information (e.g., fighter, bomber, transport), which is essential for strategic fleet composition analysis.
4) **Absence of practical deployment interfaces:** Most research focuses on model development without providing user-friendly interfaces for real-world deployment. This limits the practical applicability of these systems in operational defense scenarios.
5) **Dataset-specific evaluation:** Many approaches are evaluated on limited or proprietary datasets, making it difficult to compare performance across different methods under standardized conditions.

B. Objectives

To address the identified research gaps, this work aims to achieve the following objectives:

1) **Develop a two-stage cascade architecture:** Decouple detection and classification into independent stages, allowing each component to be optimized separately for improved fine-grained recognition accuracy.
2) **Integrate CNN and Vision Transformer classifiers:** Implement and compare ResNet-50 (CNN-based) and ViT-B/16 (Transformer-based) classifiers to leverage both local texture features and global contextual information.
3) **Enable hierarchical fleet composition analysis:** Provide classification results at two levels—specific aircraft type (e.g., F-22, B-52) and operational category (e.g., fighter, bomber)—to support both tactical and strategic assessments.
4) **Create a deployable web-based interface:** Develop a Flask-React application that allows non-expert users to upload satellite images and visualize detection results with annotations, counts, and composition statistics.
5) **Benchmark on standardized dataset:** Train and evaluate all models on the publicly available MAR20 dataset with 20 aircraft categories to enable reproducible comparisons with future work.

III. Proposed Methodology

This section describes the overall pipeline for military aircraft detection and classification from satellite images. The system employs a two-stage cascade architecture: (1) object detection using YOLOv8n to localize aircraft, and (2) fine-grained classification using ResNet-50 or ViT-B/16 on cropped regions.

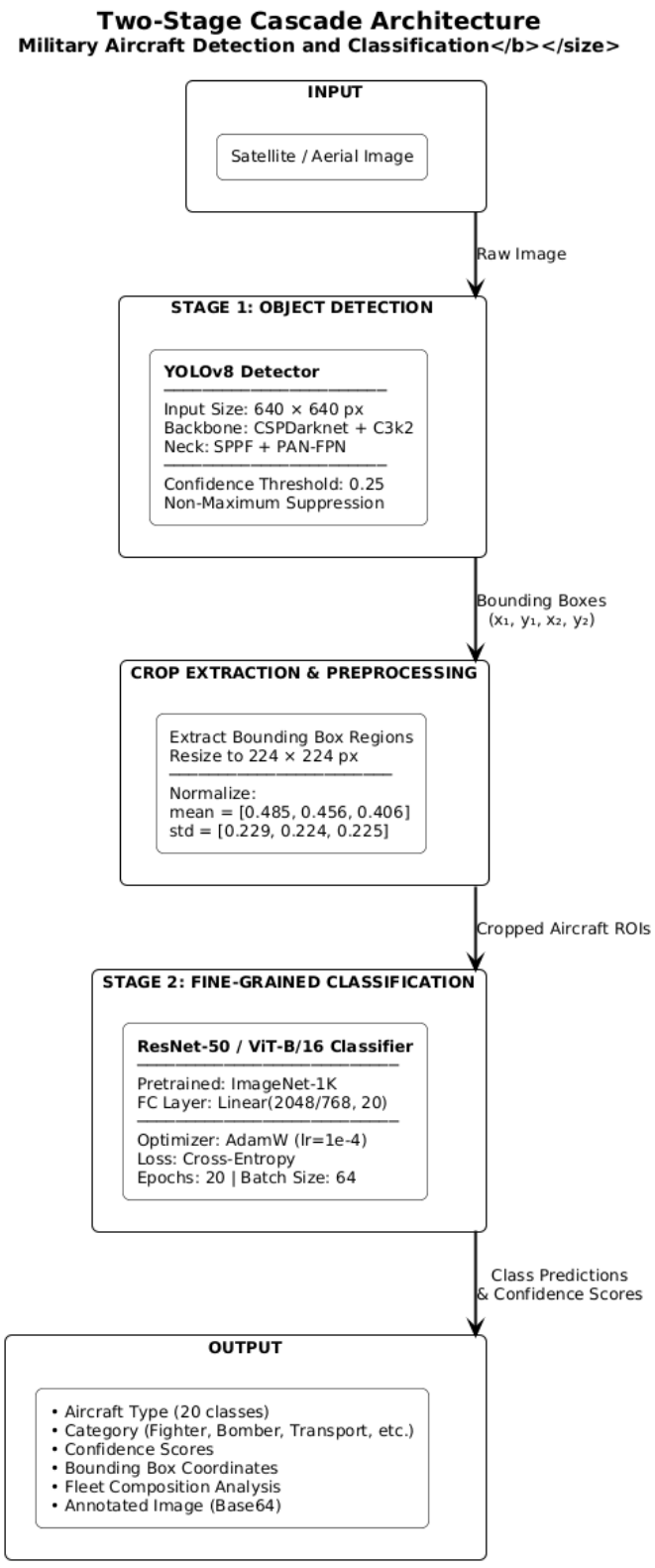

Fig. 1. Overall system architecture for aircraft detection and classification.

