Smoke Segmentation Improvement Based on Fast Segment Anything Model with YOLOv11 for a Wildfire Monitoring System

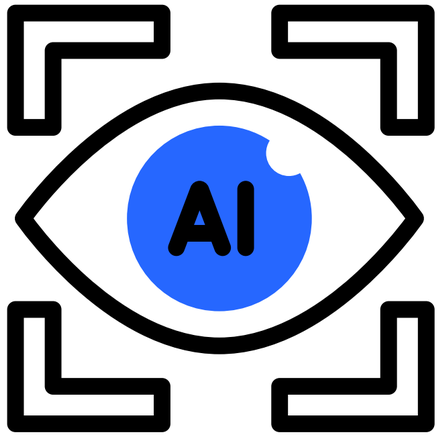
Puchit Bunpleng, Puthtipong Thunyatada, Bhutharit Aksornsuwan, Kanokvate Tungpimolrut, Ken T. Murata

10th IoTBDS, Porto, Portugal

Apr. 6-8, 2025

Objectives

- Prevent or reduce wildfire.
- Use computer vision to enhance wildfire detection by segmenting smoke.
- Make a model to predict the errors and performance without access to the ground truth, with the aim of iterative self-improvement of segmentation models.



How can we prevent this?

Current Approach

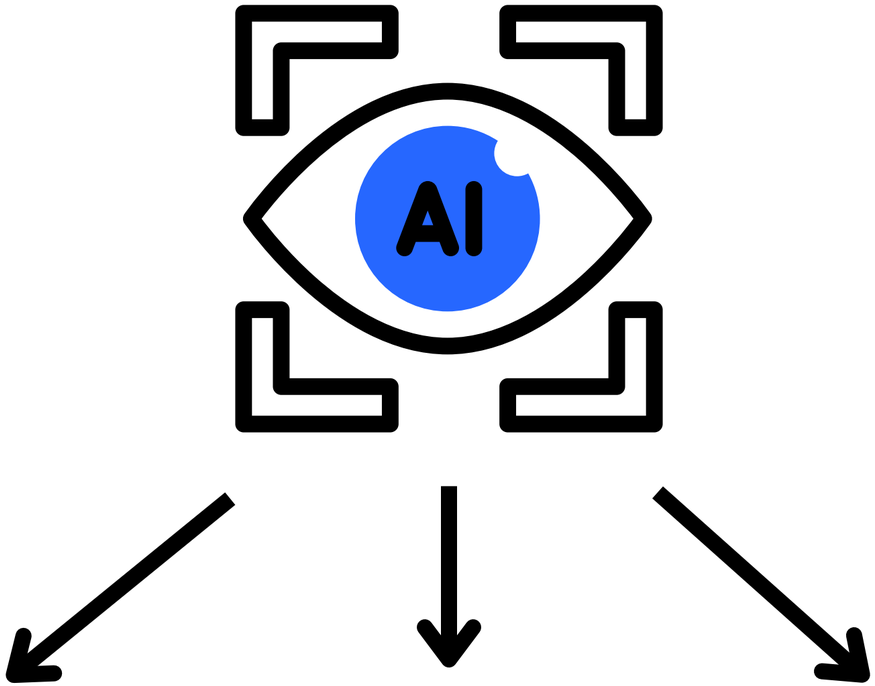
+

Can’t monitor 24/7

Human Error

Our Approach

A combination of artificial intelligence to detect & segment the smoke.



Location of the smoke

Size of the smoke

Shape of the smoke

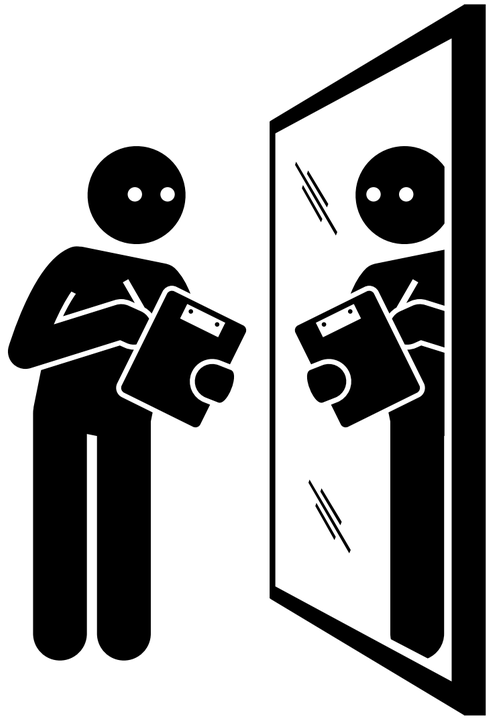
What Makes Our Approach Innovative?

Self-assessment allows continuous model improvement and optimization.

Self-assessment

↓

Self-improvement



How It Works

Our solution combines three powerful technologies.

Detect

ultralytics YOLO11

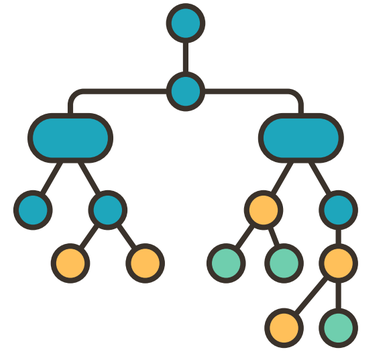
YOLOv11

Segment

Fast SAM

FastSAM

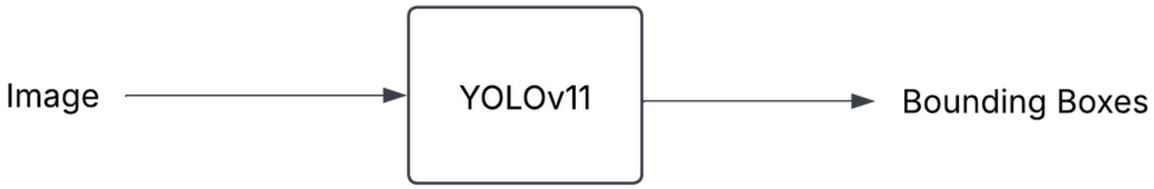
Performance Prediction



Machine Learning Models

YOLOv11

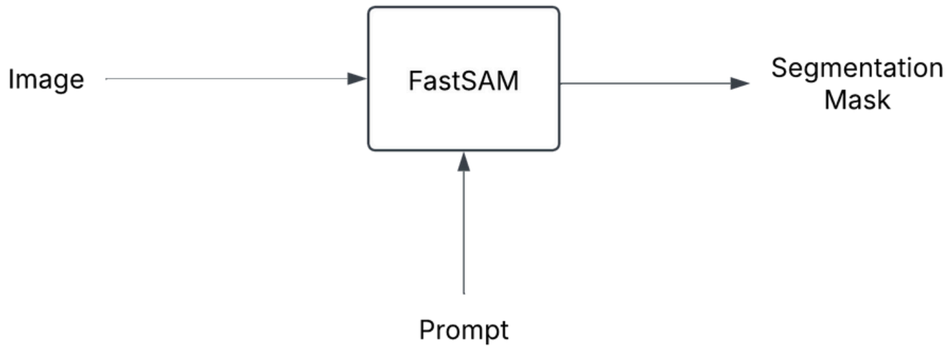
“You Only Look Once” model developed by Ultralytics. This model can detect and label objects in any given picture.

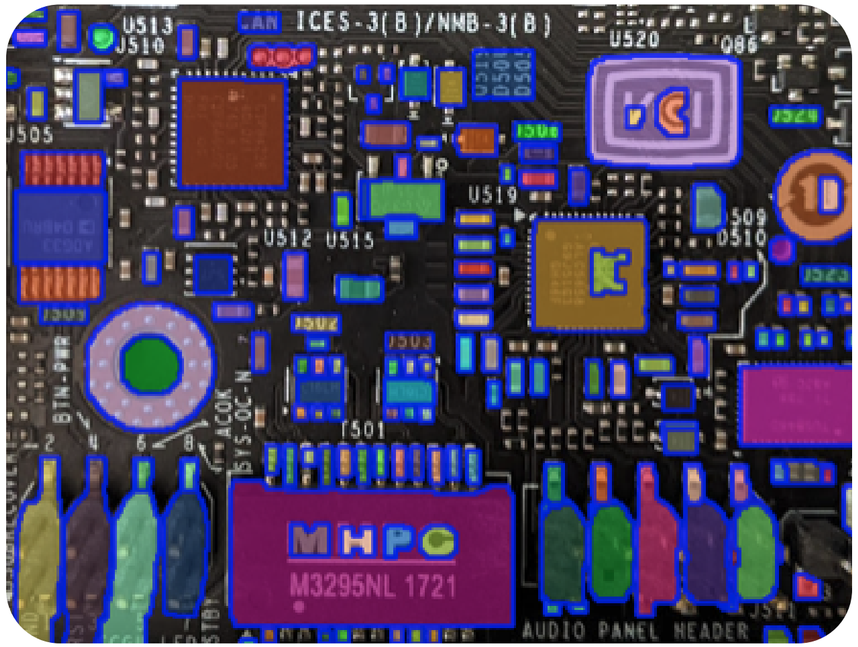


FastSAM

Fast Segment Anything Model developed by Ultralytics. This model can accurately detect and segment the objects given a prompt without fine-tuning.

