RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt26321

# Benchmarking YOLO Variants for Vision-Based Flap State Detection in Fixed-Wing UAVs

**V. Srinivasan [a, *], J.V. Sai Prasanna Kumar [a]**

[a] Department of Aeronautical Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Avadi, Chennai – 600062, India.

* Corresponding Author Email: vsrinivasanphd@gmail.com

DOI: https://doi.org/10.54392/irjmt26321

Received: 10-02-2026; Revised: 27-04-2026; Accepted: 11-05-2026; Published: 26-05-2026

**Abstract:** Fixed-wing UAVs depend on their flap state for safe operation, to do that conventional sensors cannot be used such as the potentiometers because they are rigid, weigh so much, and lack adaptation according to the place they are used. This creates a critical gap in the need for a good fault-tolerant method used for monitoring the UAV's control surface. This study aims to benchmark this fault tolerant method, as the fault tolerant method used here will be using Oak D Lite stereo vision camera and the implementation of YOLO model in it, the YOLO model to be used is benchmarked for superior performance, thus establishing a good and lightweight alternative to traditional sensors. A thorough dataset of about 1,920 raw images were taken by mounting the Oak D Lite camera at the tail section of the fixed-wing UAV, it has six flap state images taken under various lighting conditions, depth, and background conditions. After augmentation, 4,623 images are been used to train and then evaluate the YOLOv5(s/m/l), YOLOv8(s/m/l), and YOLO11(s/m/l) models, other models are either too old or too much minimal for this research. These models are evaluated under standard parameters run on an RTX 4090 GPU. Found out that all the models achieved mAP@0.5 exceeding 98%, with a good precision and recall consistency that are above 98.1% and 95.5% respectively. YOLOv5s achieved the best F1-score with the lowest confidence threshold with only 7 misdetections, while other models like YOLOv5s and YOLOv8s had a good inference speed of about 0.6ms. These findings provide a validated baseline benchmark for UAV control surface monitoring and offer practical deployment guidelines for real-time fault-tolerant UAV systems.

**Keywords:** UAV Flap Detection, Fixed-Wing UAV, Deep Learning, Computer Vision, Model Benchmarking, Autonomous Flight Control.

## 1. Introduction

Flap state detection is crucial for fixed wing UAV's safe operation, flaps influence the aerodynamics and performance of an UAV [1]. Accurate detection helps in optimizing lift to drag ratio. Flap state detection helps to improve UAV life by logging the movement data [2] and making timely maintenance. Flap state detection (flap position) sensors are available in commercial flights mainly potentiometers and if a flap fails to operate there are multiple redundancy methods to maintain flight. In case of fixed wing UAV having redundancy methods increase the weight of the UAV. In order to make a fault tolerance setup for UAV, camera is mounted to the tail section to register in real time the movement of flaps. The problem is sometimes lighting conditions, cluttered backgrounds can make the detection accuracy lagging [3]. Studies have shown that integrating deformable convolutional networks into YOLO architectures enhances detection performance under complex visual conditions, improving accuracy for irregularly shaped and small objects [4]. Traditional sensors are rigid, cannot be used in all kinds of places, like in different UAV platforms and cannot be used under different operating conditions. Manual inspections are time consuming and are prone to human errors. Similarly, traditional on-board systems, electronics do not provide accurate detections [5].

Computer vision when combined with Deep Learning (DL) offers robust solution to these kinds of challenges. Object detection frameworks have evolved from two-stage to single-stage detectors, with YOLO leading the latter category by achieving an exceptional trade-off between detection accuracy and inference speed [6]. This progress has established YOLO as the preferred model for real-time applications, including UAV-based visual monitoring. DL frameworks like YOLO helps in providing real time inference with high precision even in cluttered images [7]. YOLO represents a major milestone in single-stage object detection, with continuous innovation, the YOLO models have greatly

improved its detection speed and accuracy with increase in their versions [8]. YOLO models are built to run efficiently, fastly and reliably. When compared with their alternatives like Faster-CNN and SSD (Single Shot MultiBox Detector), YOLO models are better suitable for flap state kind detections. YOLO's single stage detention framework gives way for a fast inference speed without reduction in their accuracy [9]. This kind of speed advantage is required at real time flight, where timely detections of flap states is required to make other algorithms work in ease for fault tolerance. YOLO models help in object detection of small objects at different background and lighting conditions [10]. The correct YOLO model to be used can be hard to find, so in order to avoid that in this research we are training and detecting the flap state by using all the versions of YOLO. From YOLOv5s to YOLO11l, all of these versions have their own advantages and disadvantages. Other studies that have similar applications claim that YOLOv5 has balanced accuracy and computational efficiency [11]. YOLOv8 is said to have introduced architectural innovations that includes a c2f module for gradient flow and decoupled prediction heads, leading to a better accuracy and faster process [12]. The most recent YOLO model YOLO11 has shown that they are lightweight in design and good for deployment [13]. Many papers have comprehensively analysed YOLO's evolution up to YOLOv11, explaining its architectural innovations and effectiveness across many domains [14]. Their insights support the choice of YOLO as a robust and efficient framework for flap state detection in UAVs. Current researches leaves notable gaps. Few studies address UAV flap detection and there are no standard benchmarks for this kind of task. Existing studies are based on general aerial object detection but not on the task like control surface monitoring [15]. Objectives of this paper includes but not limited to: (i) Creating specialized dataset of various flap states, (ii) the dataset consists of variables that help for effective flap state detection training, (iii) train and evaluate the dataset using YOLOv5s, YOLOv5m, YOLOv5l, YOLOv8s, YOLOv8m, YOLOv8l, YOLO11s, YOLO11m, and YOLO 11l, (iv) compare the results and analyze which model will be better suitable for flap state detection based on the available resources like computational power of on-board systems, etc., (v) establish baseline benchmark for future research on UAV control surface monitoring.

## 2. Literature Review

The comparison of previous studies, their methods, their key findings and the available research gap in them are collected and given in table 1.

**Table 1.** Comparison of previous studies, their methods, their key findings, and the research gap found

| Reference | Objective | Method / Model | Dataset | Key Findings | Limitation / Gap |
|---|---|---|---|---|---|
| UAV Aerodynamics & Control Surface Monitoring | | | | | |
| Zhang *et al.* (2024) [1] Sensors | Improve longitudinal drone flight via morphing wings and aerodynamic surface deflection | Coupled wing morphing + aerodynamic surface deflection; experimental flight analysis | Custom fixed-wing UAV flight data | Wing morphing combined with surface deflection significantly improves longitudinal performance and stability | No automated vision-based sensing; monitoring relies on conventional actuation feedback |
| Wang *et al.* (2023) [2] UAV Control | Robust compensation control for fixed-wing UAVs under disturbances | Tube-based robust model predictive control (MPC) | Simulated fixed-wing UAV dynamics | Fast disturbance rejection; stable flight under uncertainty without real-time sensor feedback | Assumes ideal knowledge of flap states; no fault-tolerant or vision-based monitoring |
| Zhou *et al.* (2024) [5] Sensing | Self-powered digital displacement sensor for flight actuation monitoring | Triboelectric nanogenerator-based displacement sensing | Lab-scale actuator bench tests | Accurate digital displacement measurement without external power; suitable for actuator monitoring | Rigid, hardware-specific; limited cross-platform adaptability; not vision-based |