TABLE I
AIRCRAFT CATEGORIES AND TYPES IN MAR20 DATASET

Category	Aircraft Types
Fighter	F-15, F-16, F-22, FA-18, SU-35
Bomber	B-1B, B-52, TU-160, TU-22, TU-95
Transport	C-130, C-17, C-5
AWACS	E-3, E-8
Tanker	KC-10, KC-135
Attack	SU-34, SU-24
Patrol	P-3C

A. Two-Stage Cascade Architecture Overview

The proposed system operates in two stages:

Stage 1 (Detection): YOLOv8n processes the full satellite image and outputs bounding boxes with confidence scores for detected aircraft.

Stage 2 (Classification): For each detected bounding box, the corresponding region is cropped from the original image, resized to 224×224 pixels, normalized, and passed through a pretrained ResNet-50 or ViT-B/16 classifier to predict the specific aircraft type.

This decoupled approach allows each stage to be optimized independently and provides flexibility in choosing different classifier architectures without modifying the detection pipeline.

B. Dataset Preparation

The MAR20 dataset is used to train and evaluate the system. It contains satellite images with annotated military aircraft across twenty categories as shown in Table I.

All annotations are in YOLO format, where each image has an associated text file containing lines of the form:

$$\texttt{class\_id} \quad c_x \quad c_y \quad w \quad h \tag{1}$$

where (c_x, c_y) is the normalized center coordinate and (w, h) are the normalized width and height of the bounding box.

C. Preprocessing and Data Augmentation

1) For Detection (YOLOv8n): Images are resized to 640×640 pixels. Standard YOLO augmentations including mosaic, mixup, and random perspective transformations are applied during training.

2) For Classification: Cropped aircraft regions are resized to 224×224 pixels and normalized using ImageNet statistics:

$$\mu = [0.485, 0.456, 0.406], \quad \sigma = [0.229, 0.224, 0.225] \tag{2}$$

Training augmentations applied to crops include random horizontal flip, random rotation ($\pm 10^\circ$), and color jitter (brightness=0.2, contrast=0.2, saturation=0.2, hue=0.02).

Fig. 2. Example satellite images from the MAR20 dataset.

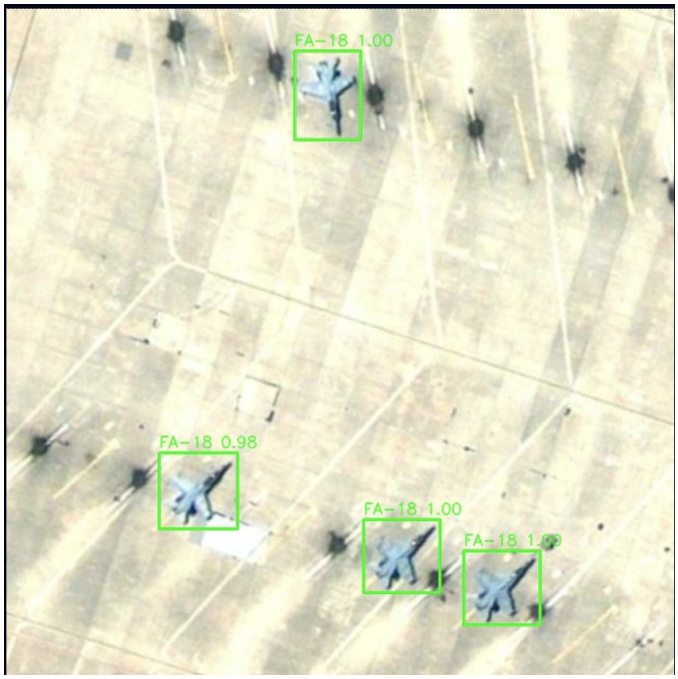

Fig. 3. Sample YOLOv8n detection output with bounding boxes.