Segment all objects → Filter out irrelevant objects using the prompt



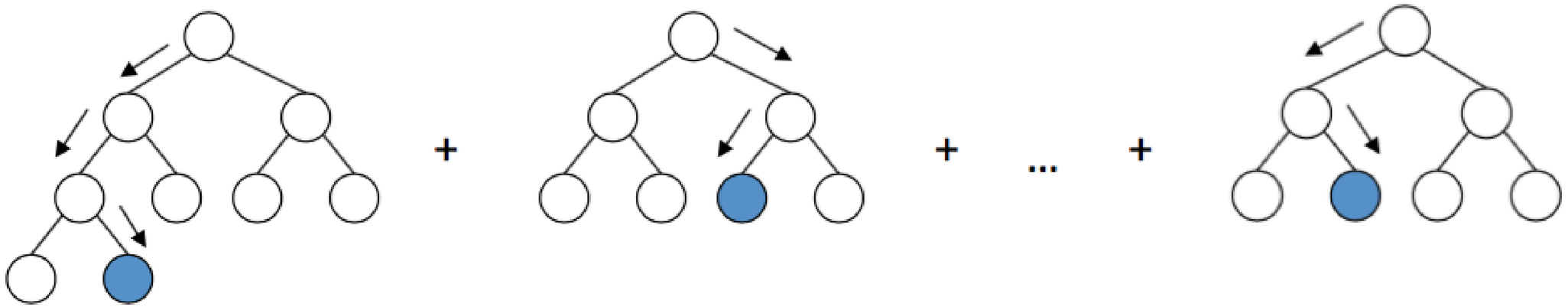


Prompt Type

- Bounding Boxes
- Text
- Points
- Combined

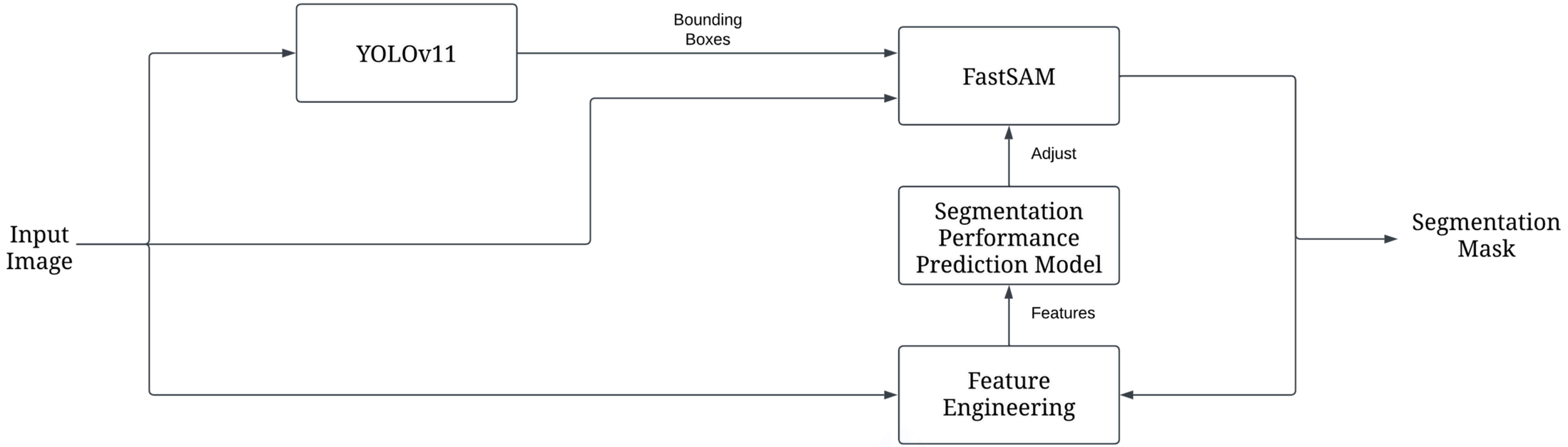
Gradient Boosting

Gradient boosting builds a series of weak learners (such as small decision trees), where each successive tree learns and corrects the errors made by previous ones.



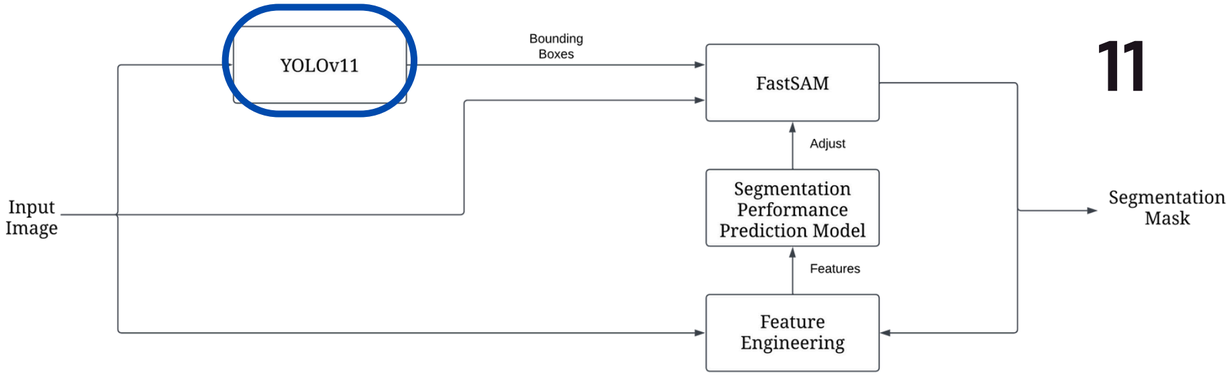
Proposed Method

Workflow Diagram



Proposed Method

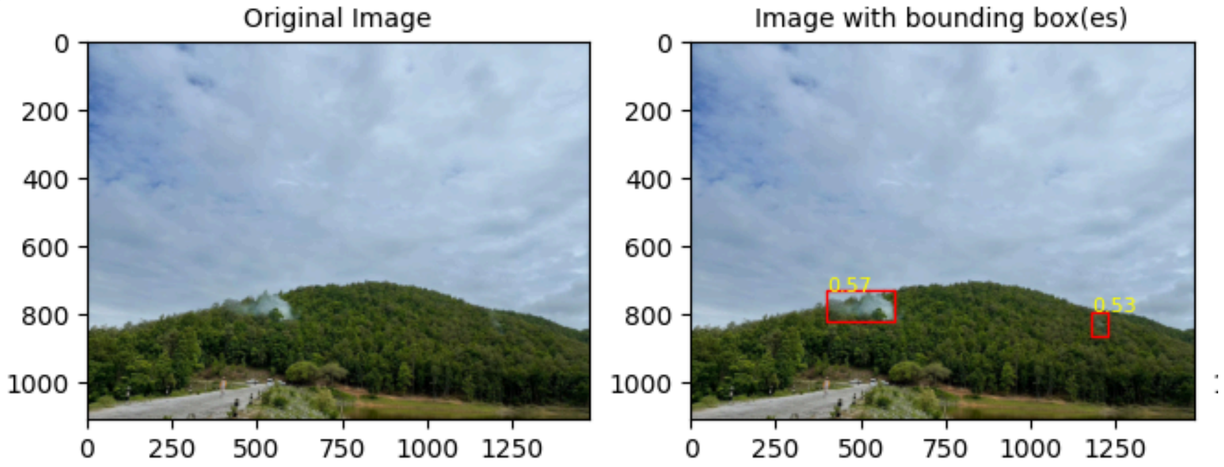
1. Smoke Detection With YOLOv11



Fine-tuned for smoke detection using ~ 800 images with and without smoke

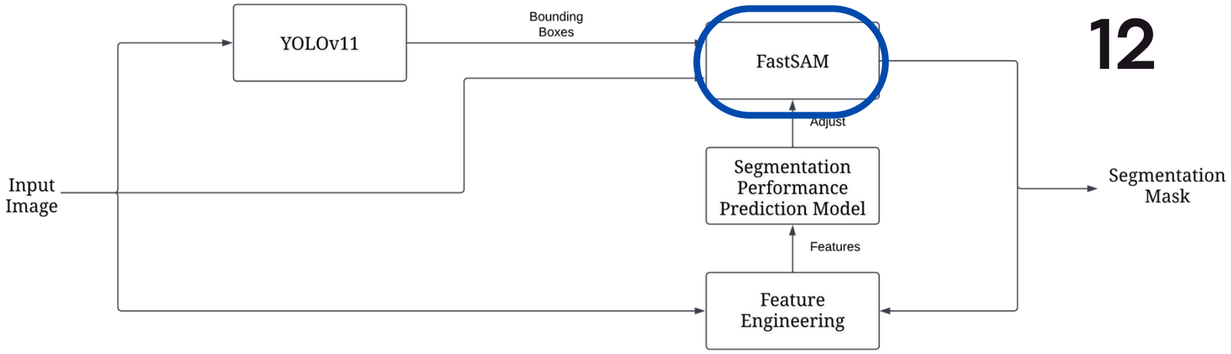
from the FireSpot dataset

- 70% training, 15% validation, 15% testing
- Resizing to 640×640 pixels
- 40 epochs with a batch size of 16