| | | | | | |
|---|---|---|---|---|---|
| YOLO Object Detection — Evolution & Benchmarks | | | | | |
| Diwan *et, al.* (2022) [6] YOLO Survey | Survey of YOLO architectures — challenges, successors, datasets, applications | Literature review covering YOLOv1–v7; architectural analysis | Multiple public benchmark datasets (COCO, VOC) | YOLO outperforms two-stage detectors for real-time tasks; each version addresses prior accuracy–speed trade-offs | No domain-specific benchmarks for UAV control surface monitoring |
| Chen (2024) [8] YOLO Survey | Research overview of YOLO-series algorithms based on deep learning | Systematic review of YOLO variants through YOLOv8 | Standard academic benchmarks | YOLO's c2f module and decoupled heads in v8 improve gradient flow and accuracy; strong across diverse tasks | Limited to general object detection; no application to UAV flight control surfaces |
| Ali & Zhang (2024) [14] YOLO Survey | Comprehensive review of YOLO evolution up to YOLOv11, benchmarks, and applications | Structured literature review + benchmark comparison | COCO, VisDrone, DOTA and others | YOLO11 achieves lightweight design and strong deployment efficiency; attention mechanisms improve small-object detection | No evaluation on specialised control surface datasets; gap in safety-critical UAV monitoring remains |
| UAV Aerial Object Detection | | | | | |
| Baidya & Jeong (2022) [3] Vision | Precise object detection in drone imagery using YOLOv5 + ConvMixer heads | YOLOv5 with ConvMixer prediction heads | VisDrone; custom drone imagery | ConvMixer heads improve small object detection in drone imagery; better spatial representation than standard heads | Focuses on general aerial objects, not UAV-specific control surface states |
| Rahman *et al.* (2024) [7] UAV | Survey of UAV detection and classification using ML approaches | Review covering CNN, YOLO, SSD, Faster R-CNN for UAV detection | Multiple aerial datasets | DL-based detectors outperform traditional methods; YOLO series preferred for real-time tasks | Survey perspective; no flap-state-specific dataset or benchmark provided |
| Loi *et al.* (2023) [11] YOLO | Real-time UAV-based vehicle detection; comparative evaluation of YOLO models | YOLOv5, YOLOv7, YOLOv8 compared on aerial vehicle detection | Custom UAV-captured vehicle dataset | YOLOv8 achieves higher mAP; YOLOv5 faster inference; no single model dominates all metrics | Domain limited to vehicle detection; no evaluation on UAV structural components |
| Triska *et al.* (2024) [10] Vision | Weed detection in strawberry fields using UAV monocular and stereo vision + ML | YOLOv8; monocular vs stereo camera comparison | Custom UAV field imagery | Stereo vision improves detection accuracy; depth perception aids localization in cluttered scenes | Agricultural domain only; no control surface monitoring application |

| | | | | | |
|---|---|---|---|---|---|
| Specialised Detection Tasks (Defect, Crack, Pavement) | | | | | |
| Guo (2025) [4] Vision | Improve detection for irregularly shaped and small objects using deformable convolutions | Deformable convolutional networks integrated into YOLO | General object detection benchmarks | Deformable convolutions enhance detection of irregular geometries under complex visual conditions | Not tested on UAV control surfaces; no real-time inference analysis for on-board deployment |
| Liu *et al.* (2022) [9] Vision | Defect detection for TO-Can packaged laser diodes using improved YOLO | Improved YOLO with modified anchor settings and augmentation | Custom industrial defect images | High-precision defect detection for small industrial components; effective under controlled lighting | Industrial setting only; no outdoor, multi-background, or variable lighting conditions |
| Chen *et al.* (2024) [12] UAV | Pavement crack detection and evaluation using UAV + deep learning | YOLOv8-based framework for crack segmentation from UAV images | UAV-captured pavement images | Deep learning enables automated crack detection from UAV imagery with high precision | Task-specific to infrastructure inspection; no onboard real-time control surface monitoring |
| Peng *et al.* (2022) [15] UAV | Vehicle recognition from UAV video using target pre-detection + deep learning fusion | Pre-detection stage fused with deep learning classifier | UAV aerial video dataset | Fusion of pre-detection with DL improves recognition accuracy over single-stage methods | Limited to vehicle recognition; no fault-detection or control surface monitoring |
| Research Gap: It is found that there are no previous studies regarding the study of various YOLO models used in the flap state detections are available nor there lies a standard dataset on the control surface, especially flap states for the use in UAV fault tolerance. | | | | | |
| Present Work Srinivasan & Sai Prasanna (2025) [22] | Benchmark YOLOv5/v8/11 variants for UAV flap state detection; establish validated baseline for control surface monitoring | YOLOv5(s/m/l), YOLOv8(s/m/l), YOLO11(s/m/l) trained on custom flap dataset; OAK D Lite stereo camera; RTX 4090; Ultralytics framework | 1,920 raw → 4,623 augmented images; 6 flap classes (LFD, LFN, LFU, RFD, RFN, RFU); varied lighting, depth, and background | All models exceed 98% mAP@0.5; YOLOv5s: best F1 & accuracy (7 misdetections, threshold 0.619, 0.6 ms inference); YOLOv5m: best deployment balance | — |

## 3. Methodology

The overall method used in this research is given as flowchart as shown in figure 1.

### 3.1. Research Framework

This study follows an experimental pipeline that is designed to evaluate the tendency of YOLO-based deep learning architectures in the process of detection of UAV flap states from images. The research process have five phases,

1) hardware setup and image acquisition,
2) dataset creation,
3) dataset annotations and preprocessing,
4) model training using standard parameters,
5) Performance evaluation using standard metrics.

This approach helps in reproducibility and easy comparability for all the nine models.

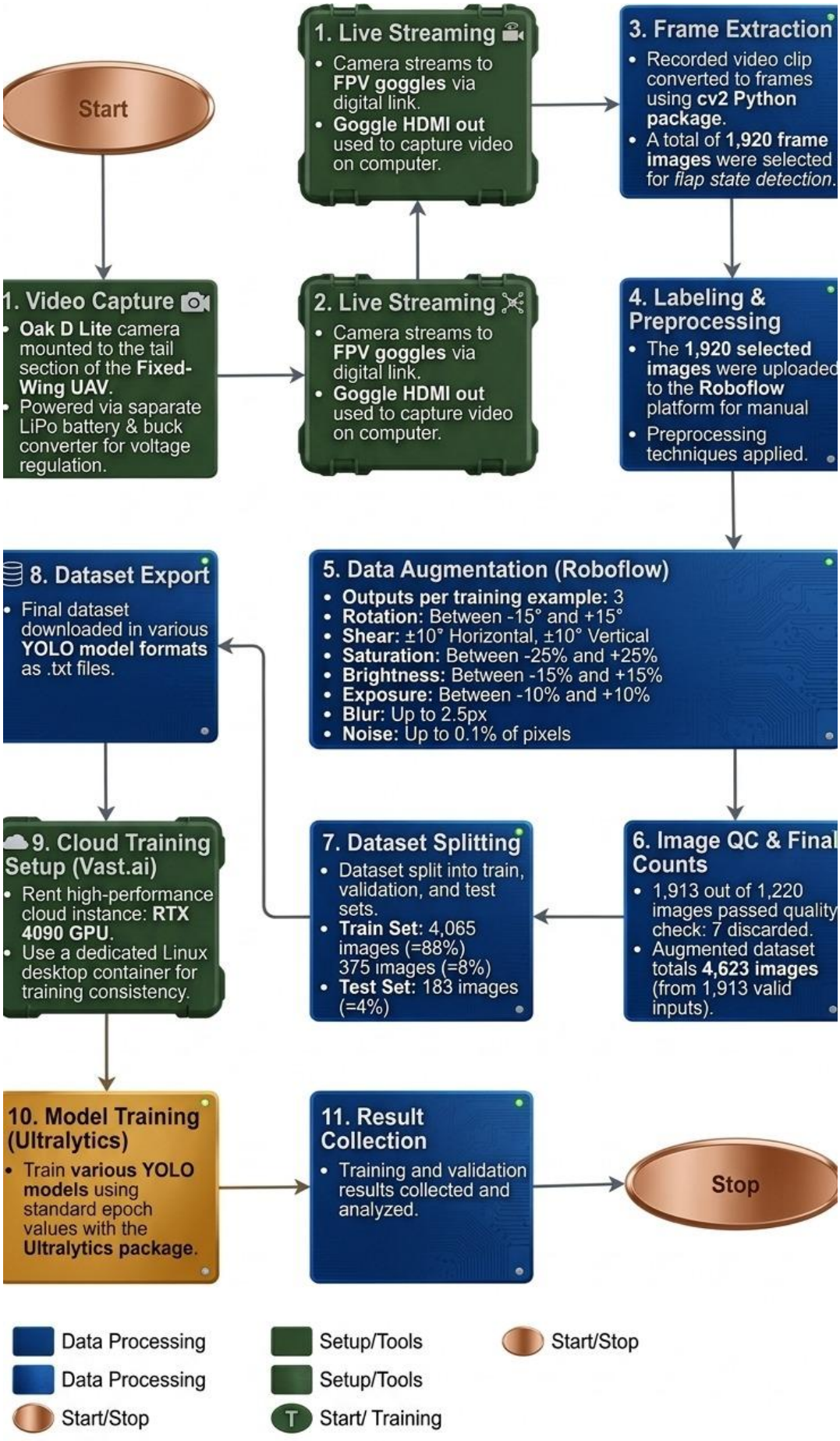


**Figure 1.** This flowchart explains the overall workflow of this research

**Table 2.** Dataset Design for UAV Flap Detection

| Factor | Levels | Images Produced |
|---|---|---|
| Baseline (Flap × Quadrant States) | ~32 unique combinations | 32 |
| Depth Variation (-z values) | 10 distinct depth values per baseline capture | 32 × 10 = 320 |
| Lighting Conditions | 2 conditions: Normal light, Dark light | 320 × 2 = 640 |
| Background Type | 3 backgrounds: Sky (default), Ground, Clutter | 640 × 3 = 1920 |