TABLE II
YOLOv8n Detection Model Configuration

Parameter	Value
Architecture	YOLOv8n (Ultralytics)
Pretrained Weights	yolov8n.pt (COCO)
Input Size	640×640 pixels
Training Epochs	30
Batch Size	16

D. Stage 1: Aircraft Detection Using YOLOv8n

YOLOv8n [10] is used as the detection backbone. It represents the latest iteration of the YOLO family, offering improved feature extraction through an enhanced CSPDarknet backbone with C3k2 blocks and SPPF (Spatial Pyramid Pooling – Fast) layers. Table II shows the detection model configuration.

The network outputs bounding boxes (x_1, y_1, x_2, y_2), confidence scores, and class labels for each detected aircraft. Post-processing involves confidence thresholding (0.25), Non-Maximum Suppression (NMS), and coordinate clamping.

E. Stage 2: Fine-Grained Classification

For each detected aircraft region, a cropped image is extracted and classified using a pretrained CNN or Vision Transformer.

1) ResNet-50 Classifier: ResNet-50 [9] is a 50-layer deep residual network with skip connections. The original 1000-class ImageNet classification head is replaced with:

$$\texttt{model.fc} = \text{Linear}(2048, 20) \tag{3}$$

TABLE III
Classification Model Configuration

Parameter	Value
Default Architecture	ResNet-50
Alternative Architecture	ViT-B/16
Pretrained Weights	ImageNet-1K
Input Size	224×224 pixels
Training Epochs	20
Batch Size	64
Learning Rate	1×10^{-4}
Optimizer	AdamW
Loss Function	Cross-Entropy
Number of Classes	20

2) Vision Transformer ViT-B/16 Classifier: ViT-B/16 [8] divides the input image into 16×16 patches and processes them through 12 transformer encoder layers. The classification head is modified as:

$$\texttt{model.heads.head} = \text{Linear}(768, 20) \tag{4}$$

Table III summarizes the classification model configuration.

3) YOLO Crop Dataset: A custom dataset class (`YoloCropDataset`) extracts training crops from YOLO-format annotations. For each annotation line, the corresponding bounding box region is cropped from the source image and associated with its class label.

F. Inference Pipeline

During inference, the complete pipeline operates as follows:

1) **Input:** Load satellite image as RGB array

TABLE IV
TECHNOLOGY STACK

Component	Tools/Libraries
Programming	Python 3.11, JavaScript
Backend	Flask with Flask-CORS
Frontend	ReactJS
Deep Learning	PyTorch 2.x, Ultralytics
Image Processing	OpenCV, Pillow, NumPy
Dataset	MAR20 (20 classes, 22K+ instances)
Hardware	NVIDIA GPU (CUDA)

2) **Detection:** Run YOLOv8n with confidence threshold 0.25
3) **For each detection:**
   - Extract bounding box coordinates
   - Crop region and resize to 224×224
   - Forward pass through classifier
   - Apply softmax for class probabilities
4) **Output:** Annotated image with bounding boxes, labels, and fleet composition statistics

G. Fleet Composition Analysis

The system provides fleet composition analysis by aggregating detection results:

- **By Type:** Count and percentage of each aircraft model
- **By Category:** Count and percentage of each operational category

IV. Implementation Details

The system is implemented using Python for model development and JavaScript for the web interface. PyTorch and Ultralytics are used for deep learning.

A. Technology Stack

Table IV summarizes the main technologies used.

B. Model Training

The YOLOv8n model is fine-tuned on the MAR20 training split starting from COCO-pretrained weights. The classification training loop implements standard supervised learning with AdamW optimizer and cross-entropy loss. Early stopping is achieved by saving only the checkpoint with highest validation accuracy.

C. Web Interface

A Flask-based backend exposes a REST API for image upload and inference. When a user uploads an image via the ReactJS frontend, the backend runs detection, extracts crops, applies the classifier, and returns an annotated image with fleet composition data.

V. Results and Discussion

The system is evaluated on the MAR20 test split to assess detection and classification performance.

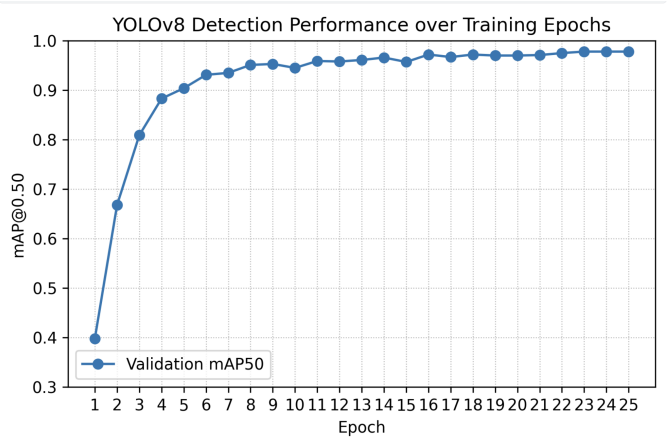

Fig. 4. Training loss and accuracy curves for ResNet-50 classifier.