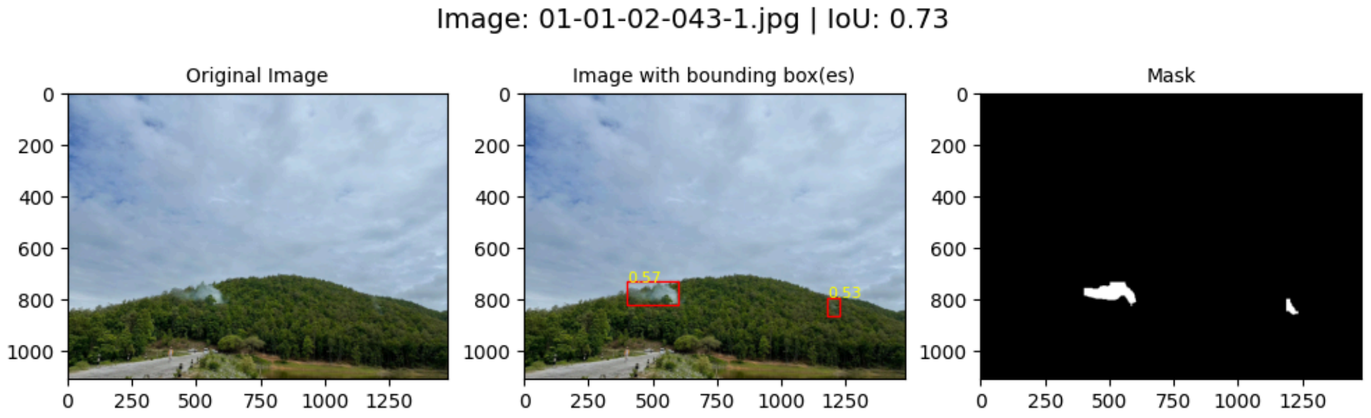


Proposed Method

2. Smoke Segmentation With FastSAM

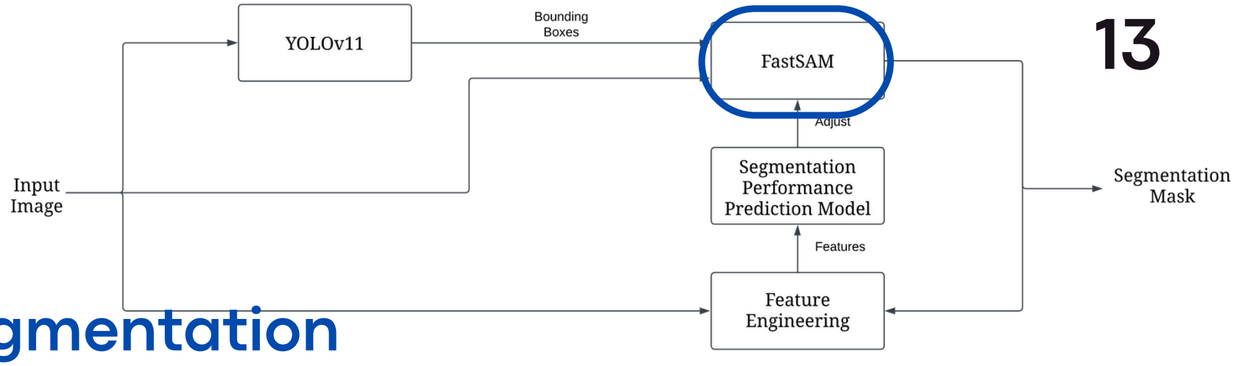


Employed **FastSAM-x** model for precise smoke segmentation



FastSAM visualization

Proposed Method

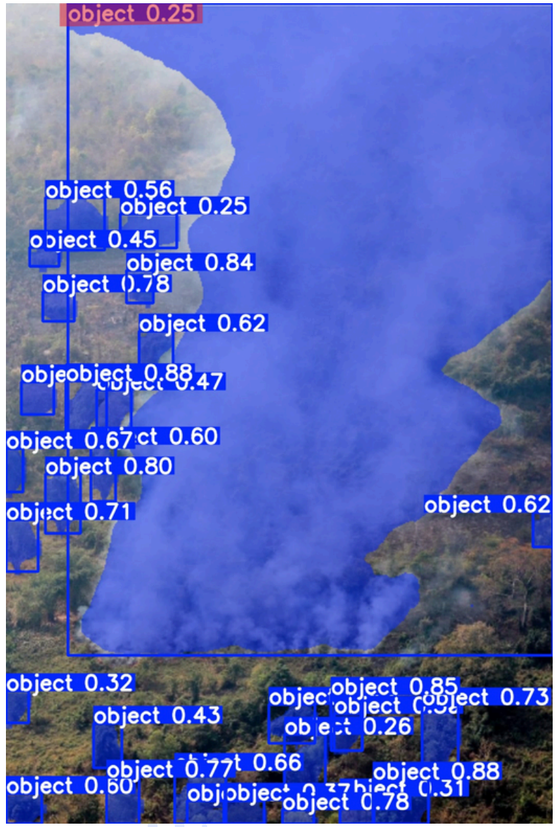


2. Smoke Segmentation With FastSAM

- **Confidence threshold for object segmentation**

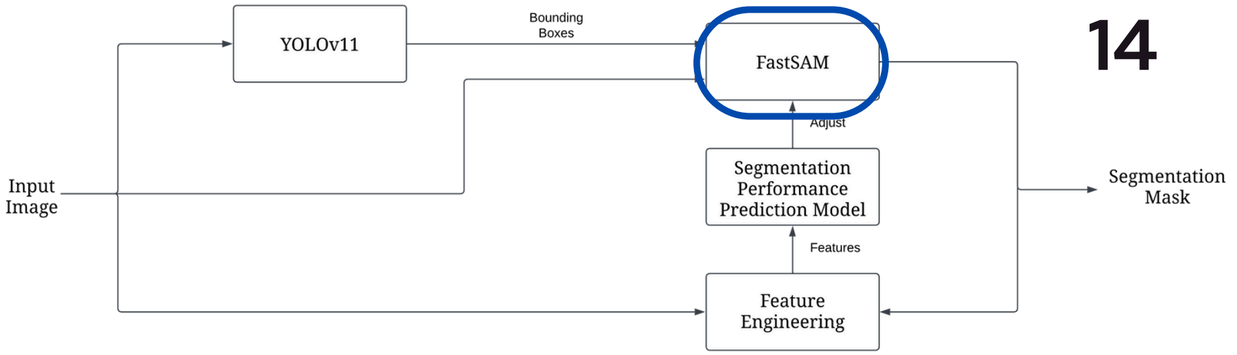
The confidence threshold is used to determine whether each detected object is counted as an object or not.

Confidence threshold α = 0.005



FastSAM objects visualization

Proposed Method

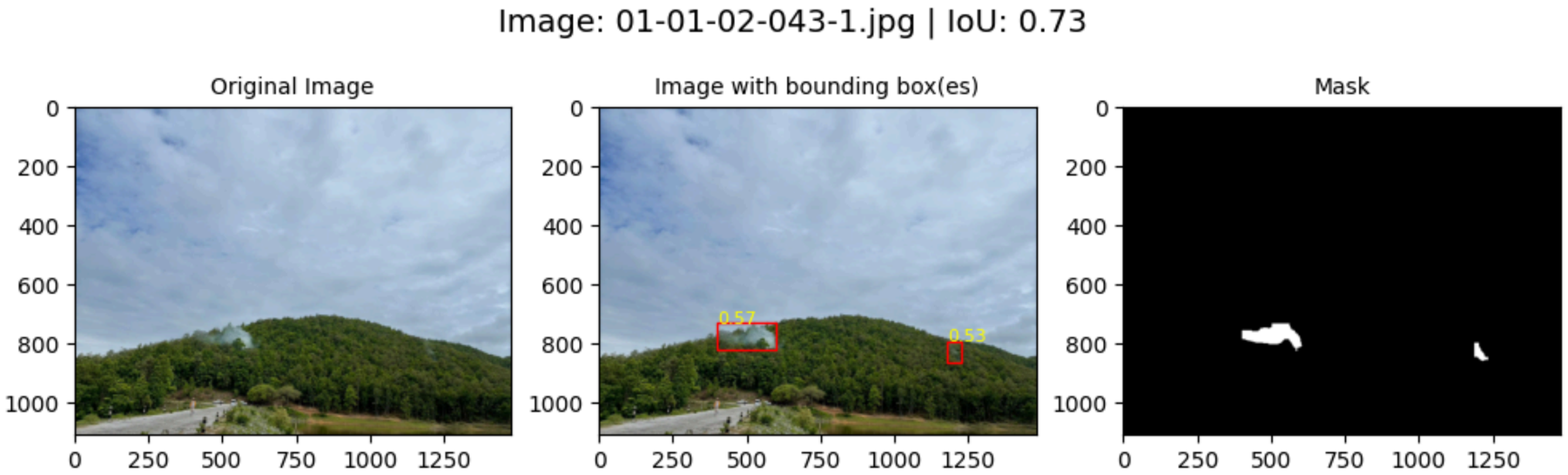


2. Smoke Segmentation With FastSAM

- **Prompt optimization**

Optimal combination of prompts for FastSAM

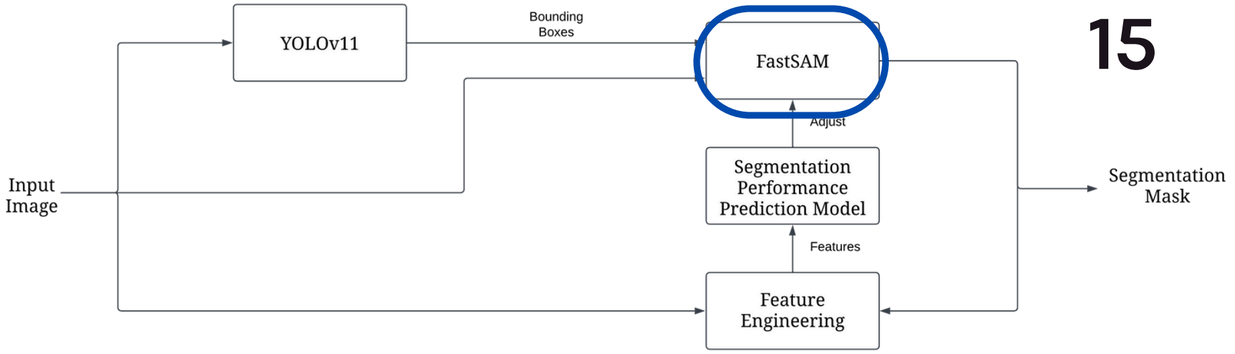
Using only the bounding boxes from YOLOv11 as prompts produced the best segmentation results



Proposed Method

2. Smoke Segmentation With FastSAM

- **Segmentation Model Validation**

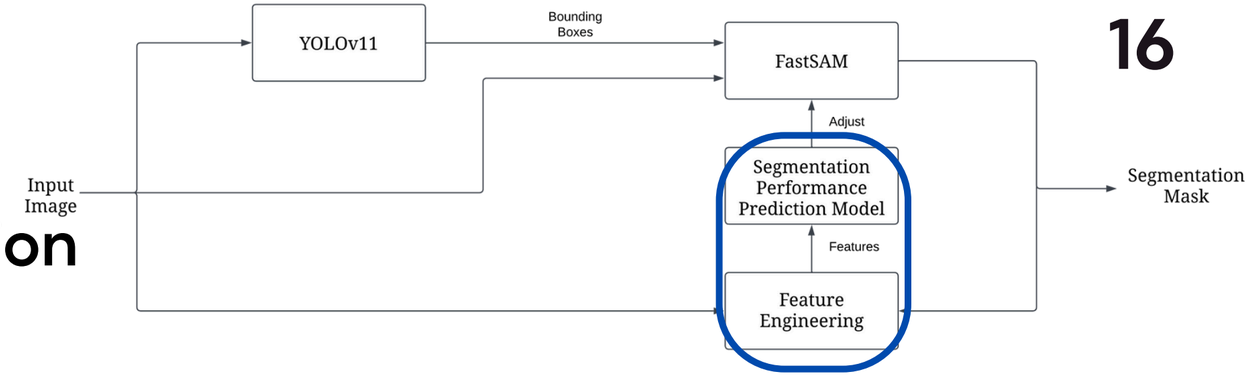


Calculated the Intersection over Union (IoU) to evaluate segmentation quality

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

Proposed Method

3. Segmentation Performance Prediction



A system that predicts segmentation performance without access to ground truth masks

Three key metrics to predict

1. True Positive Proportion (TPP) - equivalent to IoU
2. False Negative Proportion (FNP) - proportion of ground truth not included in the output mask
3. False Positive Proportion (FPP) - proportion of pixels that were incorrectly classified as smoke

$$\mathrm{TPP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}},$$

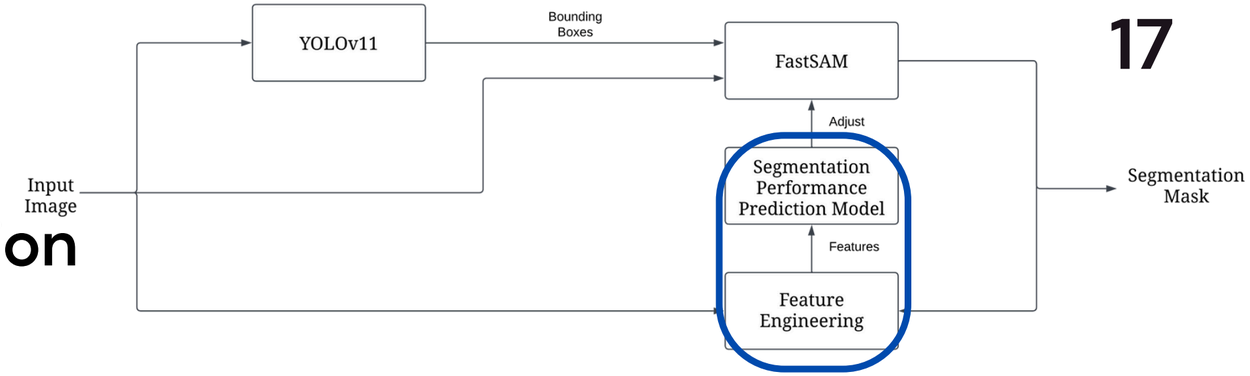
$$\mathrm{FNP} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}},$$

$$\mathrm{FPP} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}},$$

Proposed Method

3. Segmentation Performance Prediction

- **Machine Learning Models**



Implemented two approaches for predicting our metrics (TPP, FNP, FPP)

1.Gradient Boosting Models

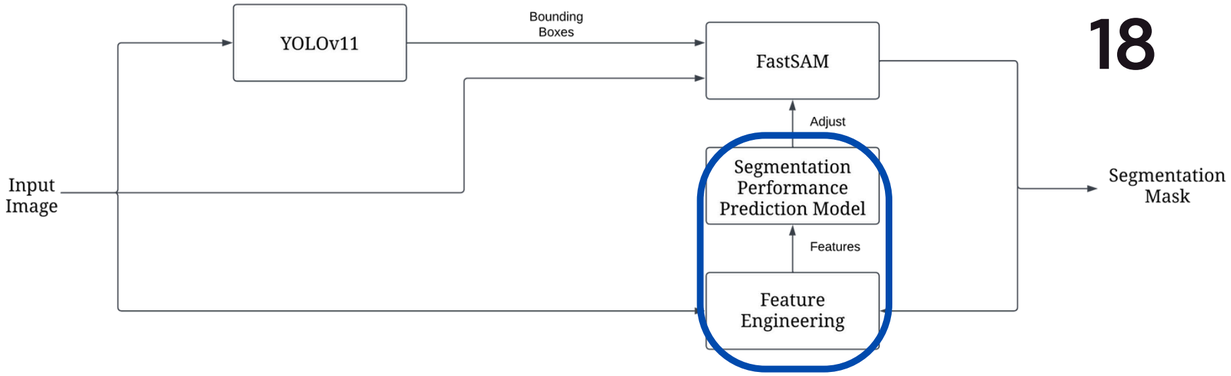
- Scikit-learn
- LightGBM
- XGBoost

2.Pre-trained Deep Learning Models

- ResNet18
- ResNet50
- EfficientNet-B2

- 80:20 train-test split
- Mean Squared Error (MSE)
- Coefficient of Determination (R^2)

Proposed Method



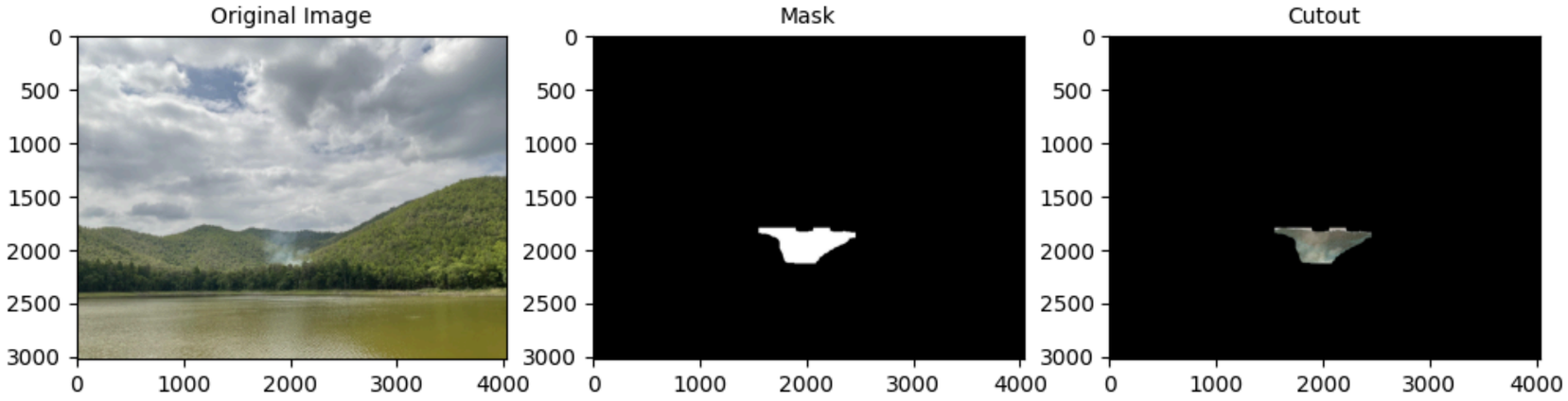
3.1 Gradient Boosting

- **Features Engineering**

1. Isolate regions of analysis (obtain cutout)

Multiply each pixel of the original image with the corresponding pixel of the mask.

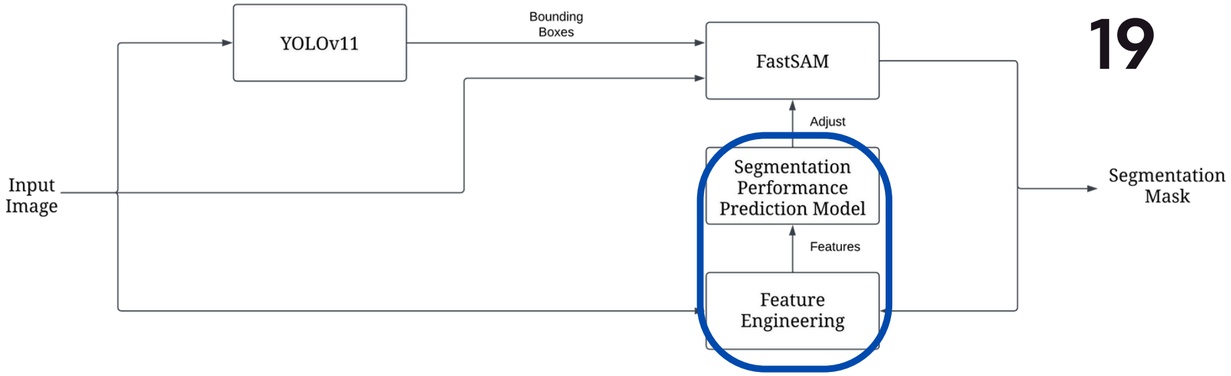
Smoke pixels = 1 (white)
All other pixels = 0 (black)



Proposed Method

3.1 Gradient Boosting

- **Features Engineering**



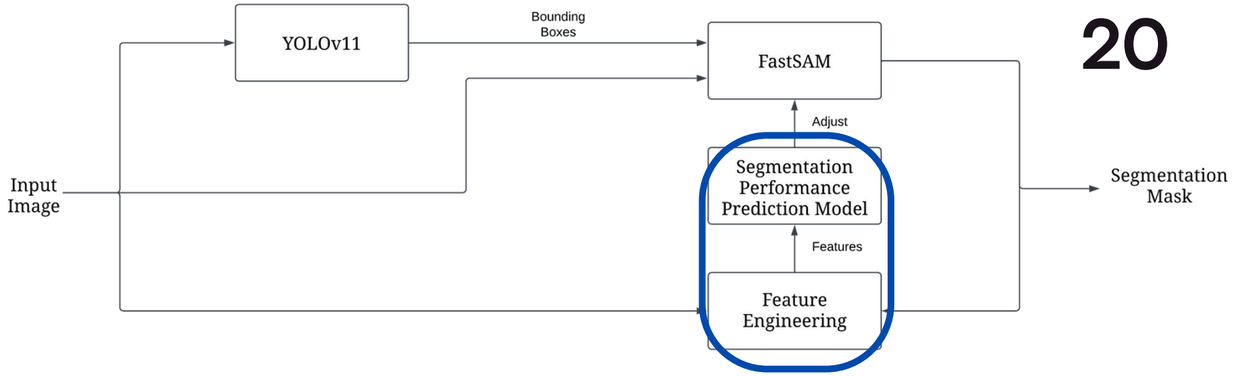
2. Extract all features designed to differentiate smoke from other elements

1. Color Histograms
2. Color Moments
3. Edge Features
4. Texture Analysis
5. Color Coverage

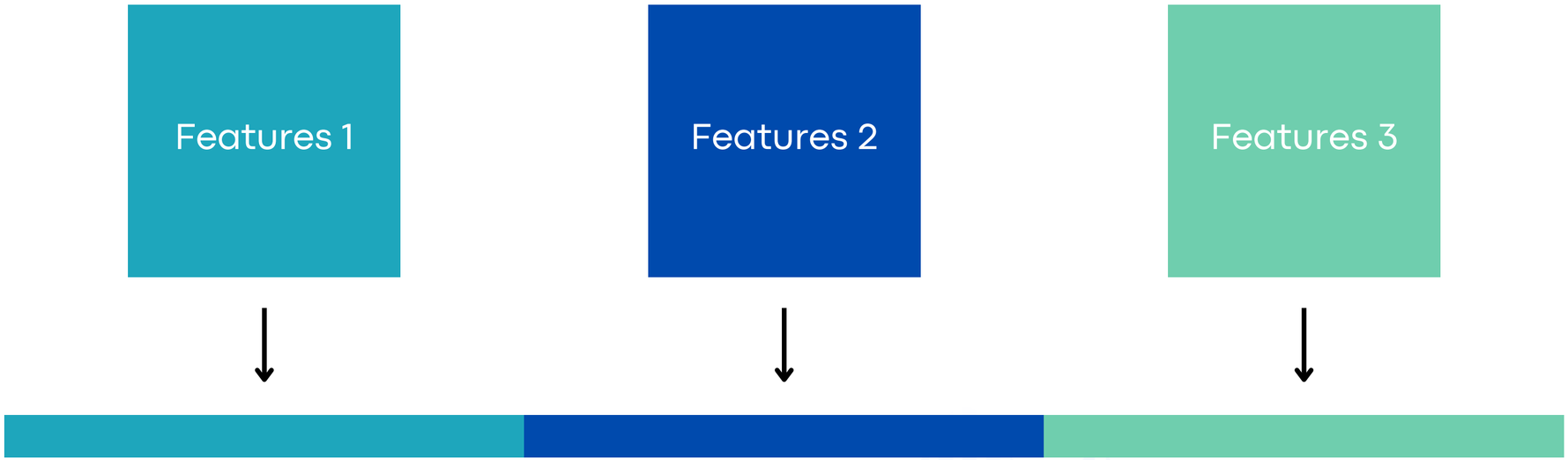
Proposed Method

3.1 Gradient Boosting

- **Features Engineering**



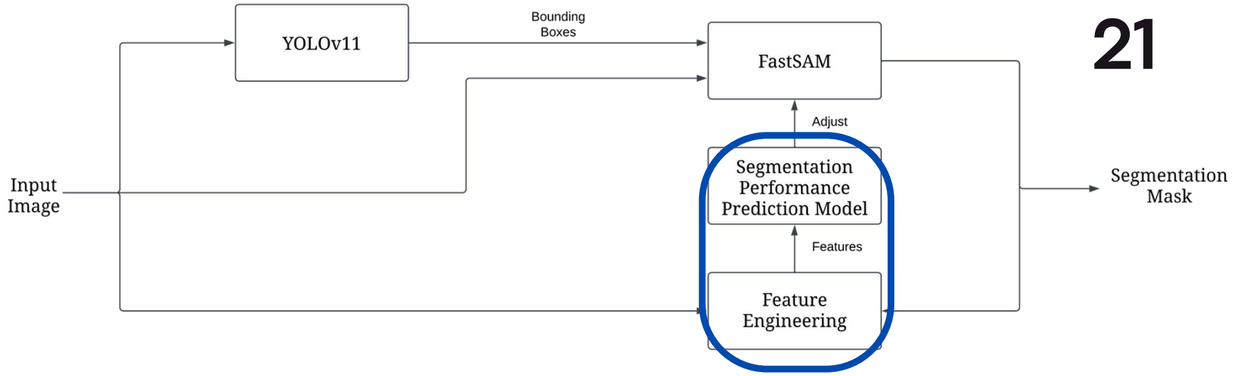
3. Flatten all the features and concatenate them into a 1D vector.



Proposed Method

3.2 Pre-trained Deep Learning

- **Architecture**



Backbone:

- ResNet18
- ResNet50
- EfficientNet-B2

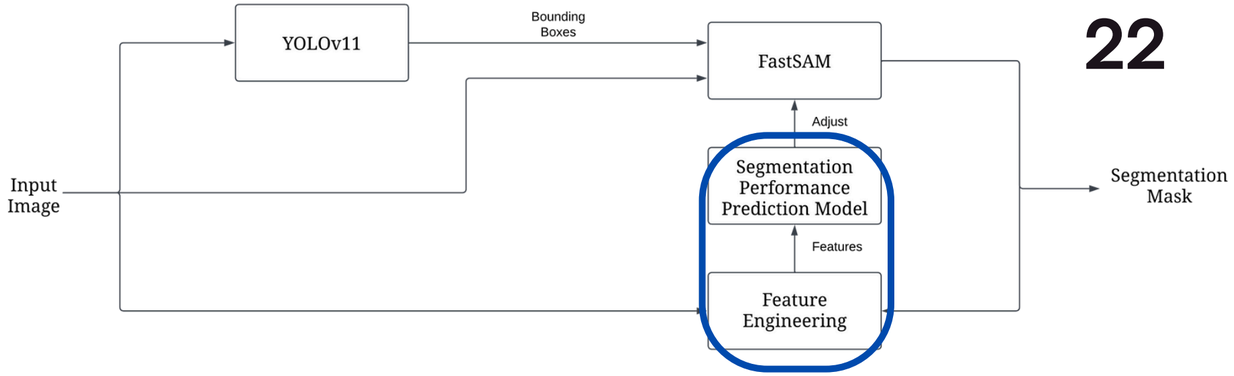
Classifier layer:

- 3 sigmoid activation functions for predicting our metrics (TPP, FNP, FPP)

Proposed Method

3.2 Pre-trained Deep Learning

- **Feature Engineering**



1. For each layer of the original image, concatenate it with its mask.
2. Resize them to 224 x 224

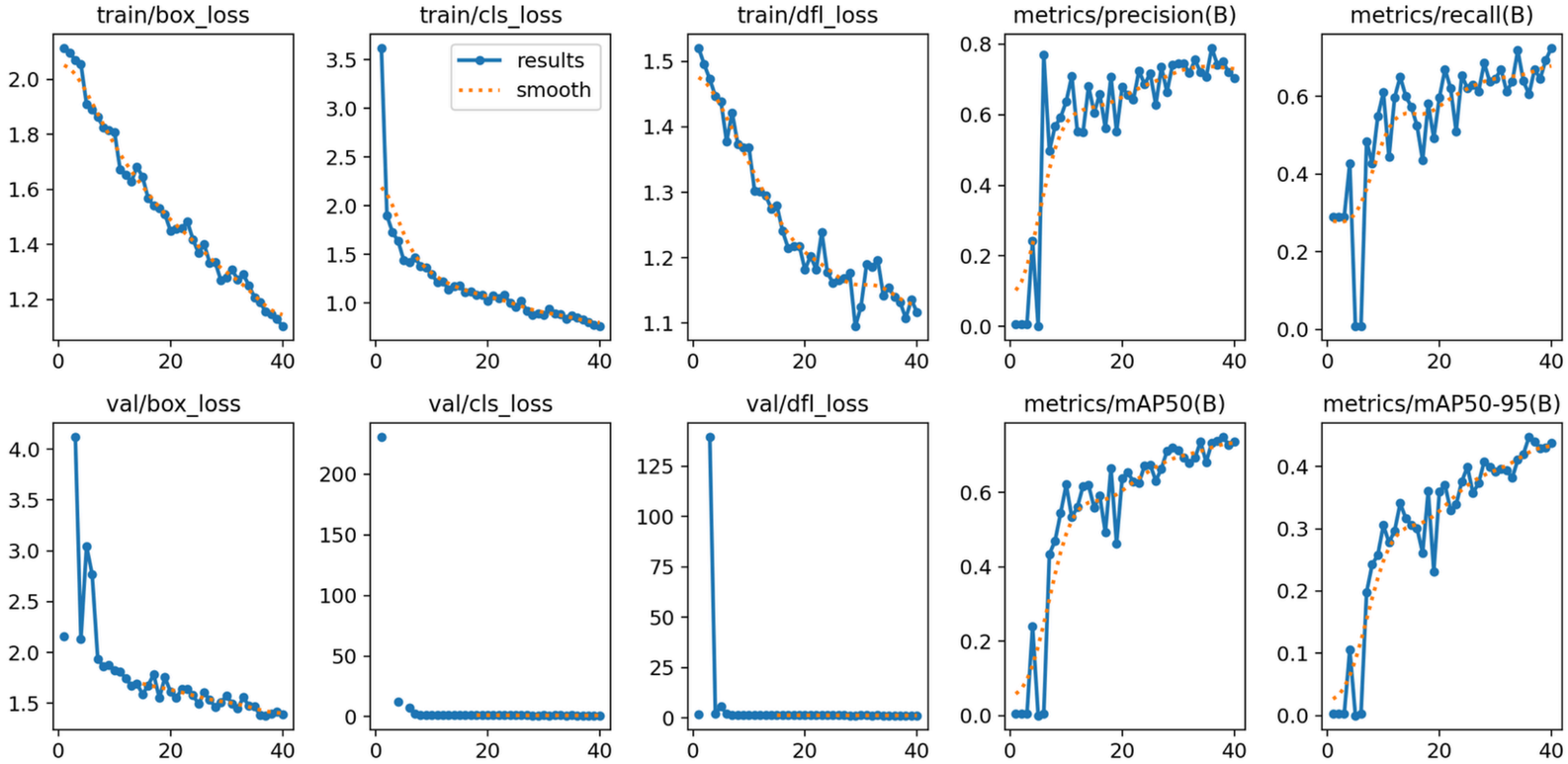
Results and Evaluation

Results and Evaluation

YOLOV11 Fine-tuning Performance

- Precision of 0.70 and recall of 0.72 on the validation set
- Mean Average Precision (mAP) of 0.74 at IoU 0.5
- mAP of 0.44 across the IoU range from 0.5 to 0.95

- box_loss = box locations accuracy
- cls_loss = classification accuracy
- dfl_loss = box locations accuracy through the probability distribution
- precision = how many were actually correct?
- recall = how many were correctly identified?

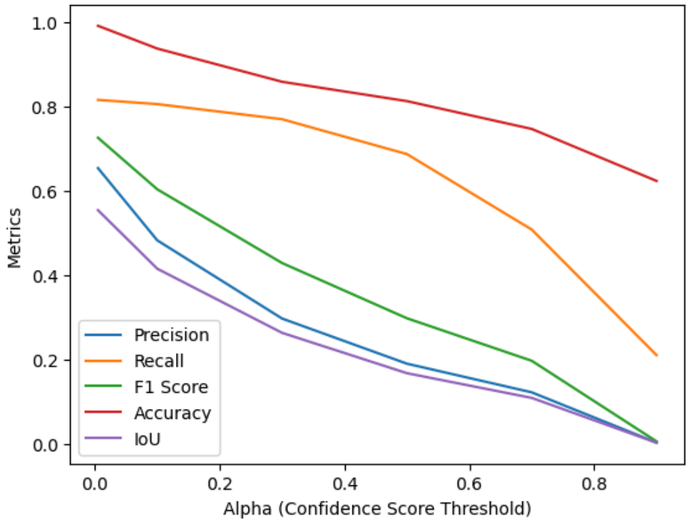


Results and Evaluation

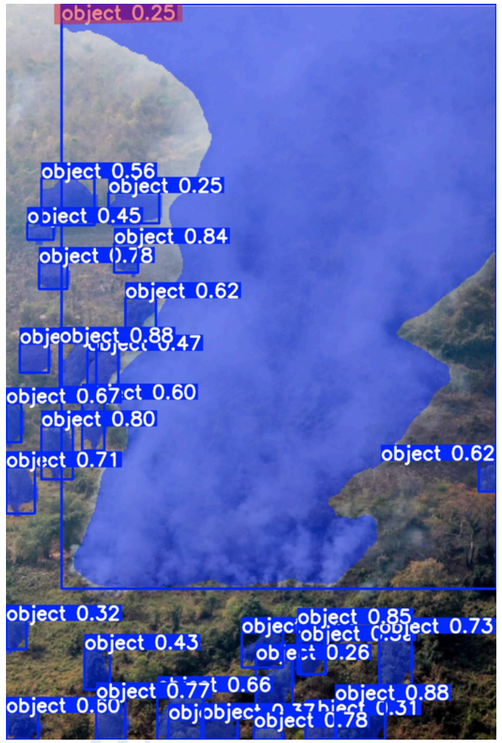
FastSAM Segmentation Performance

- Bounding box inputs from YOLO provided the most significant performance gains
- Point prompts alone yielded only marginal segmentation performance
- Text prompts via CLIP encoder delivered moderate improvements

Optimal threshold α = 0.005
(As close to 0 as possible)



Confidence Score



FastSAM visualization

Results and Evaluation

Model Comparison: Gradient Boosting vs. Deep Learning

Gradient Boosting models substantially outperformed

Deep Learning approaches across key metrics

Model	Target	MSE	R^2
Scikit-learn	FNP	0.0523	-0.0404
	TPP	0.0335	0.4164
	FPP	0.0002	0.9604
LightGBM	FNP	0.0563	-0.1194
	TPP	0.0315	0.4510
	FPP	0.0010	0.7916
XGBoost	FNP	0.0567	-0.1285
	TPP	0.0336	0.4607
	FPP	0.0002	0.9404

Gradient Boosting

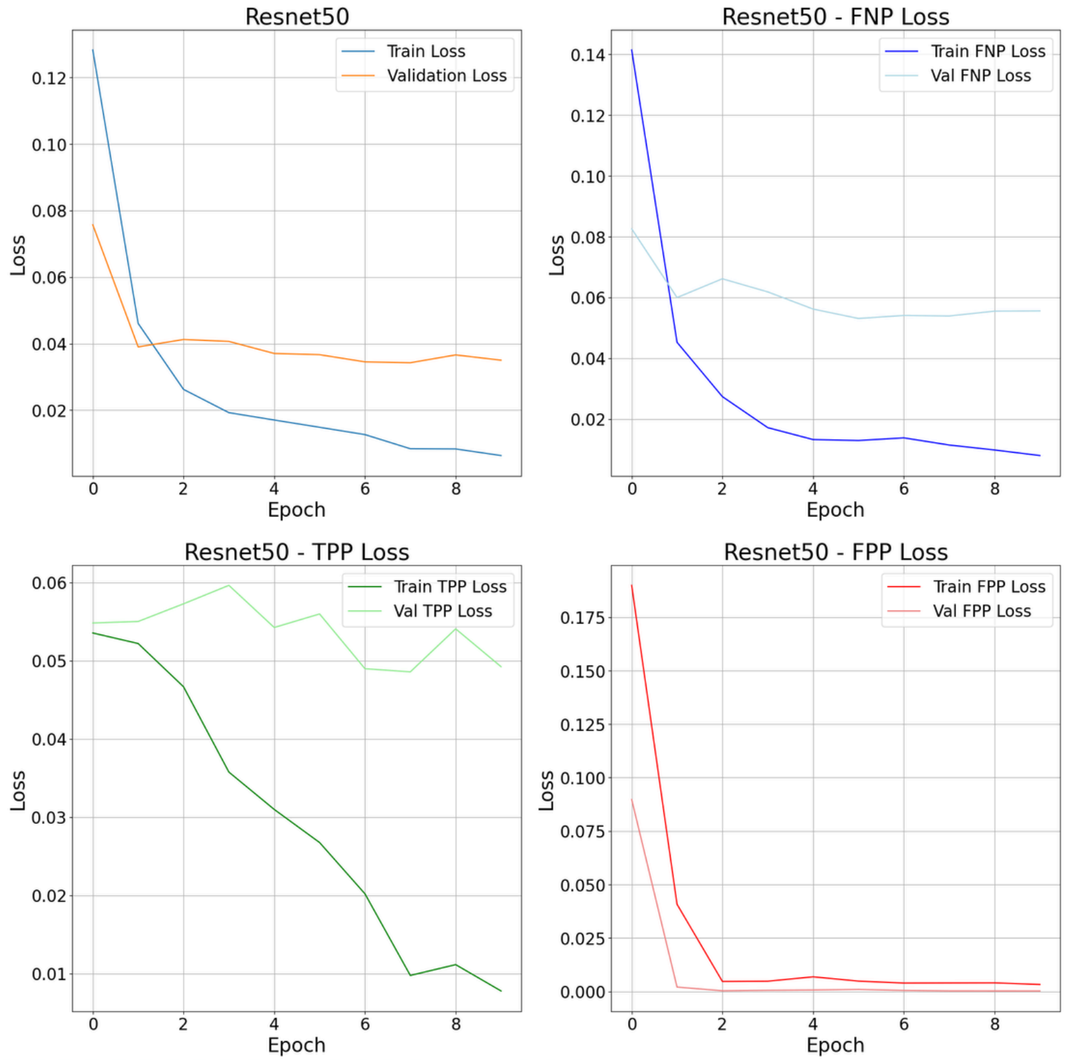
Model	Target	MSE	R^2
ResNet50	FNP	0.0565	-0.0654
	TPP	0.0462	0.0659
	FPP	0.0002	-1.1521
ResNet18	FNP	0.0519	0.0622
	TPP	0.0430	0.1278
	FPP	0.0003	-4.1705
EfficientNet-B2	FNP	0.0502	0.0475
	TPP	0.0439	0.0996
	FPP	0.0007	-11.0456

Deep Learning

Results and Evaluation

Training Dynamics

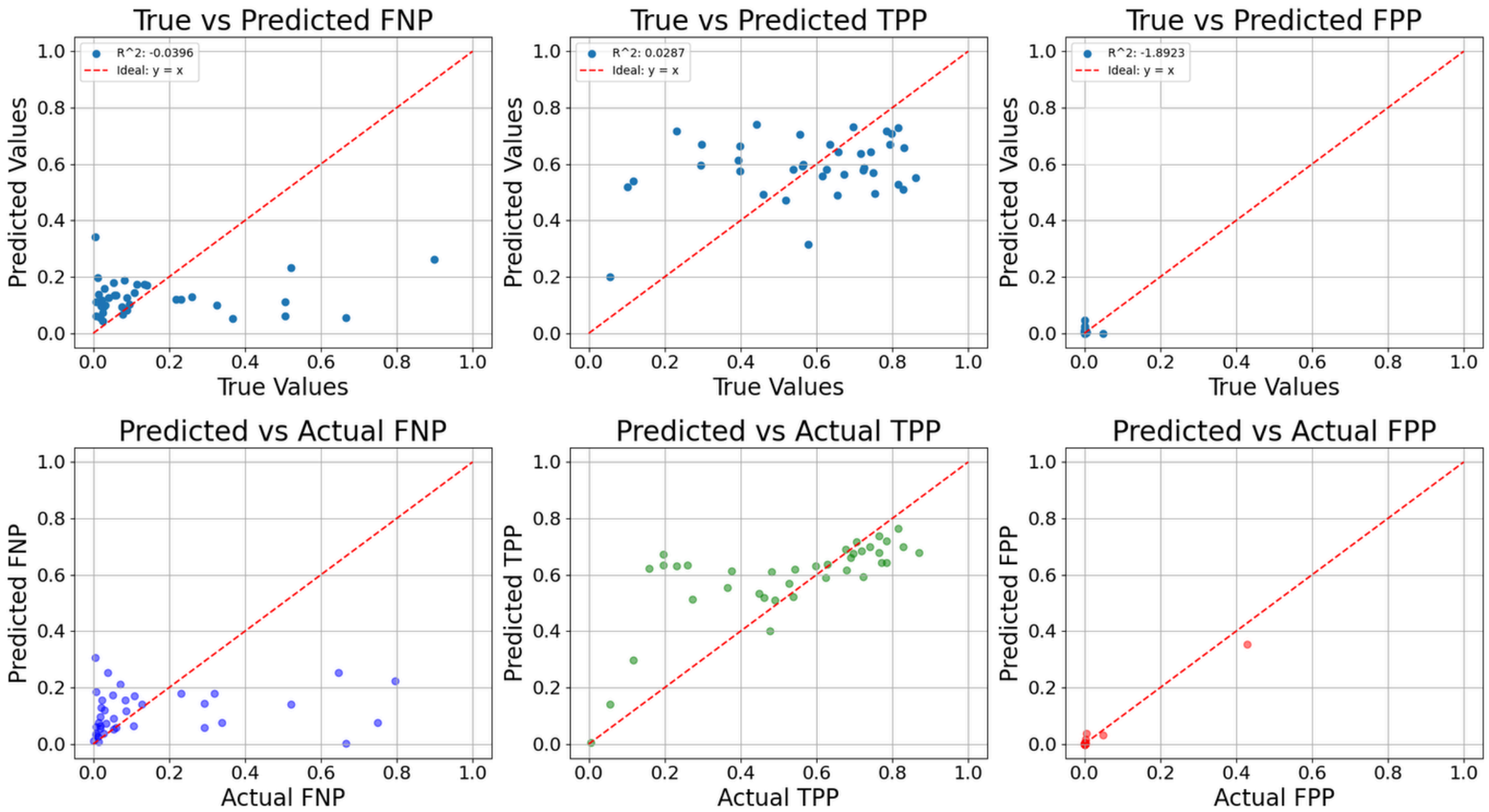
- Sign of overfitting
- 189-image dataset (80% training and 20% testing)



Results and Evaluation

Deep Learning

Gradient boosting



The result should align with the red dashed line

Results and Evaluation

Inference Latency

Gradient boosting offers significant speed advantages

Model	Inference Latency
Scikit-learn	$4.97\,\text{ms} \pm 429\,\mu\text{s}$
LightGBM	$8.36\,\text{ms} \pm 468\,\mu\text{s}$
XGBoost	$3.78\,\text{ms} \pm 579\,\mu\text{s}$

Gradient Boosting

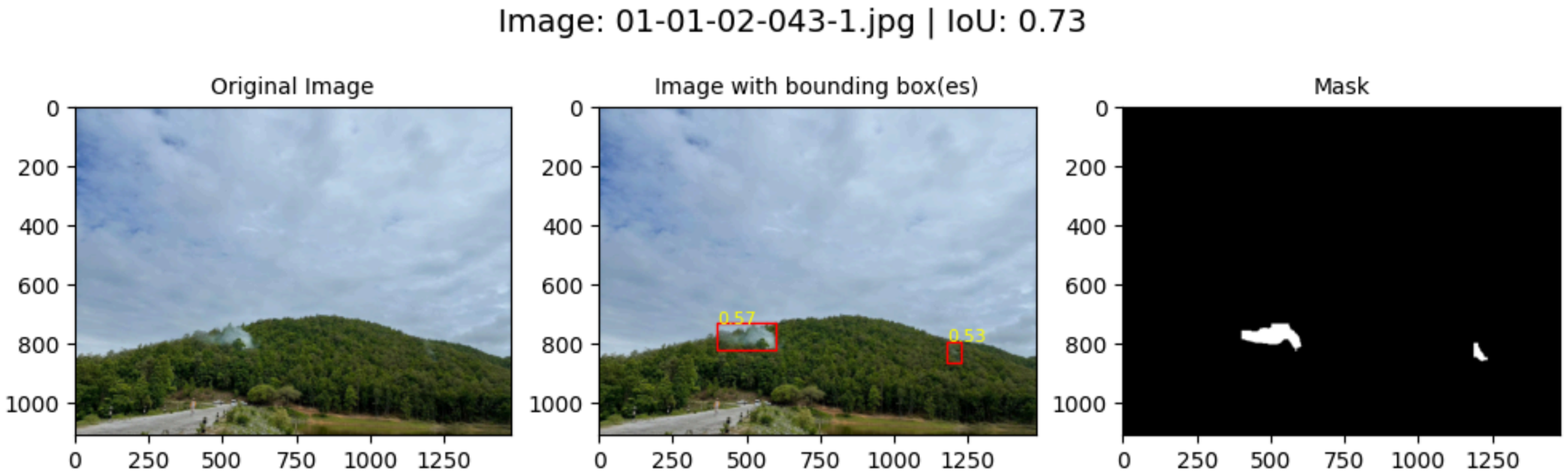
Model	Inference Latency
ResNet18	$35.1\,\text{ms} \pm 850\,\mu\text{s}$
ResNet50	$109\,\text{ms} \pm 101\,\mu\text{s}$
EfficientNet-B2	$67.3\,\text{ms} \pm 141\,\mu\text{s}$

Deep Learning

Results and Evaluation

Practical Implications

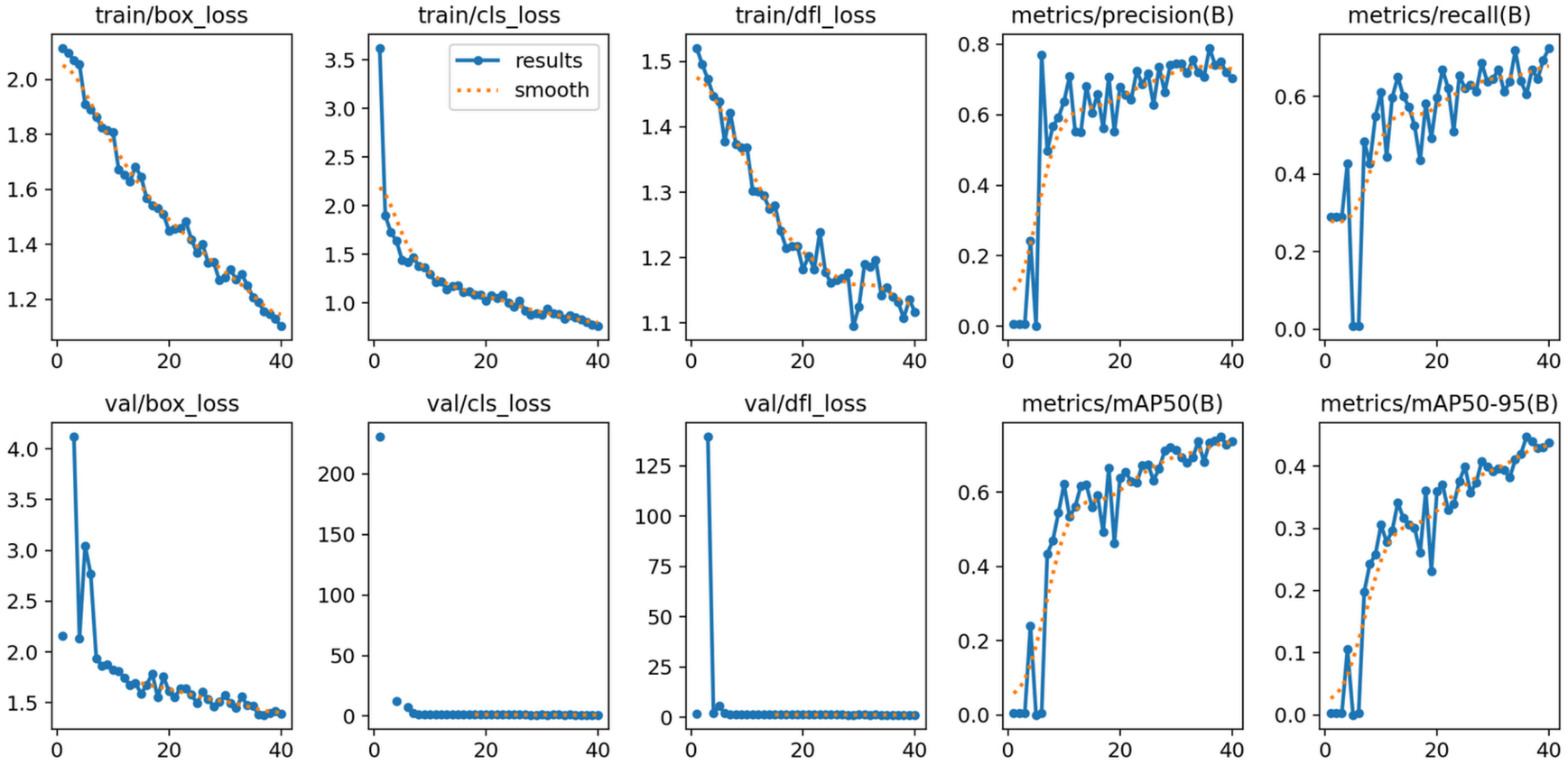
1. Accurately detects and segments smoke in diverse environments
2. Operates with low latency, making it suitable for real-time applications
3. Provides estimation of its own performance without ground truth



Discussion

YOLOv11 Analysis

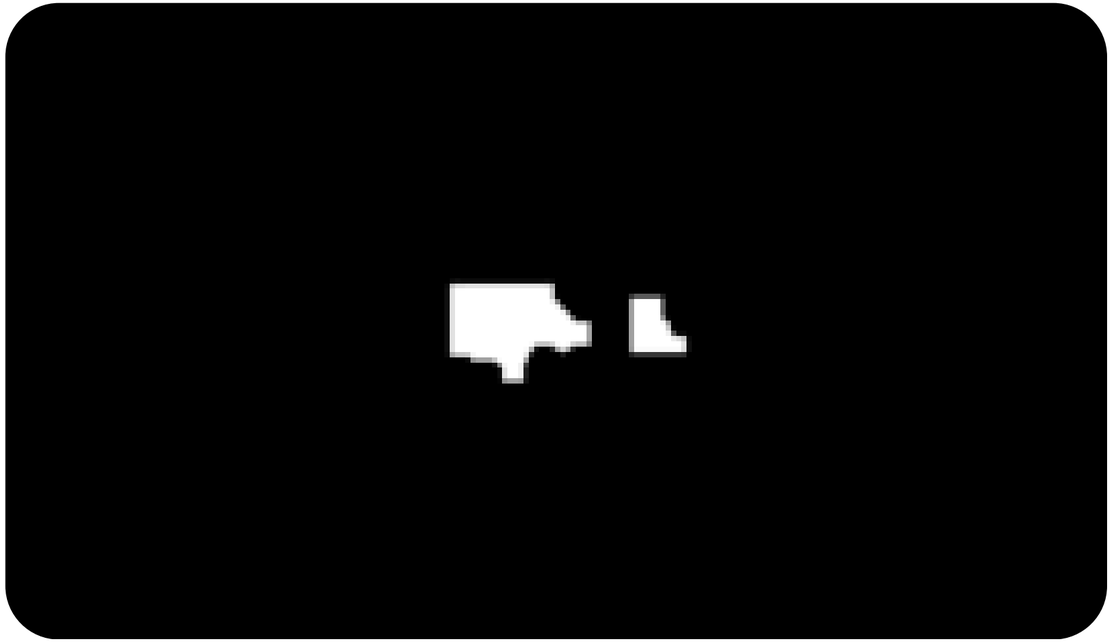
YOLOv11 performs noticeably well



FastSAM Limitations

Correct Segmentation

Original Image



Mask

FastSAM Limitations

Incorrect Segmentation

Original Image

Mask

FastSAM Limitations

Incorrect Segmentation Explanation

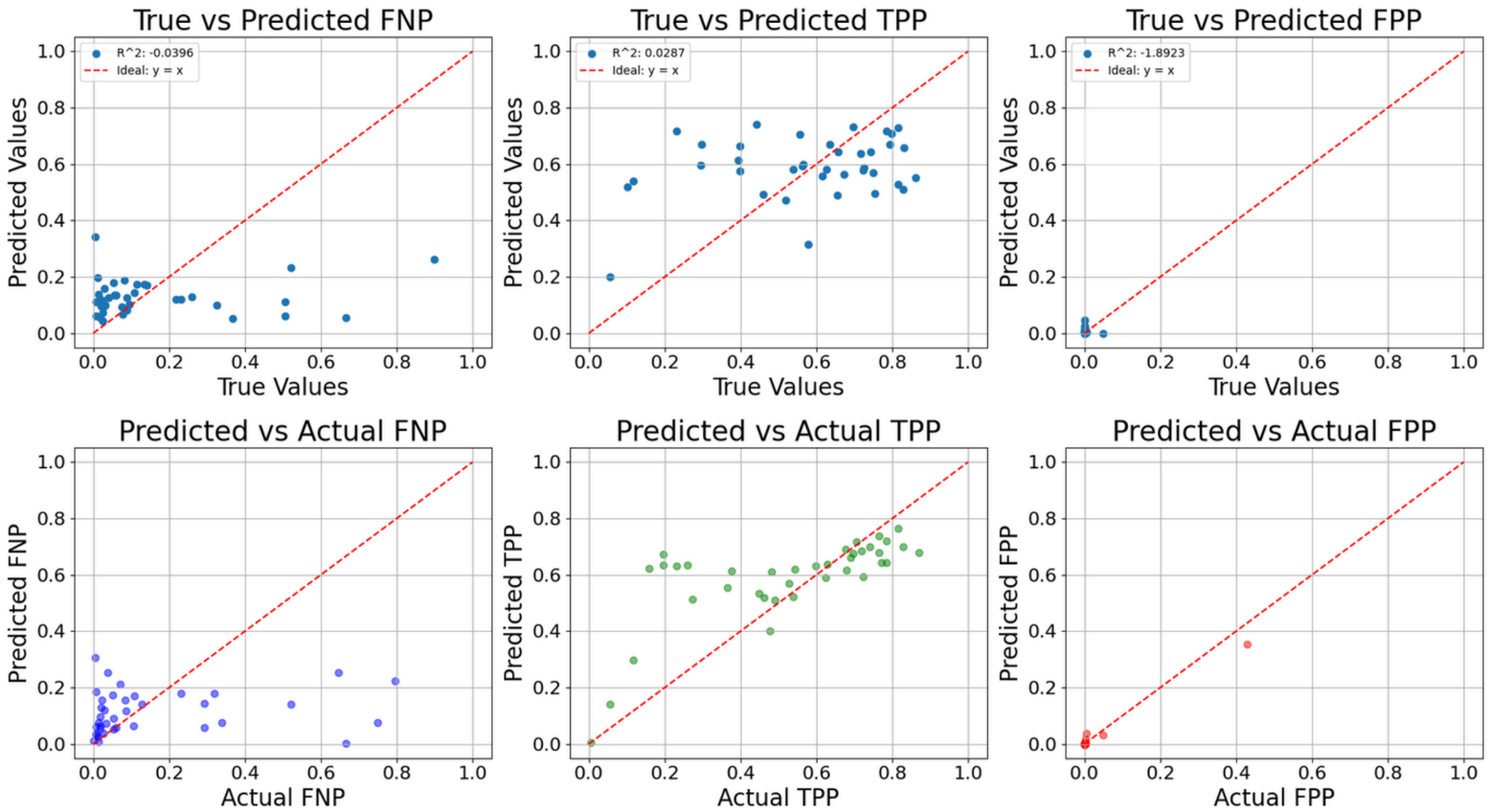
- FastSAM model segments all the objects in the image first, then filters irrelevant objects out using the prompt. If there’s an overlap between the smoke and other objects, then the bonding boxes prompt won’t be able to accurately filter out the irrelevant overlapping object (i.e., mountain)

Comparing Model Approaches

Deep Learning

Gradient boosting

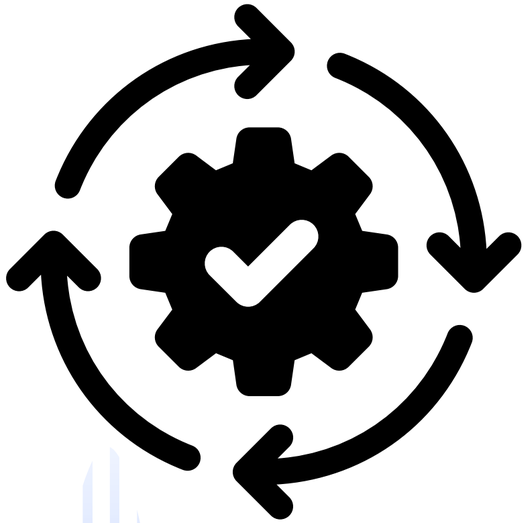
Imbalanced Dataset Problem!



Computational Efficiency

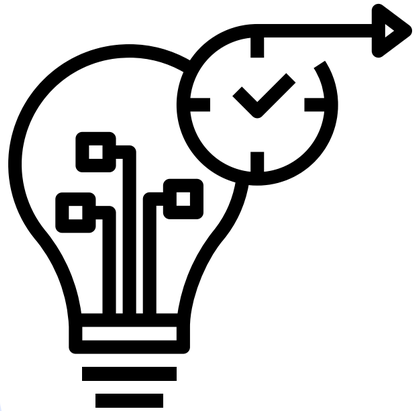
Our gradient boosting models have achieved:

- Good performance
- Fast training
- Low inference latency
- Better adaptability



Future Directions

- Address FNP and FPP prediction challenges (due to imbalanced dataset).
- Feature importance analysis can be employed to optimize the feature.
- Fine-tune FastSAM for better segmentation.
- Explore more advanced promptable segmentation models to improve segmentation accuracy.
- Implement a method to predict the smoke direction.
- Extending our approach beyond smoke to other environmental monitoring applications.



Conclusion

We integrated object detection, segmentation, and performance prediction models into a robust pipeline for monitoring wildfire smoke, with the potential to extend the system into a self-adjusting system by gauging its own errors.

Our method enables early detection and segmentation of wildfire smoke, allowing for timely alerts to be sent to authorities.

Acknowledgement

The ASEAN IVO project (http://www.nict.go.jp/en/asean ivo/index.html), titled ‘Visual IoT Network for Environmental Protection and Disaster Prevention,’ was involved in the production of the contents of this work and financially supported by NICT (http://www.nict.go.jp/en/index.html).

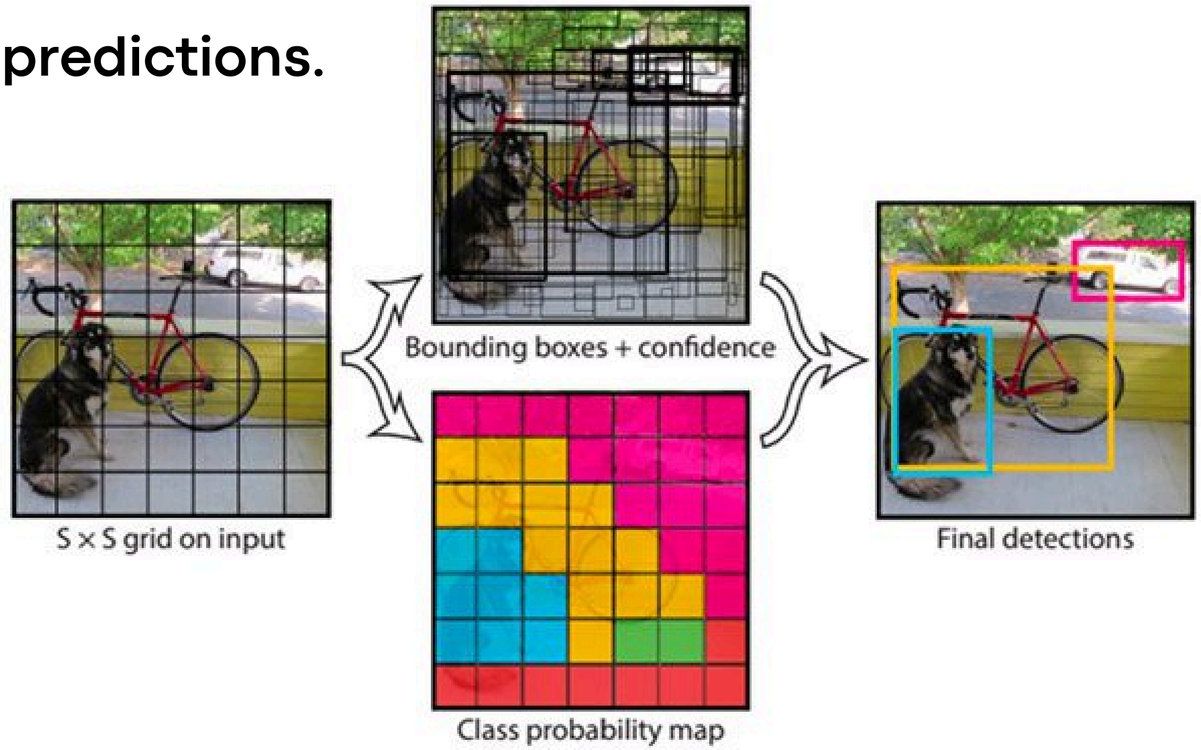
Thank you
for your attention

Q & A

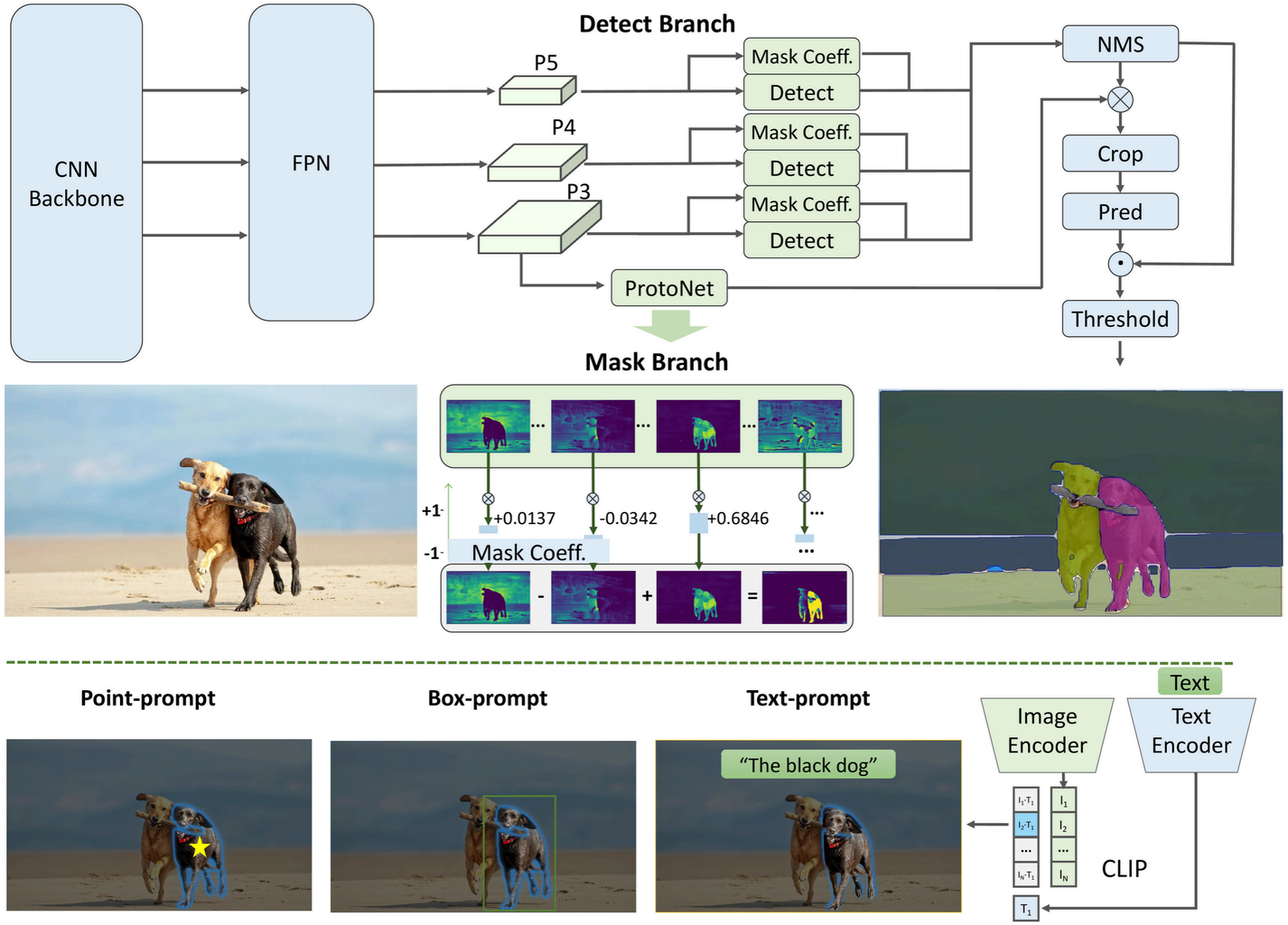
Appendix

YOLOv11

YOLOv11 is a detection model that can detect objects with a single-shot approach. It divides the image into grids and predicts the class probability for each grid. Parallelly, YOLO also predicts the bounding boxes. The class probabilities and the bounding boxes are then combine together to obtain the final predictions.



FastSAM Architecture



Feature Extraction (Color Histograms)

Color Histograms compute the distribution of colors in the RGB, HSV, and LAB color spaces to identify distinct color profiles of smoke versus foliage.

RGB: captures raw color intensities
HSV: separates chromatic information from brightness, and
LAB: approximates human perception of color differences

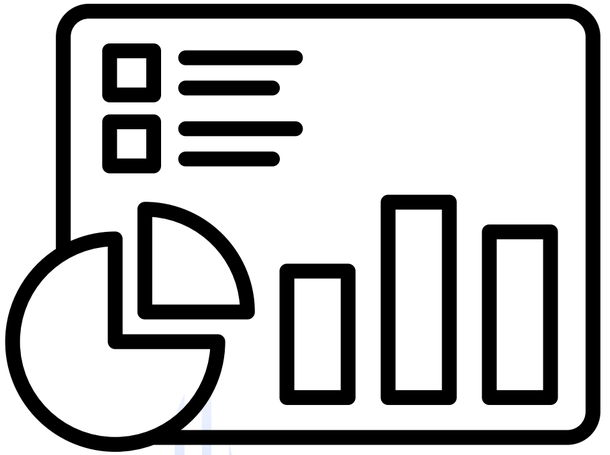
Smoke regions often exhibit unique color distributions characterized by muted tones such as white, gray, or pale blue, which contrast with the vivid greens of vegetation or the bright blues of the sky

Feature Extraction (Color Moments)

Color Moments calculates statistical measures such as mean, standard deviation, and skewness for each of the color spaces.

Mean captures the overall brightness and color tone
Standard deviation reflects color variability
Skewness identifies asymmetries in the distribution

Smoke tends to have less variability and more uniform tones compared to natural elements like trees, which we hypothesize exhibit higher color variability due to shadows and highlights



Feature Extraction (Edge Features)

Edge features assess edge density using the Canny edge detection algorithm to highlight the edges of objects.

Smoke regions are often characterized by a lack of sharp boundaries due to their diffuse and amorphous nature, resulting in a lower density of detected edges.

Feature Extraction (Texture Analysis)

Texture analysis employs the gray-level co-occurrence matrix (GLCM) to extract patterns related to texture attributes like contrast and homogeneity.

Smoke often appears as a homogeneous or low-contrast texture compared to the heterogeneous and high-contrast textures of tree canopies or other natural elements.

Feature Extraction (Color Coverage)

Color coverage measures the percentage of pixels within specific gray value ranges that indicate the presence of smoke.

Smoke typically exhibits a high concentration of some tones, which is less common in natural elements like trees or the sky.