### 3.2. Hardware Setup

First of all the Oak D Lite Camera is mounted to the tail section of the UAV as shown in figure 2.

This is used because it helps to capture the full wingspan of the UAV, once the wingspan is covered, video is taken by varying flap states during flight. The camera's video is streamed to FPV goggles from there the video is run in a computer by the use of HDMI out port in the FPV goggle. The video is recorded and then turned into frames/images, selection of images is done at various places, depths and the process is outlined as shown in table 2.

### 3.2. The Dataset

The dataset contains several images that are having variable environmental conditions. Initially, 32 unique combinations of images are taken that have various flap states (flaps up, flaps down, flaps neutral) this combined with the differentiation between right and left flaps, further multiplied by quadrant states as shown in figure 2, forms the baseline dataset (32 images). For each of these baseline captures, 10 distinct depth variations along the z-axis were introduced, resulting in 320 images.

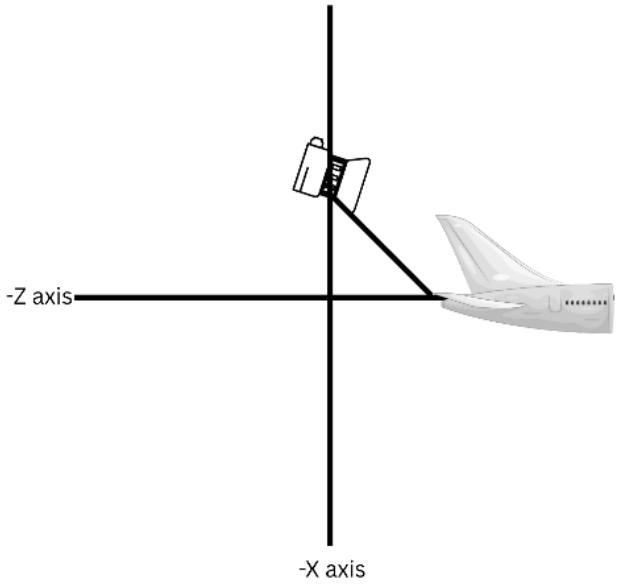


**Figure 2.** Representative Camera Positions for Dataset Creation

To account for illumination changes, data was further collected under two lighting conditions (normal light and dark light), doubling the dataset to 640 images. Finally, environmental variability was introduced by capturing images across three different background types: sky (default), ground, and clutter. This expanded the dataset to a total of 1920 images. When feeding this to roboflow website, after labelling augmentation is done to those images as given in table 3.

**Table 3.** Augmentation Details

| | |
|---|---|
| Outputs per training example | 3 |
| Rotation | Between -15° and +15° |
| Shear | ±10° Horizontal, ±10° Vertical |
| Saturation | Between -25% and +25% |
| Brightness | Between -15% and +15% |
| Exposure | Between -10% and +10% |
| Blur | Up to 2.5px |
| Noise | Up to 0.1% of pixels |

The images were annotated with six different classes corresponding to the flap position with respect to the wing span, they are Left Flap Down(LFD), Left Flap Up(LFU), Left Flap Neutral(LFN), Right Flap Down(RFD), Right Flap Up(RFU) and Right Flap Neutral(RFN), and by default background class is taken by the YOLO model. Some example images are given in figure 3. These classes are labelled using bounding boxes with separate colors in order to provide differences in the classes. The dataset was chosen for DL the flap states on the control surface in order to do flap state detection, because of its huge amount of image sets and different sample size, thereby aids in providing a depiction of flap position and movements with respect to the wing span. Table 4 gives the detailed outlining the number of raw images, augmented images, discarded images, and final images used in each split.

The labelled images was split between 88% for training, 8% for validating the images, and 4% for testing the image sets, as shown in table 5. Effective training was made possible by this distribution, which will help to get a separate representation for each class in the images. Still, flap state detection may differ in practical applications, which could make precise detection and categorization difficult.

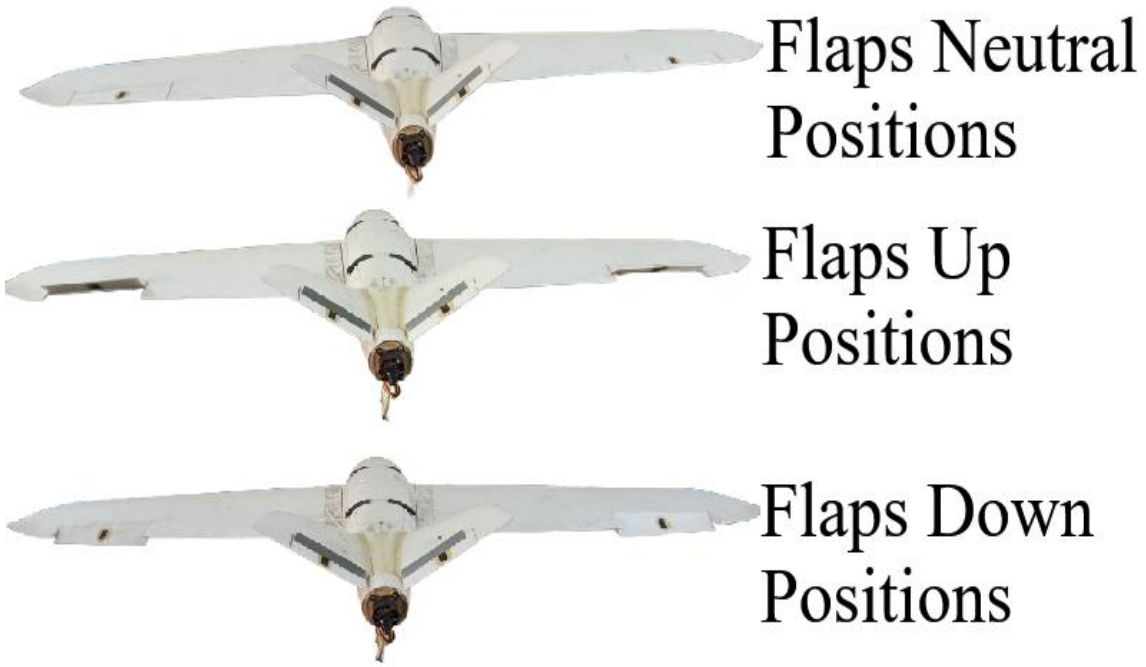


**Figure 3.** Representative Flap Positions for Dataset Creation

**Table 4.** Detailed table outlining the number of raw images, augmented images, discarded images, and final images used in each split

| Raw Images | Augmented Images | Discarded Images | Final Images | Train Set | Valid Set | Test Set |
|---|---|---|---|---|---|---|
| 1920 | 4623 | 7 | 4623 | 4065 | 375 | 183 |

**Table 5.** Dataset distribution for fixed wing UAV flap detection

| Dataset | Number of images | Percentage |
|---|---|---|
| Training | 4065 | 88% |
| Validation | 375 | 8% |
| Testing | 183 | 4% |
| **Total** | 4623 | 100% |

Nevertheless, the dataset's varied samples from a range of environmental situations improved the model's learning and generalization, resulting in a good detection performance across the flap's various states.

The training procedure was streamlined by the manual annotation of the dataset. The labelled images were good enough to enable efficient model learning thanks to additional preprocessing and augmentation techniques like auto orientation, a default photo size of 640x640, auto-adjust contrast, rotation between -15 degrees and +15 degrees, shear of ≥10 degrees horizontal and ≥10 degrees vertical, saturation levels between 25% increased saturation and 25 % decreased saturation, brightness levels between 15% increased brightness and 25 % decreased brightness, exposure levels between 10% decreased exposure level and 10% increased exposure levels, blur levels till 2.5 pixels, and noise levels till 0.1% of pixels. The YOLO models, YOLOv5, YOLOv8, and YOLOv11, are able to perform strong detection at various flap positions because of the dataset variability.

### 3.3 Model Training

The model training was conducted using YOLOv5(s,m,l), YOLOv8(s,m,l) and YOLO11(s,m,l) models. The workstation used was Xenon E5-2680 v4 with 28CPU, 64GB RAM, CT2000P3PSSD8 1TB, 24GB RTX 4090 GPU, Max CUDA 12.6. During training, the models do several iterations, targeted at identifying objects of different categories of classes.

**Table 6.** Parameters that are used to train the models.

| Hyperparameter | Value | Description |
|---|---|---|
| Epochs | 100 | Total number of times the training dataset was run. |
| Batch Size | 16 | Number of samples that are processed prior to the model's parameter adjustments. |
| Image Size (imgsz) | 640 | The dimension to which all training input images are shrunk in order to balance computational expense and accuracy. |
| Initial Learning Rate (lr0) | 0.01 | The initial learning rate, which establishes the optimizer update step size. |
| Final Learning Rate (lrf) | 0.01 | In order to guarantee progressive convergence, the learning rate was applied in the last epoch. |
| Warmup Epochs | 3.0 | The number of first epochs in which training is stabilized by gradually increasing the learning rate. |
| Momentum | 0.937 | The optimizer's hyperparameter that speeds up convergence and smoothes weight updates. |
| Weight Decay | 0.0005 | A regularization parameter was included to lessen overfitting in the model. |
| Box Loss Gain (box) | 7.5 | The localization accuracy is prioritized by applying a multiplier to the bounding box regression loss. |
| Class Loss Gain (cls) | 0.5 | During training, a multiplier is applied to the detection loss to modify its contribution. |
| Definition Loss Gain (dfl) | 1.5 | Adjusting for focus loss, which improves bounding box prediction accuracy. |

The training employs a set of pre-trained weights provided by ultralytics in order to accelerate the picture recognition and its precision as well.

The YOLO models were all trained with one parameter that is standardized to ensure there would be consistency and comparability across all the experiments. Table 6 shows the parameter setting. This training took 100 epochs, and thus presented sufficient number of iterations of the models to learn well in the variabilities of the data in the dataset. A batch size of 16 was employed to have the most desirable trade-off between memory efficiency and speed of computation. Input photos were downsampled to 640 × 640 in order to bring all the models into the same level of data. In order to guarantee controlled changes in the learning process, both the beginning and final learning rates were set to 0.01 during the training. The momentum was 0.937 to stabilize the parameter updates, and the weight decay rate of 0.0005 was included to reduce overfitting. To enable the learning parameters to be steadily stabilized, three warming epochs were included during the training process.

## 3.4. Coding Brief

The coding part for training involves the usage of the Ultralytics python package as it is having all the parameters standardized for object detection, the roboflow python package was used to import the dataset, since the labelling part was much easier and hassle free using the website. Bounding box-based labelling was used to match the position of the flaps on the overall fixed wing UAV. The training was done on the standard RTX4090 GPU, and that was installed with Linux and Jupyter notebook docker container. The code used was having epochs number i.e., epochs=100 and the image size, i.e., imgsz=640, other parameters were predefined by the ultralytics package, and the parameters are given in the table 6. By varying the model for the train command, the values for each model are taken. The exact code used for training is:

yolo task=detect mode=train model=yolov8m.pt data={dataset.location}/data.yaml epochs=100 imgsz=640

And the exact software versions are Ultralytics 8.3.220, Python-3.12.3, torch-2.9.0+cu128, NVIDIA GeForce RTX 4090, 24109MiB.

## 3.5. Evaluation Metrics

The evaluation metrics employed in this study are Intersection over Union (IoU), Precision, Recall and mean Average Precision (mAP) were selected since they collectively provide a comprehensive assessment of the detection capability of the model.

Intersection over the Union (IoU) thresholds:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \quad (1)$$

The fundamental criteria for determining whether a predicted bounding box sufficiently overlaps with the ground truth annotation is served by the IoU.

This metric is important because detection of flaps in the control surfaces is required.

Precision:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positive\ (FP)} \quad (2)$$

Precision metric helps to evaluate the reliability of detections by measuring the number of true positives in all of the predicted positives.

This reduces the false detections where background items could be incorrectly identified as flaps in the context of UAV flap detection.

Recall:

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP)+False\ Negatives(FN)} \quad (3)$$

Recall metric helps in measuring the ability of the model to capture all correct instances thereby making sure of correct flap position detection.

Since missed detections can compromise the detection of flaps, this is very important for reducing the UAV flap state detection count

Mean Average Precision:

$$mAP = \left(\frac{1}{N}\right) \times \sum_{i=1}^{N}(AP_i) \quad (4)$$

Mean Average Precision metric involves both precision and recall creating a balanced measurement for the overall detection performance.

It gives a robust single metric to evaluate the model under various factors.

These metrics therefore ensure that the evaluation captures not only the accuracy of bounding box localization but also the robustness of detection across diverse conditions like lighting variations, depth changes and background complexity.

## 4. Results and Discussions

A full-fledged understanding of the precision, functionality and efficiency of these cutting edge object detection models is provided by the comparison of YOLOv5, YOLOv8 and YOLOv11 for flap state detection. The results that include precision, recall and mAP for the flap states are shown in figure 6.

These results are explained methodologically in the following sections, giving a full-fledged explanation on the strength and weaknesses of each model [16]. Thus, this paper provides a detailed insight on the powerfulness of these versions of YOLO in the region of flap state detection of fixed wing UAV by analysing the detection precision, reliability and preprocessing efficiency. The table 7 provides the comparison metrics acquired for each YOLO model.

### 4.1. Performance Evaluation: Precision, Recall, Patterns of errors and analysis of confidence.

All the models remarkably achieved high detection performance, considering all six flap position classes with mAP @ 0.5 values consistently exceeded 98%. This might be good, but it is only helpful if there is any meaningful difference between YOLOv5m and YOLO11s as two performing models in precision range 99.1% and 99.2%, similarly recalls in range 99.7% and 99.6% give a deep interpretation architecturally. YOLO11s benefits from refined neck structure and improved pyramid aggregation introduced in the later YOLO model generations, which thus tend to reduce false negatives on small or partially occluded objects similar to the UAV inspection task. The marginal superiority of YOLO11s over YOLOv5m in precision likely reflects these structural refinements than a fundamental increase in detection capacity, suggesting lowered returns as the architecture evolves.

The Confucian Matrix Analysis (Figure 4) reveals more than aggregate mAP values alone. Even though YOLOv5S and YOLO11L produced fewest misdetections (7 each) the YOLOv5M and YOLOv8s produced 15 misdetections comparatively, as given in Table 8. These differences are to be taken seriously. In a safety-critical application such as flap state detection, even seven misdetections are a critical thing. The tendency of YOLOv5S toward lower confusion is consistent with the findings in the broader literature review. However, attributing this image solely to model size would be an over-simplification. Dataset class balance, augmentation strategy, and the visual similarity between adjacent flap states (e.g., Neutral with a slight flap up and down) are likely contributing factors that lead to a controlled investigation.

The conclusion that the YOLO family is a mature and reliable architecture for this class of UAV inspection task is supported by the near ceiling performance across all models. Nevertheless, the marginal performance differences between variants carry real engineering tradeoffs: YOLOv5s and YOLO11s offer compelling accuracy to computational cost ratios that favour onboard, edge deployment scenarios where inference latency and power consumption are constrained, whereas larger variants such as YOLO11l may be better suited to ground station processing pipelines where computational overhead is less critical.

Several limitations of the present study should be acknowledged.

**Table 7.** Comparison based on YOLO model

| Class | Image | Instances | Model | Precision (%) | Recall (%) | mAP@0.5(%) |
|---|---|---|---|---|---|---|
| all | 375 | 748 | YOLOv5s | 0.992 | 0.996 | 0.993 |
| | 375 | 748 | YOLOv5m | 0.991 | 0.997 | 0.994 |
| | 375 | 748 | YOLOv5l | 0.991 | 0.996 | 0.991 |
| | 375 | 748 | YOLOv8s | 0.991 | 0.995 | 0.991 |
| | 375 | 748 | YOLOv8m | 0.99 | 0.995 | 0.992 |
| | 375 | 748 | YOLOv8l | 0.99 | 0.991 | 0.994 |
| | 375 | 748 | YOLO11s | 0.992 | 0.996 | 0.994 |
| | 375 | 748 | YOLO11m | 0.992 | 0.993 | 0.994 |
| | 375 | 748 | YOLO11l | 0.992 | 0.995 | 0.994 |
| Left Flap Down | 92 | 92 | YOLOv5s | 0.994 | 1 | 0.995 |
| | 92 | 92 | YOLOv5m | 0.997 | 1 | 0.995 |
| | 92 | 92 | YOLOv5l | 0.986 | 1 | 0.994 |
| | 92 | 92 | YOLOv8s | 0.998 | 1 | 0.995 |
| | 92 | 92 | YOLOv8m | 0.997 | 1 | 0.995 |
| | 92 | 92 | YOLOv8l | 0.992 | 1 | 0.995 |
| | 92 | 92 | YOLO11s | 0.997 | 1 | 0.995 |
| | 92 | 92 | YOLO11m | 0.998 | 1 | 0.995 |
| | 92 | 92 | YOLO11l | 0.997 | 1 | 0.995 |
| Left Flap Neutral | 108 | 108 | YOLOv5s | 0.999 | 1 | 0.995 |
| | 108 | 108 | YOLOv5m | 0.989 | 1 | 0.995 |
| | 108 | 108 | YOLOv5l | 0.999 | 1 | 0.995 |
| | 108 | 108 | YOLOv8s | 0.998 | 1 | 0.995 |
| | 108 | 108 | YOLOv8m | 0.991 | 0.999 | 0.995 |
| | 108 | 108 | YOLOv8l | 0.982 | 1 | 0.995 |
| | 108 | 108 | YOLO11s | 0.999 | 1 | 0.995 |
| | 108 | 108 | YOLO11m | 0.987 | 0.991 | 0.995 |
| | 108 | 108 | YOLO11l | 1 | 0.993 | 0.995 |
| Left Flap Up | 173 | 173 | YOLOv5s | 0.987 | 1 | 0.993 |
| | 173 | 173 | YOLOv5m | 0.986 | 1 | 0.994 |
| | 173 | 173 | YOLOv5l | 0.987 | 1 | 0.988 |
| | 173 | 173 | YOLOv8s | 0.988 | 0.992 | 0.992 |
| | 173 | 173 | YOLOv8m | 0.983 | 1 | 0.993 |
| | 173 | 173 | YOLOv8l | 0.982 | 1 | 0.993 |
| | 173 | 173 | YOLO11s | 0.987 | 1 | 0.993 |
| | 173 | 173 | YOLO11m | 0.988 | 0.991 | 0.993 |
| | 173 | 173 | YOLO11l | 0.983 | 1 | 0.994 |
| Right Flap Down | 105 | 105 | YOLOv5s | 0.988 | 1 | 0.993 |
| | 105 | 105 | YOLOv5m | 0.988 | 1 | 0.995 |
| | 105 | 105 | YOLOv5l | 0.981 | 0.999 | 0.995 |
| | 105 | 105 | YOLOv8s | 0.989 | 1 | 0.994 |
| | 105 | 105 | YOLOv8m | 0.989 | 1 | 0.995 |
| | 105 | 105 | YOLOv8l | 0.99 | 1 | 0.995 |
| | 105 | 105 | YOLO11s | 0.988 | 1 | 0.994 |
| | 105 | 105 | YOLO11m | 0.989 | 1 | 0.995 |
| | 105 | 105 | YOLO11l | 0.988 | 1 | 0.995 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Right Flap Neutral | 117 | 118 | YOLOv5s | 0.99 | 0.983 | 0.99 |
| | 117 | 118 | YOLOv5m | 0.99 | 0.983 | 0.991 |
| | 117 | 118 | YOLOv5l | 0.991 | 0.983 | 0.982 |
| | 117 | 118 | YOLOv8s | 0.981 | 0.983 | 0.98 |
| | 117 | 118 | YOLOv8m | 0.988 | 0.975 | 0.982 |
| | 117 | 118 | YOLOv8l | 1 | 0.955 | 0.993 |
| | 117 | 118 | YOLO11s | 0.991 | 0.983 | 0.992 |
| | 117 | 118 | YOLO11m | 0.997 | 0.975 | 0.989 |
| | 117 | 118 | YOLO11l | 0.99 | 0.983 | 0.987 |
| Right Flap Up | 152 | 152 | YOLOv5s | 0.992 | 0.993 | 0.994 |
| | 152 | 152 | YOLOv5m | 0.993 | 0.997 | 0.994 |
| | 152 | 152 | YOLOv5l | 0.993 | 0.992 | 0.991 |
| | 152 | 152 | YOLOv8s | 0.993 | 0.992 | 0.993 |
| | 152 | 152 | YOLOv8m | 0.991 | 0.993 | 0.994 |
| | 152 | 152 | YOLOv8l | 0.993 | 0.991 | 0.993 |
| | 152 | 152 | YOLO11s | 0.99 | 0.993 | 0.992 |
| | 152 | 152 | YOLO11m | 0.992 | 1 | 0.995 |
| | 152 | 152 | YOLO11l | 0.993 | 0.992 | 0.995 |

**Note:** Sometimes it is possible that reading the results of experiment can give impossible findings likewise once the results of right flap neutral wear checked it felt impossible that multiple occurrences of the same class has appeared but after sometime it was found that more than two flaps got detected because there was a glass pane next to the UAV where the camera even detected that image and marked an annotation on it, this explains the yield of RFN from 117 images to 118 instances.

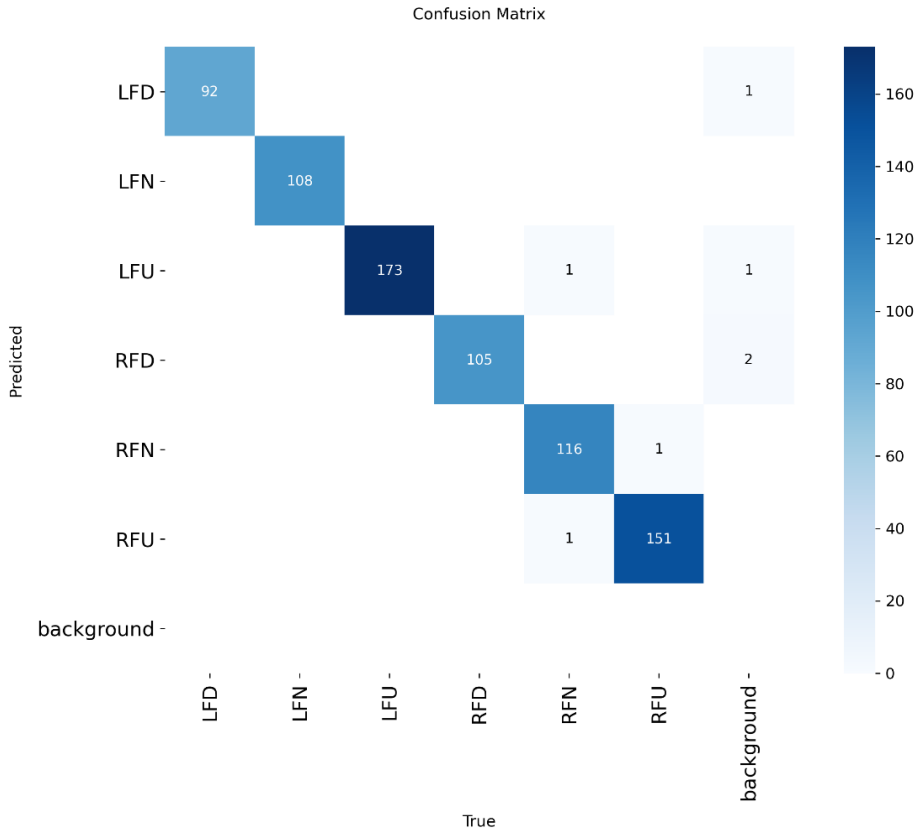


**Figure 4.** Confusion matrix of YOLOv5s model

**Table 8.** Total Misdetections across each YOLO model taken from confusion matrix

| | LFD | LFN | LFU | RFD | RFN | RFU | Background | Total |
|---|---|---|---|---|---|---|---|---|
| YOLOv5s | 1 | 0 | 2 | 2 | 1 | 1 | 0 | 7 |
| YOLOv5m | 0 | 5 | 3 | 1 | 5 | 4 | 0 | 18 |
| YOLOv5l | 1 | 1 | 2 | 2 | 3 | 1 | 0 | 10 |
| YOLOv8s | 0 | 4 | 3 | 1 | 7 | 3 | 0 | 18 |
| YOLOv8m | 0 | 1 | 4 | 1 | 5 | 3 | 0 | 14 |
| YOLOv8l | 1 | 7 | 3 | 2 | 2 | 2 | 0 | 17 |
| YOLO11s | 0 | 1 | 2 | 1 | 2 | 4 | 0 | 10 |
| YOLO11m | 0 | 3 | 4 | 1 | 5 | 3 | 0 | 16 |
| YOLO11l | 0 | 0 | 3 | 1 | 2 | 1 | 0 | 7 |

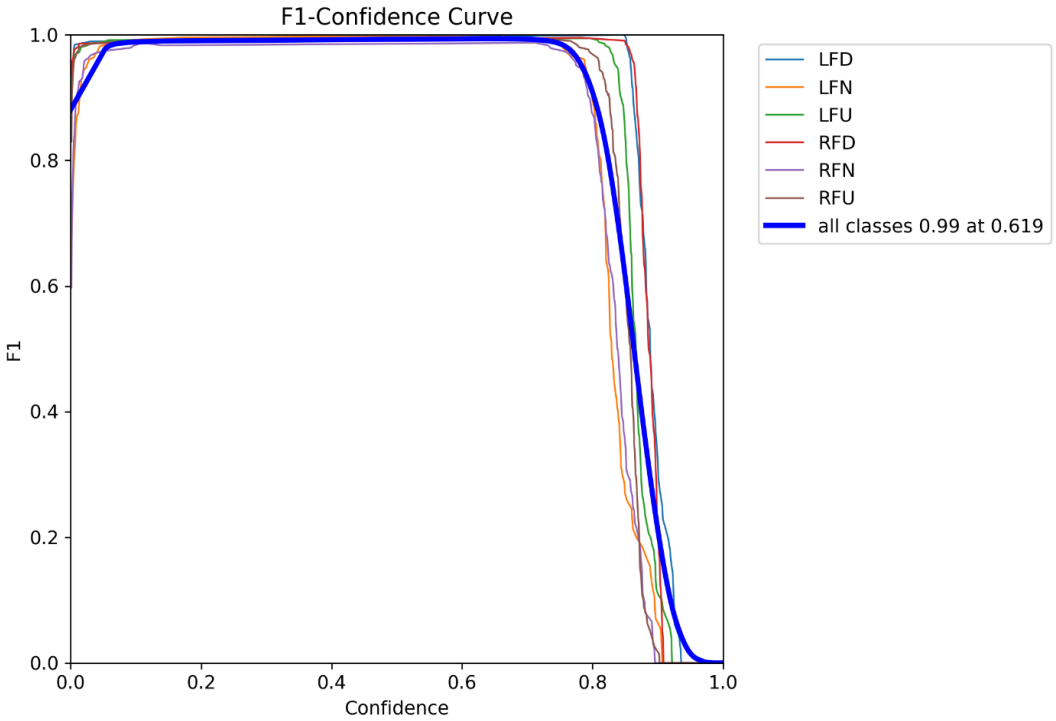


**Figure 5.** F1-confidence curve of YOLOv5s model

**Table 9.** Threshold cutoff of all models after crossing F1 Score of 0.99 in the increasing order

| | Threshold | F1-Score |
|---|---|---|
| YOLOv5s | 0.619 | 0.99 |
| YOLOv5l | 0.626 | 0.99 |
| YOLO11s | 0.634 | 0.99 |
| YOLO11l | 0.634 | 0.99 |
| YOLOv8l | 0.643 | 0.99 |
| YOLOv5m | 0.658 | 0.99 |
| YOLOv8m | 0.661 | 0.99 |
| YOLOv8s | 0.719 | 0.99 |
| YOLO11m | 0.722 | 0.99 |

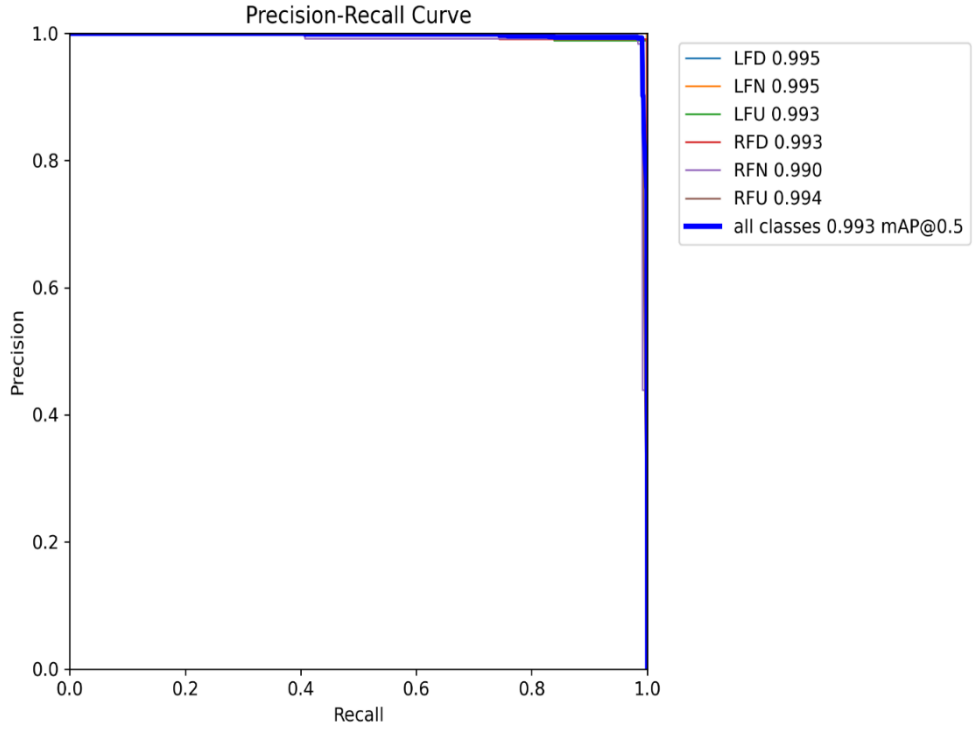


**Figure 6.** Precision-recall (PR) curve of YOLOv5s model

**Table 10.** Precision-Recall values for all models in the increasing order

| | Precision-Recall |
|---|---|
| YOLOv5l | 0.991 |
| YOLOv8s | 0.991 |
| YOLOv8m | 0.992 |
| YOLOv5s | 0.993 |
| YOLOv5m | 0.994 |
| YOLOv8l | 0.994 |
| YOLO11s | 0.994 |
| YOLO11m | 0.994 |
| YOLO11l | 0.994 |

All evaluations were conducted under controlled conditions like performance under real world variables, wind induced vibration, changing illumination and sensor noise remains untested. Although the dataset is carefully annotated. It represents a specific UAV flap geometry and generalisation to other aircraft configurations is not guaranteed. The study also does not report inference speed or memory footprint metrics which are necessary for a complete model selection decision in embedded systems contexts. Future work should address these gaps through field validation trials and a multi criteria evaluation framework that jointly optimises detection accuracy, computational efficiency and robustness to environmental perturbation.

Table 8 reveals that misdetections are not uniformly distributed across all the classes. The right flap neutral (RFN) class consistently shows lower recall compared to that of other classes, indicating a class-specific difference. This could be due to geometry symmetry between neutral and adjacent flap states up and down, which explains the situation. Additionally, minor flap detection and clear annotation uncertainty lead to further confusion. False positives are observed when slightly deflected flaps are being classified as neutral, while false negatives occur due to lighting and cluttered backgrounds. From an operational perspective, misdetections between flap down and neutral are more critical than confusion between neutral and up states. Particularly during take-off and landing phases. However, error rates remain below 1%. Still, there is room for threshold tuning and redundancy elimination.

The F1–confidence curves as given in figure 5, provide a deeper understanding of the comparative performance and stability of the evaluated YOLO models across varying confidence thresholds. Across all nine YOLO variants, the F1-confidence curves exhibit

consistently high detection stability for all six flap position classes, Left Flap Down (LFD), Left Flap Neutral (LFN), Left Flap Up (LFU), Right Flap Down (RFD), Right Flap Neutral (RFN), and Right Flap Up (RFU). Where the YOLOv5s model used a confidence threshold of 0.619, for F1-score of 0.99, while YOLO11m had cutoff threshold only at 0.722 as shown in table 9, thus the YOLOv5s performed better, it was best calibrated for the flap state detection, strong detections even at low confidence, more stable and reliable.

The Precision-Recall curves as given in figure 6, provides the minimal confusion occurring in the model. The higher the value the lower the confusion. YOLOv5m, YOLOv8l, YOLO11s, YOLO11m, and YOLO11l all gave a good precision-recall value of 0.994 as shown in table 10, while YOLOv5l and YOLOv8s had the least value of 0.991 which is not bad for its parameter size [17, 18]. Overall, the PR curve analysis confirms that all YOLO models achieve near-perfect detection capability, with minimal performance gaps among them. From an application standpoint, YOLOv5m proves to be the most consistent and reliable model overall, making it ideal for critical flight control systems where safety and accuracy are paramount.

### 4.2. Computational Efficiency Analysis: Image Processing Speed

Inference Speed Performance: YOLOv5s model and YOLOv8s model demonstrated superior inference speed achieving remarkable 0.6ms inference time, followed by YOLO11s at 0.7ms. YOLOv8l models exhibited higher inference time of 2.6ms. The total processing time per image of the preprocessing and postprocessing takes between 0.4ms (YOLOv8m) to 3.7ms (YOLO11s). These are given in table 11.

**Table 11.** Summary of all the YOLO models and their values

| Model | Layers | Parameters | GFLOPs | Speed (ms/image) |
|---|---|---|---|---|
| YOLOv5s | 84 | 91,13,858 | 23.8 | Preprocess:0.1ms Inference:0.6ms Loss:0.0ms Postprocess:1.1ms |
| YOLOv5m | 106 | 2,50,48,690 | 64.0 | Preprocess:0.1ms Inference:1.7ms Loss:0.0ms Postprocess:0.7ms |
| YOLOv5l | 128 | 5,31,36,034 | 134.7 | Preprocess:0.1ms Inference:2.2ms Loss:0.0ms Postprocess:1.0ms |
| YOLOv8s | 72 | 1,11,27,906 | 28.4 | Preprocess:0.1ms Inference:0.6ms Loss:0.0ms Postprocess:2.1ms |
| YOLOv8m | 92 | 2,58,43,234 | 78.7 | Preprocess:0.1ms Inference:1.6ms Loss:0.0ms Postprocess:0.3ms |
| YOLOv8l | 112 | 4,36,11,234 | 164.8 | Preprocess:0.1ms Inference:2.6ms Loss:0.0ms Postprocess:0.5ms |
| YOLO11s | 100 | 94,15,122 | 21.3 | Preprocess:0.1ms Inference:0.7ms Loss:0.0ms Postprocess:3.6ms |
| YOLO11m | 125 | 2,00,34,658 | 67.7 | Preprocess:0.1ms Inference:1.6ms Loss:0.0ms Postprocess:0.7ms |
| YOLO11l | 190 | 2,52,83,938 | 86.6 | Preprocess:0.1ms Inference:2.2ms Loss:0.0ms Postprocess:0.6ms |

**Table 12.** Category based conclusion as a single framework

| Category | Model |
|---|---|
| **Speed leader** | YOLOv5s / YOLOv8s |
| **Accuracy leader** | YOLOv5s |
| **Best balance (deployment)** | YOLOv5m |


**Figure 7.** Flap state predicted using YOLOv5m, using stereo vision camera for depth perception.

Efficiency Analysis: YOLOv5l architecture is highly optimized in terms of efficiency, with a much faster inference speed despite the number of parameters being a few times more than the corresponding YOLOv8l variants of this architecture [19]. YOLOv5l is the most efficient in terms of parameter usage (minimum of 53.13M parameters) and also has a competitive inference speed (2.2ms). The least parameter usage is done by YOLOv5s (minimum 9.11M parameters) [20] but has a good inference speed of 0.6ms. Preprocessing overhead is also low in all models (0.1ms) and postprocessing is a significant variation between 0.3ms (YOLOv8m) and 3.6ms (YOLO11s).

Resource-Performance Trade-offs: The findings indicate that there are different optimization philosophies, YOLOv5m is focused on accuracy by adding more model depth and parameters but sacrifices inference speed, it is the best in terms of the balance between speed and accuracy due to the architectural innovations, and is more focused on parameters efficiency with offering competitive performance. To obtain real-time UAV results [21] and demand sub-3ms processing times, the YOLO11l variant can be used to obtain high accuracy at the cost of speed.

In order to relate model performance to practical UAV operations, system-level acceptance criteria are in place [22]. For real-time flap monitoring, the inference latency must remain well below 5 ms [23] for timely response. A missed detection rate below 1% is acceptable among class errors [24]. These things are critical during landing and take-off up and down flaps) compared to confusion between neutral flaps, these results provide a practical basis for evaluation rather than pure numerical performance metrics. Table 12 gives the category-based conclusion in single framework.

Figure 7 represents the flap state detection tried after the training session, detection is done through OAK D-Lite stereo vision camera for depth perception for further progress [25].

## 5. Conclusion

This study thus provides a detailed comparative evaluation of YOLOv5, YOLOv8 and YOLO11 model variants for vision-based flap state detection thereby establishing the effectiveness of deep learning models in automated control surface monitoring. All models achieved mAP@0.5 exceeding 98% thus clearly

showing that YOLO based architectures are highly reliable for safety critical applications. A deep class wise performance analysis shows that the neutral flap states are consistently one of the most difficult to classify primarily due to their geometric similarity with the other flap states. The need for an enhanced feature extraction technique and improved class differentiation techniques in future is pointed out by this limitation. YOLOv5s emerged as the best performing model in terms of detection accuracy among all of the tested configurations thereby achieving the highest F1 score at a low confidence threshold of 7 misdetections. This makes it particularly suitable for high precision offline analysis or ground based monitoring systems. YOLOv5m provides the optimal performance between speed and accuracy thereby maintaining inference latency below 5 ms for real time onboard UAV applications where computation resources are limited YOLOv5s and YOLOv8s thus demonstrated that the fastest raw inference speeds at approximately 0.6 ms. This research study overall establishes a strong benchmark thus offering a clear model selection guide for UAV flap monitoring systems.

## 6. Future Directions

Now that the YOLO based flap detection models have been trained, the further research will be based on using these results the hardware requirements for the fixed wing UAV will be fixed for flap state detection. After that the communication lines will be set up, that combines with the lowest inference, preprocess and postprocess timing. Both the data from inference and fight controller will be taken and compared, and check whether the fault tolerance can be achieved. When using a depth camera to do flap state detection it is possible to achieve accurate flap angle prediction with the help of the depth values combined with the class predicted and by the use of object tracking and the movement direction and magnitude of the flaps.

## References


[1] J. Zhang, Y. Liu, L. Gao, Y. Zhu, X. Zang, H. Cai, J. Zhao, Enhancing Longitudinal Flight Performance of Drones through the Coupling of Wings Morphing and Deflection of Aerodynamic Surfaces. Advanced Intelligent Systems, 6(8), (2024) 2300709. https://doi.org/10.1002/aisy.202300709

[2] L. Wang, S. Zheng, W. Wang, H. Wang, H. Liu, T. Yue, Fast Tube-Based Robust Compensation Control for Fixed-Wing UAVs. Drones, 7(7), (2023) 481. https://doi.org/10.3390/drones7070481

[3] R. Baidya, H. Jeong, YOLOv5 with ConvMixer Prediction Heads for Precise Object Detection in Drone Imagery. Sensors, 22(1), (2022) https://doi.org/10.3390/s22218424

[4] J. Guo, (2025) Research on Object Detection Methods Based on Deformable Convolutional Networks. Proceedings Volume 13486, Fourth International Conference on Computer Vision, Application, and Algorithm (CVAA 2024), Chengdu, China. https://doi.org/10.1117/12.3055890

[5] Z. Zhou, Z. Xu, L. N. Y. Cao, H. Sheng, C. Li, Y. Shang, W. Tang, Z. L. Wang, Triboelectricity Based Self-Powered Digital Displacement Sensor for Aircraft Flight Actuation. Advanced Functional Materials, 19, (2023) 2311839. https://doi.org/10.1002/adfm.202311839

[6] T. Diwan, G. Anirudh, J.V. Tembhurne, Object Detection Using YOLO: Challenges, Architectural Successors, Datasets and Applications. Multimedia Tools and Applications, 82, (2022) 9243–9275. https://doi.org/10.1007/s11042-022-13644-y

[7] M.H. Rahman, M.A. Shakil Sejan, M.A. Aziz, R. Tabassum, J.I. Baik, H. Song, A Comprehensive Survey of Unmanned Aerial Vehicles Detection and Classification Using Machine Learning Approach: Challenges, Solutions, and Future Directions. Remote Sensing, 16(5), (2024) 879. https://doi.org/10.3390/rs16050879

[8] B. Chen, Research Overview of YOLO Series Object Detection Algorithms Based on Deep Learning. Journal of Computing and Electronic Information Management, 15(3), (2024). https://doi.org/10.54097/p81rtv77

[9] J. Liu, X. Zhu, X. Zhou, S. Qian, J. Yu, Defect Detection for Metal Base of TO-Can Packaged Laser Diode Based on Improved YOLO Algorithm. Electronics, 11(10), (2022) 1561. https://doi.org/10.3390/electronics11101561

[10] D. Triska, D. Uryeu, E. C. Becerra, M. Haddad, S. Bhandari, A. Raheja, (2024) Investigating the Performance of Monocular and Stereo Vision for the Detection of Weeds in a Strawberry Field Using UAVs and Machine Learning Techniques. Proceedings Volume 13053, Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IX, United States. https://doi.org/10.1117/12.3014678

[11] N.K. Loi, K.-D. Nguyen, K. Nguyen, Real-Time Uav-Based Vehicle Detection: A Comparative Evaluation of Yolo Models. Indian Journal of Computer Science and Engineering, 14(6), (2023) 902-911. https://doi.org/10.21817/indjcse/2023/v14i5/231406039

[12] X. Chen, C. Liu, L. Chen, X. Zhu, Y. Zhang, C. Wang, A Pavement Crack Detection and Evaluation Framework for a UAV Inspection System Based on Deep Learning. Applied Sciences, 14(3), (2024) 1157. https://doi.org/10.3390/app14031157

[13] J. Kim, S.H. Ahn, S.K. Yoon, H.R. Noh, H.J. Jeong, C.W Sun, B.K Cho, (2025). Development of a Smartphone-Based Bone Maturity Detection Algorithm with XAI for Beef Carcass Grading. Research Square. https://doi.org/10.21203/rs.3.rs-7362863/v1

[14] M.L. Ali, Z. Zhang, The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection. Computers, 13(12), (2024) 336. https://doi.org/10.3390/computers13120336

[15] B. Peng, H. Zhang, N. Yang, J. Xie. Vehicle Recognition From Unmanned Aerial Vehicle Videos Based on Fusion of Target Pre-Detection and Deep Learning. Sustainability, 14(13), (2022) 7912. https://doi.org/10.3390/su14137912

[16] F. Zhang, W. Cao, J. Gao, S. Liu, C. Li, K. Song, H. Wang, Underwater Object Detection Algorithm Based on an Improved YOLOv8. Journal of Marine Science and Engineering, 12(11) (2024) 1991. https://doi.org/10.3390/jmse12111991

[17] Z. Kaleem, Lightweight and Computationally Efficient YOLO for Rogue UAV Detection in Complex Backgrounds. IEEE Transactions on Aerospace and Electronic Systems, 61(2) (2025) 5362–5366. https://doi.org/10.1109/taes.2024.3464579

[18] Z. Yang, X. Lan, H. Wang, Comparative Analysis of YOLO Series Algorithms for UAV-Based Highway Distress Inspection: Performance and Application Insights. Sensors, 25(5) (2025) 1475. https://doi.org/10.3390/s25051475

[19] C. Ma, K. Du, A. Hamdulla, Small Target Detection Algorithm for Flapping Wing UAV based on Improved YOLOv8. Second International Symposium on Computer Applications and Information Systems, 12721, (2023) 419-425. https://doi.org/10.1117/12.2683452

[20] B. Aydin, S. Singha, Drone Detection Using YOLOv5. Eng, 4(1) (2023) 416–433. https://doi.org/10.3390/eng4010025

[21] B. Govindarajan, R. Jaganraj, V. Srinivasan, T. Kumaran, Evolution of UAV Technology from Early Innovations to Future Horizons. In Innovations and Developments in Unmanned Aerial Vehicles IGI Global Scientific Publishing, (2025) 53-94. https://doi.org/10.4018/979-8-3693-8462-6.ch004

[22] V. Srinivasan, J.V. Sai Prasanna Kumar, AI-based calibration of 8-beam LiDAR for fault detection in fixed-wing UAV flaps. International Journal of Sustainable Aviation, 12(1), (2026) 57-74. https://doi.org/10.1504/IJSA.2026.10076972

[23] R. Tapia, J. Luna-Santamaria, I. Rodríguez, J. Gomez, J. Dios, A. Ollero, Flight of the Future: An Experimental Analysis of Event-Based Vision for Online Perception Onboard Flapping-Wing Robots. Advanced Intelligent Systems, 7(10) (2025) 2401065. https://doi.org/10.1002/aisy.202401065

[24] J. Kim, S.M. Lee, D.E. Kim, S. Kim, M.J. Chung, Z. Kim, T. Kim, K.T. Lee, Development of an Automated Free Flap Monitoring System based on Artificial Intelligence. JAMA Network Open, 7(7), (2024) 2424299. https://doi.org/10.1001/jamanetworkopen.2024.24299

[25] J. Liu, J. Xiang, Y. Jin, R. Liu, J. Yan, L. Wang, Boost Precision Agriculture with Unmanned Aerial Vehicle Remote Sensing and Edge Intelligence: A Survey. Remote Sensing, 13(21) (2021) 4387. https://doi.org/10.3390/rs13214387

## Acknowledgments

The authors thank Dr. V. Kamalakar for his technical support through Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Avadi, Chennai for conducting this research under the seed-funded project work VTU/SEED FUND/FY2023-2024/009. The authors also thank Dr. P. Chandrakumar B.E., M.Tech., Ph.D., Dean R&D for motivation and support.

## Authors Contribution Statement

V. Srinivasan: Conceptualization, Methodology, Investigation, Writing – Original Draft. J. V. Sai Prasanna Kumar: Validation, Formal Analysis, Supervision, Funding Acquisition. All the authors have read and agreed to the published version of the manuscript.

## Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

## Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

## Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

## Has this article screened for similarity?

Yes

## About the License