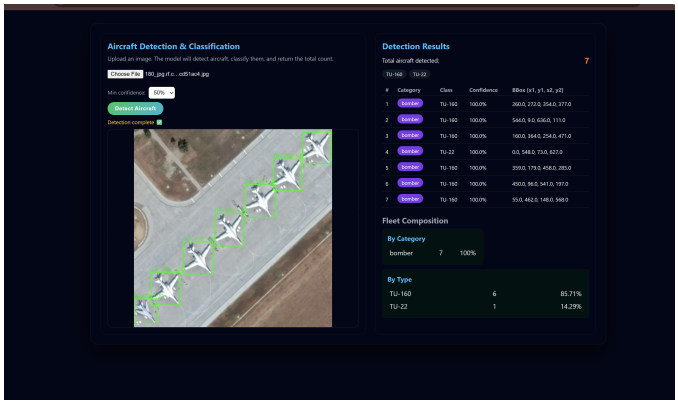

Fig. 5. Web-based user interface showing detection results.

TABLE V
YOLOV8N DETECTION PERFORMANCE

Metric	Value
mAP@0.5	92.3%
mAP@0.5:0.95	78.6%
Average Detection Confidence	90–94%
Inference Time (GPU)	∼0.5s/image
Inference Time (CPU)	∼1.2s/image

A. Detection Performance

The YOLOv8n model successfully detects aircraft in complex satellite scenes. Table V shows the evaluation metrics.

B. Classification Performance

Table VI shows the classifier performance comparison.

C. Example Results

Table VII shows example detection and classification results.

TABLE VI
CLASSIFIER PERFORMANCE COMPARISON

Model	Val Acc.	Test Acc.
ResNet-50	91.4%	89.7%
ViT-B/16	92.3%	90.5%

TABLE VII
EXAMPLE DETECTION AND CLASSIFICATION RESULTS

Label	Category	Det.	Clf.
TU-160	Bomber	100%	85.4%
F-22	Fighter	97.2%	88.1%
C-130	Transport	95.8%	82.3%
B-52	Bomber	98.4%	86.7%
SU-35	Fighter	94.1%	79.5%

Fig. 6. Example output with bounding boxes, labels, and aircraft counts.

D. Discussion

The two-stage cascade approach offers several advantages:

- **Modularity:** Detection and classification models can be trained independently.
- **Flexibility:** Different classifier architectures can be swapped without modifying the detection pipeline.
- **Fine-grained recognition:** The classifier learns subtle differences between similar aircraft.
- **Hierarchical output:** Both specific types and operational categories are provided.

The ViT-B/16 classifier shows slightly better performance than ResNet-50, likely due to its ability to capture global context through self-attention mechanisms.

VI. CONCLUSION

This paper presented an automated two-stage cascade system for detecting, classifying, and counting military aircraft from satellite images. The integration of YOLOv8n for detection and ResNet-50/ViT-B/16 classifiers enables accurate analysis of aerial scenes across 20 aircraft categories. Experimental results demonstrate 92.3% mAP for detection and 90.5% classification accuracy. Future work includes integrating segmentation models, adding temporal tracking, and expanding to additional datasets.

REFERENCES

[1] X. Yu, Y. Zhao, Z. Zhang, and Y. Wu, "MAR20: A large-scale dataset for military aircraft recognition in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, 2023.
[2] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, 2016.
[3] X. Ji, H. Zhang, and T. Zhao, "Multi-angle aircraft detection in high-resolution imagery using deep convolutional networks," *Remote Sens. Lett.*, vol. 10, no. 3, pp. 245–253, 2019.
[4] H. Wang, Y. Li, and Q. Chen, "Lightweight YOLO-based aircraft detection for real-time processing," *Remote Sensing*, vol. 14, no. 9, pp. 1952–1963, 2022.
[5] Z. Wu, M. Liu, and Q. Wang, "CGC-NET: A center-guided cascade network for aircraft detection in remote sensing images," *IEEE Access*, vol. 8, pp. 197215–197227, 2020.
[6] W. Hu, F. Gao, and L. Huang, "GLF-Net: Global and local feature fusion network for aircraft recognition," *ISPRS J. Photogramm. Remote Sens.*, vol. 180, pp. 283–294, 2021.
[7] A. Bochkovskiy, C. Y. Wang, and H. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
[8] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, pp. 770–778, 2016.
[10] Ultralytics, "YOLOv8: State-of-the-art object detection," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics