ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

BUSINESS ANALYTICS
Master of Science

ADVANCING SAFETY WITH AI: SMOKE AND FIRE DETECTION

MACHINE LEARNING AND CONTENT ANALYTICS

Professor: Haris Papageorgiou
Assistant professor: George Perakis

Students:
Vasiliki Terizaki (p2822224)
Christos Pantoleon (p2822219)
Eleftherios Fthenos (p2822227)
Christiana Kolai (p2822213)

Athens 17/09/2023

Contents

Introduction

Our Challenge

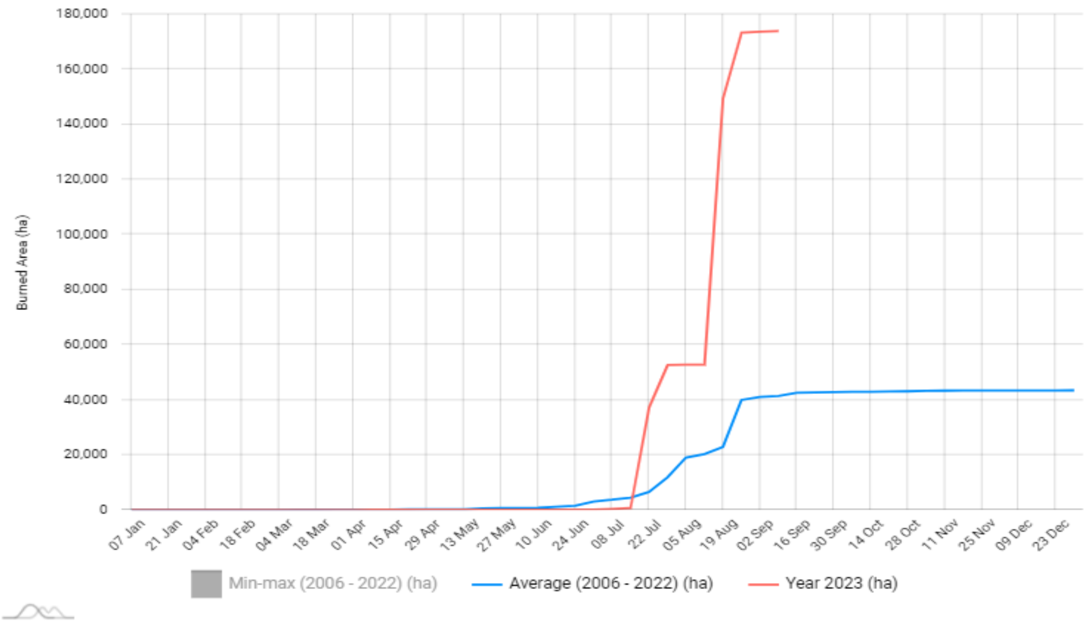
Forests are indispensable to planetary well-being, serving as lungs for the Earth, controlling climate, filtering air and water, and providing habitat to numerous species. However, these ecosystems face an ever-increasing threat due to devastating wildfires. In recent decades, the frequency and severity of these fires have been intensified, largely fueled by climate changes and human activities. The situation is dire in Europe, with Greece being one of the most impacted countries. As wildfires take a toll on human lives, economies, and ecosystems, there is an imperative need for more efficient and proactive solutions to mitigate these disasters.

According to The European Forest Fire Information System (EFFIS), 2022 was the second worst year for wildfires in Europe since 2006, with 2017 being the most catastrophic. By July 2023, wildfires had already ravaged over 182,569 hectares in the EU. Greece, Italy, Spain and Portugal have been disproportionately affected, with homes, lands, and natural reserves at significant risk. Analysis of economic damage by fires in the EU estimates losses of around 2.5 billion euros. In Greece, the surge in area destroyed by wildfires is alarming, seeing a 673% increase of burned area from 2022 to 2023 and 270% increase comparing 2023 to the average burned area of the period 2002 – 2022. The current year, the country experienced over 49 separate fire incidents, decimating around 170,000 hectares of land. This has not only had devastating ecological impacts but also economic repercussions.

Beyond the environmental costs, human suffering is heartbreaking. A total of 28 lives were lost from January to August 2023, following a peak of 103 fatalities in 2018. Communities close to these fires face losses of homes and businesses, displacement, and adverse health impacts from smoke and pollution. In terms of firefighting costs, in 2021 general government total expenditure in EU countries on fire protection services amounted to €34.1 billion, representing a 2.5% increase compared with 2020. The share in general government total expenditure was 0.5%. Overall, in the EU, government expenditure on fire protection services as a share of overall total expenditure remained stable at around 0.4 – 0.5% since the beginning of the time series in 2001. The share of government expenditure on fire protection as a ratio of total expenditure varies among EU countries. In 2021, Greece among other countries had one of the highest share with 0.6% of the total expenditure.

Current response mechanisms, largely reactive in nature, are insufficient in preventing the loss of life and property. Delayed response times and the high costs associated with firefighting efforts post-ignition underscore the necessity for proactive measures. With climate change worsening drought conditions and increasing the likelihood of wildfires, a new approach to fire prevention and management is urgently required.

Greece faces an uphill battle in safeguarding its natural landscapes and rural communities. The escalating frequency and devastation of wildfires necessitate a strategic rethinking of current management practices. The situation calls for local, national, and international bodies to collaboratively invest in innovative, efficient, and proactive solutions. These must aim not only to prevent fires but also to minimize their impact when they occur, thereby protecting both human lives and the critical ecosystems that sustain us all.



Data source: EFFIS

Figure 1. Weekly cumulative burnt areas in Greece, measured in hectares.

Country	Country area (ha)	Date Range	Annual Avg. (ha)	Year 2023 (ha)	(*) Annual Avg. (%)	(*) Year (%) 2023
Portugal	9187803	[2006 - 2022]	97081.53	29662	1.06	0.32
Spain	50604375	[2006 - 2022]	81058.65	84733	0.16	0.17
Italy	30075506	[2006 - 2022]	54243.12	70342	0.18	0.23
Greece	13257480	[2006 - 2022]	43489.76	173918	0.33	1.31

Data source: EFFIS

Table 1. EU Burned Area: Greece 4th place from 2006-2022.

Country	Country area (ha)	Date Range	Annual Avg. (ha)	Year 2023 (ha)	(*) Annual Avg. (%)	(*) Year (%) 2023
Greece	13257480	[2006 - 2022]	43489.76	173918	0.33	1.31
Spain	50604375	[2006 - 2022]	81058.65	84733	0.16	0.17
Italy	30075506	[2006 - 2022]	54243.12	70342	0.18	0.23
Portugal	9187803	[2006 - 2022]	97081.53	29662	1.06	0.32

Data source: EFFIS

Table 2. EU Burned Area: Greece 1st place in 2023.

Our Project

At present, fire detection heavily relies on terrestrial and satellite monitoring. While these methods have their merits, they can be slow, covering vast areas in a less timely manner. Moreover, these methods sometimes miss small fire outbreaks that can rapidly escalate. The European Union, grappling with this issue, spends a staggering €2 billion annually just on firefighting, not to mention the costs of rebuilding, reforestation, and compensation. This situation has led to an initiative of developing a solution that will help to prevent forest fires from escalating rapidly and thus lives will be saved, the ecosystem will be protected, and resources and infrastructure will be preserved.

This initiative refers to the design and training of an AI model which will be able to detect fire and smoke in forestlands and broadcast warning signals. This model will take as input snapshots or video streams from Unmanned Aerial Vehicles (UAV). UAVs equipped with imaging capabilities can swiftly traverse challenging terrains and reach remote or hazardous locations with ease, providing invaluable real-time data that can be used by firefighting teams and will help the authorities to make well-informed decisions.

By detecting fires at their nascent stages and monitoring the existence of smoke and flames from aerial vantage points, these UAVs offer early warnings, enabling rapid response and containment efforts. Moreover, the ability of UAVs to operate autonomously or be remotely piloted reduces the risk to human firefighters, ensuring their safety during dangerous firefighting operations. Embracing UAV technology in fire and smoke detection not only enhances the efficiency and effectiveness of firefighting efforts but also empowers disaster management agencies to proactively tackle wildfires, protecting lives, ecosystems, and critical infrastructure from the devastating consequences of uncontrolled infernos.

Type	Configuration	Endurance	Data collection altitude (agl)	Max range (miles)	Typical sensors
1	Fixed-wing	6–14 hours	3,500-8,000	50	EO/Mid wave IR
	Rotorcraft	NA	NA	NA	High quality IR
2	Fixed-wing	1–6 hours	3,500-6,000	25	EO/Long wave IR
	Rotorcraft	NA	NA	NA	Moderate quality IR
3	Fixed-wing	20-60 min.	2,500 and below	5	EO/IR video and stills
	Rotorcraft	20-60 min.	2,000 and below	5	Moderate quality IR
4	Fixed-wing	Up to 30 min.	1,200 and below	<2	EO/IR video and stills
	Rotorcraft	Up to 20 min.	1,200 and below	<2	Moderate quality IR

Table 3. Types of UAVs and their characteristics.

These flying machines revolutionize the way we tackle fire prevention. Here's why drones could be the future of fire management:

- Mobility: Unlike the limited view of fixed cameras or the delayed input from satellites, drones swiftly scan diverse terrains, offering an all-encompassing bird's-eye perspective.
- Real-time Monitoring: With high-end sensors, drones transmit real-time information, allowing immediate decision-making by authorities.
- Economic Efficiency: Beyond the initial investment, drones are relatively cheaper to operate.

With their efficient coverage, drones can substantially cut firefighting expenses. Adopting drones for widespread fire surveillance might reduce these costs by an impressive 20-30%. The high-quality visual data our drones provide doesn't stop being valuable after a fire. It serves to evaluate the extent of the damage and inform recovery strategies, providing critical assistance for insurance claims and post-fire investigations.

Stremlit App (Fire and Smoke Detection using YOLOv8 & YOLOv7)

While the model's capabilities are extensive, we recognize that technology is most impactful when it is accessible and easily understood by a broad range of users. With this understanding, we have chosen to take our innovation a step further by developing a user-friendly application using Streamlit[1].

[1] https://forestfiresmokedetection.streamlit.app/#fire-and-smoke-detection-using-yolov8-yolov7

For business purposes, this application serves as a proof-of-concept that accomplishes three critical goals:

- **Rapid Evaluation and Testing**
  First, it allows us as developers to quickly evaluate and test our model's performance, ensuring that we are continuously iterating on its capabilities. For example:
- In scenarios where drone footage has questionable elements or false positives/negatives, an expert can quickly upload the image to the app for additional verification. In questionable alerts, we might decide to send a ground team for further investigation before committing more significant resources.
- Our model will likely go through multiple iterations and improvements. The app allows us to easily switch between different versions of our model for comparative analysis.

- **User-Friendly Tool for Non-Experts**
  Second, and perhaps more importantly, the simplicity of a drag-and-drop interface makes this technology more accessible to non-technical stakeholders. By demonstrating this app, it's easier to secure buy-in from stakeholders.
- The current app serves as a prototype for market validation, enabling us to gather user feedback and possibly attract initial investors or grants for further development.
- By starting small, we can manage costs and risks while still showcasing the core technology and its potential benefits.

- **Initial Data Collection and Analysis**
  Third, this app can serve as a pilot stage for collecting a more extensive dataset, as users can easily upload images for analysis. It could also be an initial step toward crowd-sourced data collection.
- Machine learning models like YOLO-v8 improve with more data. If the model makes an error, we can use that information to refine the algorithm.
- For something as serious as fire detection, collecting enough diverse and real-world data to train our model effectively, is very challenging. The app can be opened up to the public or specific user groups (like forest rangers, environmentalists, etc.) to upload images they capture of fires or potential fire zones.

Our Vision

Our primary goal is to revolutionize fire management by providing a proactive solution to forest fire detection and monitoring. We aim to leverage cutting-edge technology to mitigate the risks and impacts associated with forest fires, thereby protecting our ecosystems and communities.

We also aim to increase the speed and accuracy of our drones, for forest fire detection. By doing so, we will facilitate rapid response and firefighting efforts, which can significantly reduce the loss of life, property and environmental damage caused by fires.

Our project's evolution doesn't end with its current state. We recognize that fires occur due to numerous factors. Thus, our long-term vision encompasses drones sophisticated enough to consider:

- Environmental Metrics: Vegetation type, soil organic content, moisture, and structure.
- Ignition Sources: Human activities, lightning, and escaped fires.
- Fuel Metrics: Condition, continuity, amount, and management.
- Topographical Aspects: Slope, elevation, and aspect.
- Weather Indicators: Wind patterns, air temperature, relative humidity and solar radiation.
- Fire Behavior Analysis: Intensity, spotting, spread rate, and flame height.
- Impact Mitigation: Comprehensive fire management, evacuation strategies, and recovery planning.

Wildfires arise from a combination of environmental, weather-related, physical, and human influences. These elements together determine the likelihood, behavior, duration, size, and impact of the fire. With changes in many of these factors, the risk of wildfires is growing worldwide. For example, climate change is leading to more frequent and intense conditions that favor wildfires, and shifts in population in high-risk areas are raising the potential consequences of these fires

In the longer term, we picture our solution being adopted across various sectors - from residential and commercial buildings to defense and military installations. Our scalable solution can either be integrated with existing systems or serve as a standalone solution, providing immense flexibility and potential for growth.

Methodology

Given the challenges of processing, and understanding the varied visual indicators in images captured by drones, especially in dynamic environments like forests with potential fire or smoke, selecting the right models and techniques becomes paramount. Deep learning, leveraging artificial neural networks, offers a potent solution to handle this complexity, making it a go-to approach for image-based tasks like object detection. Given the rapid and unpredictable nature of fires, prioritizing faster processing speeds for early detection is essential. Hence, the YOLO family of models was selected over others due to its capacity for rapid predictions, stemming from its single-pass image processing architecture. In our specific scenario, we evaluated YOLO-v8, the current state-of-the-art model, along with its predecessor, YOLO-v7. The former is more lightweight and faster, while the latter provides richer feature extraction. Eventually, the design and efficiency of YOLOv8, tailored for object detection tasks, allowed us to achieve real-time and accurate detection of forest fires and smoke, making it the model of choice for our drone-based solution.

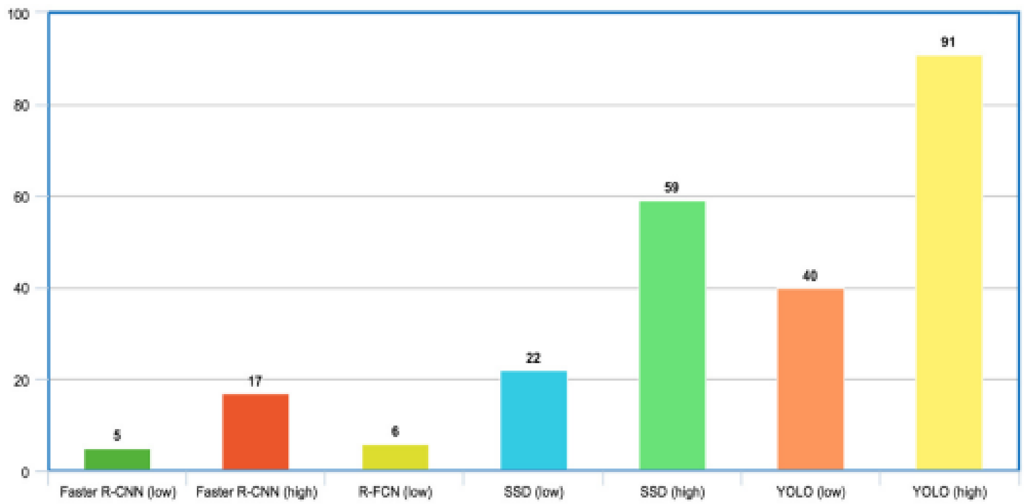


Figure 2. Comparison of frames per second.

Data Collection & Overview

Gathering the right input data is an essential first step before we begin training different models and assessing their effectiveness. In our quest to find the most suitable dataset, we looked into various data sources. Our pursuit initially led us to utilize a dataset from Kaggle[2], containing various forest images, some with smoke and fire and others without. The dataset includes 4.468 images of fire,

[2] Kaggle Dataset (https://www.kaggle.com/datasets/kutaykutlu/forest-fire?select=train-smoke)

smoke and no fire situations. There have been created three separate folders, containing corresponding photos of each category. The labeling and annotation process were executed via Roboflow. We annotated these images, aiming to prepare a reliable training set for our model. However, despite the efforts in parameter tuning, data configuration, and experimenting with varying epochs, the models did not produce satisfactory results. Upon further examination, we determined that inaccuracies in the annotation process might have been a significant contributor to the low performance. With time constraints and the need for precision, we decided to pivot to a pre-annotated dataset from Roboflow[3]. This dataset, comprising 1.706 images, includes both close-up shots and aerial views of fires and smoke, offering a diverse range of perspectives crucial for drone-based fire detection.

Data Processing & Annotation

After collecting our initial dataset, we proceeded to the phase of data refinement and preparation. To ensure the quality of our dataset, we first filtered out images with insufficient resolution, focusing our efforts on those with higher quality. Following this initial curation, we applied a series of essential data pre-processing steps, which included annotation and augmentation, using the tool Roboflow. In the annotation stage, we defined the two primary classes we intended to work with: 'fire' and 'smoke' and then we proceeded to label each image accordingly, providing a clear and structured foundation for our subsequent tasks. During the augmentation phase, we employed a variety of image enhancements to improve the diversity and robustness of our dataset. This involved making adjustments to elements such as image orientation, colors, lighting conditions, blurring and cropping, resulting in a more comprehensive and representative collection of images. Finally, to facilitate the training and evaluation of our machine learning models, we partitioned the images resulting from these preprocessing steps into three distinct sets: the training set, the validation set, and the test set. This separation ensured that our models can be effectively developed, fine-tuned, and evaluated, ultimately enhancing their performance in detecting and classifying instances of 'fire' and 'smoke' in real-world scenarios.

Since our YOLO models didn’t perform as expected, we experimented again with data augmentation techniques and also reevaluated if the length of each of our classes was balanced. Despite that, our models still didn't perform well and this led us to conclude that the issue likely lied in the annotation of 'fire' and 'smoke,' possibly due to inaccurate bounding boxes or inconsistent labeling, thus affecting the model's learning and detection.

Algorithms, Computer Vision Models & Architectures

In the methodology there is extensive usage of the YOLO family models. YOLO, which stands for "You Only Look Once," is a series of artificial intelligence models designed for object detection and classification in the field of computer vision. These models are highly regarded for their ability to perform real-time object recognition, making them useful in various applications.
YOLO belongs to the category of one-stage object detection models, which means that it processes an entire image in a single forward pass through a deep convolutional neural network (CNN). This is in contrast to two-stage detection models like R-CNN, which follow a more complex workflow. In greater detail the basic steps of YOLO model family are:

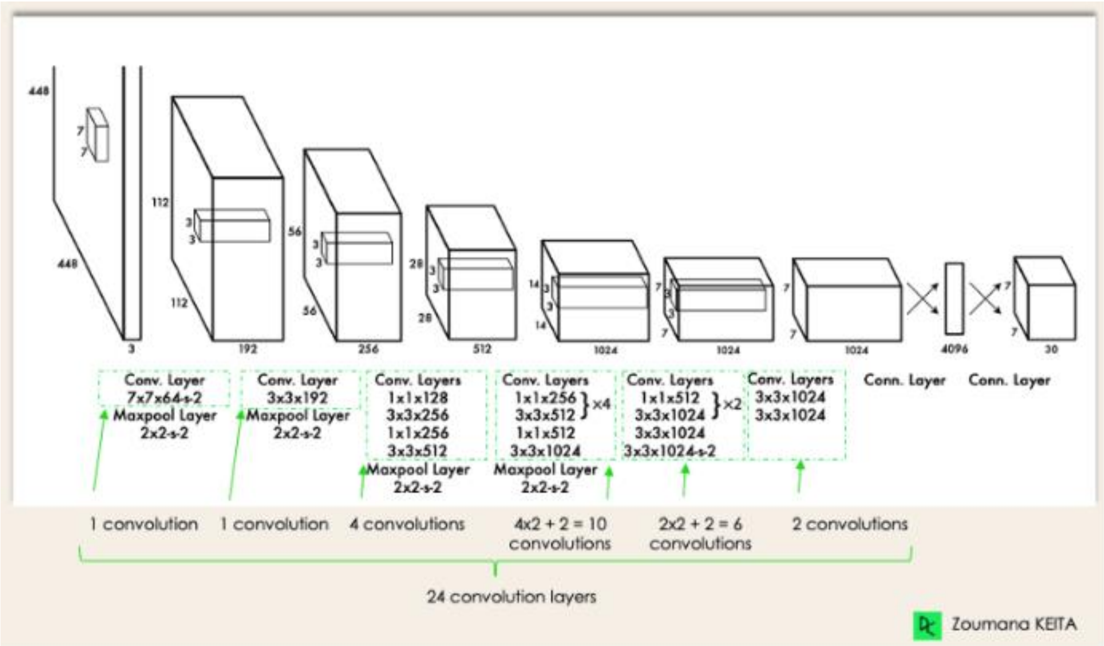
- Single Forward Pass: YOLO processes the entire input image in one go, without the need for multiple passes or iterations. This single forward pass through the CNN is a key feature that enhances its speed and efficiency.
- Grid-based Approach: YOLO divides the input image into a grid of cells, typically, a fixed grid size like 7x7 or 9x9. Each grid cell is responsible for predicting objects present in its corresponding region.

---

[3] Roboflow Dataset (https://universe.roboflow.com/kirzone/fire-iejes/browse?queryText=class%3A%22Marked+Null%22&pageSize=50&startingIndex=100&browseQuery=true)

- Bounding Box Prediction: In each grid cell, YOLO predicts multiple bounding boxes (rectangles) that might contain objects. For each bounding box, it predicts the coordinates (x, y) of the box's center, its width, height, and a confidence score that indicates the probability of an object being present within that box.
- Class Probability: YOLO also predicts class probabilities for each bounding box, representing the likelihood of the object belonging to a particular class (e.g., "car," "person," "dog" etc.).
- Non-Maximum Suppression (NMS): After the predictions are made, YOLO applies a post-processing step known as Non-Maximum Suppression. This step filters out redundant or overlapping bounding boxes and retains the most confident ones, ensuring that each object is only detected once.

The advantages of YOLO models, besides their real-time capabilities, include their simplicity as they eliminate the need for multiple stages of processing. However, they may be less accurate than two-stage models like R-CNN in certain scenarios. Nonetheless, YOLO's efficiency and speed make it a popular choice for applications like smoke and fire detection, where real-time object recognition is crucial. Over the years, YOLO has seen several iterations and improvements, with each version aiming to enhance both accuracy and speed, making it a significant tool in the computer vision field.



Data source: [DataCamp]

Figure 3. YOLO models' architecture.

Among the multitude of machine learning YOLO models available, we've chosen to pair our drones with versions 7 and 8 algorithms. Both YOLOv7 and YOLOv8 have been noted for their scalable architecture and improved object detection techniques, making them strong candidates for real-time detection tasks. Both were trained and tested on the same dataset with similar hyperparameters and number of epochs in order for their results to be comparable.

Training Overview of both models

Both YOLO-v7 and YOLO-v8 models were trained on the following dataset and training parameters:

- **Dataset**:
  Training Dataset: 1194 images of both fire and smoke instances

Validation Dataset: 340 images of both fire and smoke instances
Testing Dataset: 172 images of both fire and smoke instances

- **Number of Epochs**: 30
- **Adjusted Images Side:** 640 x 640 px
- **Batch Size:** YOLOv7: 16 - YOLOv8: 20
- **Learning Rate (LR)**: 0.01
- **Training Platform:** Google Colab instance with a T4 GPU

The model weights and training hyperparameters can both be found on the corresponding Github Resource of this project.

YOLO-v7

The YOLO-v7 model was launched in the middle of 2022 by Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao and became fast popular for object detection tasks as it has a faster and stronger network architecture that provides a more effective feature integration method, more accurate object detection performance, a more robust loss function, and an increased label assignment and model training efficiency. In our case we used a pre-trained model where the weights and the layers were freezed in order not to be updated during the training. This model was further trained using images in our dataset and learned how to specifically detect smoke and fire. The training process of the YOLO-v7 was pretty straight-forward and simple, and can be found on the relevant Google Colab Notebook with the full analysis of each step of the process.

The model was trained using a batch size consisting of 16 images and was subjected to 30 training epochs to optimize its performance.
Considering the computational resources at our disposal, the total duration required to complete the training process was approximately 42 minutes. This timeframe is indicative of the model's efficiency and is a crucial metric for evaluating the overall effectiveness of the ML pipeline.

After the training process was completed, we ran an evaluation step on the validation dataset, to observe how well the model performs on data it has not been trained on, providing insights into its generalization capabilities. Furthermore, the performance on the validation set can serve as an estimation of how well the model will perform on the test set, or in a production environment.

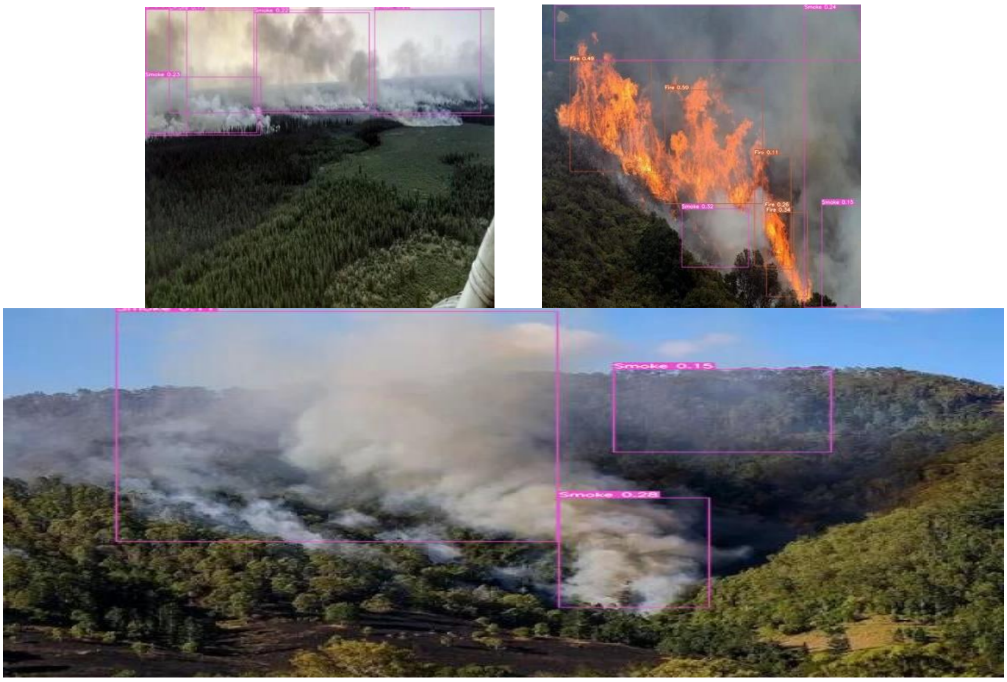


Figure 4. Bounding boxes' results on the validation dataset of YOLO-v7 model.

YOLO-v8

The state-of-the- art model YOLO-v8 was launched in the beginning of 2023 by Ultralitycs and came with new features that achieve higher accuracy and faster training than the previous versions. For the detection of smoke and fire, the model is not trained from scratch but again transfer learning was performed, as we did with YOLO-v7.
We adopted a training approach similar to that of YOLO-v7 for our second model, which was based on the YOLO-v8[4].
It is noteworthy that the YOLOv8 prototype model was developed by Ultralytics, in contrast to YOLO-v7, which was a collaborative research initiative. Its training process and algorithm was a bit different command-wise. Specifically, YOLO-v7 was trained using Command-Line Interface (CLI) commands, while the YOLO-v8 model was trained through Python scripts, adhering to the guidelines provided in Ultralytics' official documentation.
In a similar matter, we once again procured our custom dataset from Roboflow. Additionally, we leveraged pretrained weights from the YOLO-v8 Medium Model (see Appendix C on YOLOv8 Medium Model Characteristics) to use for our training process. After setting things up, we started training the model for 30 epochs, using groups of 20 images each time. It's worth mentioning that the Ultralytics library's "model.train" command does two things at once: it trains the model on the new data and also checks its performance on the validation dataset, after a certain number of steps. This dual action helps us to make sure that the model is learning effectively.

Given our computational resources, the model training took approximately 2.5 hours. This timeframe is notably longer than what was experienced during the YOLOv7 training phase. Some possible reasons for that training-time difference include the following:

1. **Architecture Complexity:** YOLOv8 may have a more complex architecture compared to YOLOv7, leading to higher computational costs per training iteration.
2. **Hyperparameter Differences:** Our analysis revealed key numerical differences also in hyperparameters[5]. For instance:
   Final Learning Rate: YOLOv8 used a final learning rate of 0.05, compared to 0.1 in YOLOv7. A lower final learning rate could slow down the convergence, potentially requiring more epochs to reach optimal performance.
   Weight Decay: YOLOv8 has a weight decay of 0.001, which is twice as high as YOLOv7's 0.0005. A higher weight decay acts as a stronger regularizer, which could slow down the training convergence.
   Box Loss Weight: YOLOv8 has a significantly higher box loss weight of 7.5, compared to 0.05 in YOLOv7. This means YOLOv8 places more emphasis on getting the bounding boxes correct, which could require more computational effort per iteration.

Upon initial observation, it is apparent that while the model excels in detecting instances of fire, its performance in identifying smoke within the images is somewhat lacking. This observation suggests two potential courses of action for improvement: one option could be to freeze the current model weights and conduct additional training exclusively on images containing smoke. Alternatively, a hybrid approach employing both the YOLOv7 and YOLOv8 custom models on the Unmanned Aerial Vehicles (UAVs) could be considered to achieve more reliable and accurate predictions. However, it's important to note that such a hybrid approach may result in a decrease in processing speed, given that the image data would be subjected to multiple layers across the two models. This trade-off between predictive accuracy and operational efficiency warrants careful consideration.

---

[4] The Google Colab Notebook for training the model can be found in the following address. (https://colab.research.google.com/drive/1oOhKRR0QGHGdBYt3ru9HHZj8VXdTlAv3)

[5] YOLOv7 Model Training Results: https://github.com/Lefyd24/Forest-Fire-Smoke-Detection/tree/main/models/yolov7/runs/train/exp
YOLOv8 Model Training Results: https://github.com/Lefyd24/Forest-Fire-Smoke-Detection/tree/main/models/yolov8

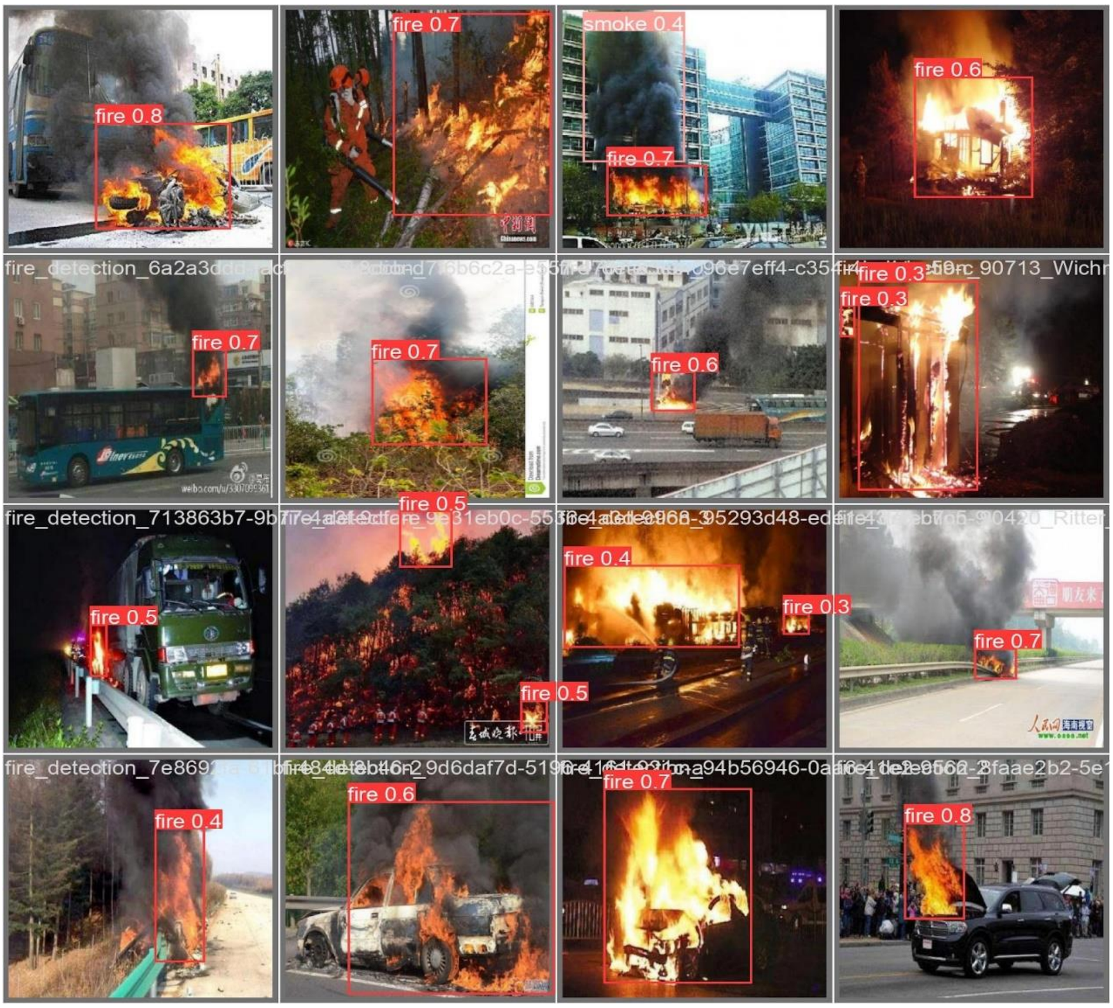


Figure 5. A sample batch of images' predictions along with their corresponding precision, during the validation process of the YOLO-v8 model.

Experiments, Setup & Configuration

Results & Quantitative Analysis

Through a comprehensive analysis of the results obtained from both models, we aim to discern which one stands as the superior choice for our task. Our evaluation process comprises a multifaceted approach, including metrics such as detection accuracy, speed, and robustness across diverse environmental conditions. By examining these key factors, we seek to make an informed and data-driven decision regarding the optimal model for forest fire detection.

Regarding YOLO-v7, the model's performance improves progressively over epochs, as evidenced by decreasing validation losses and increasing accuracy metrics. Notably, it exhibits a stronger capability in detecting fire instances compared to smoke, as indicated by higher precision and recall scores for the "fire" class. Analyzing the results presented in Table 2, it becomes apparent that the model's performance in detecting fires is moderately successful, with 37% of actual fires correctly predicted. However, the model misses capturing 63% of existing fires, resulting in false negatives. In contrast, the model's performance in detecting smoke instances is less effective, with only 16% of actual smoke instances correctly identified. The majority, 84%, of photos containing smoke are incorrectly classified as background, leading to a high rate of false negatives. In images devoid of both smoke and fire, the model exhibits a notable rate of false positives, with 70% being false positives for fire and 30% for smoke. This suggests that the model tends to incorrectly identify fire or smoke in images where neither is present.

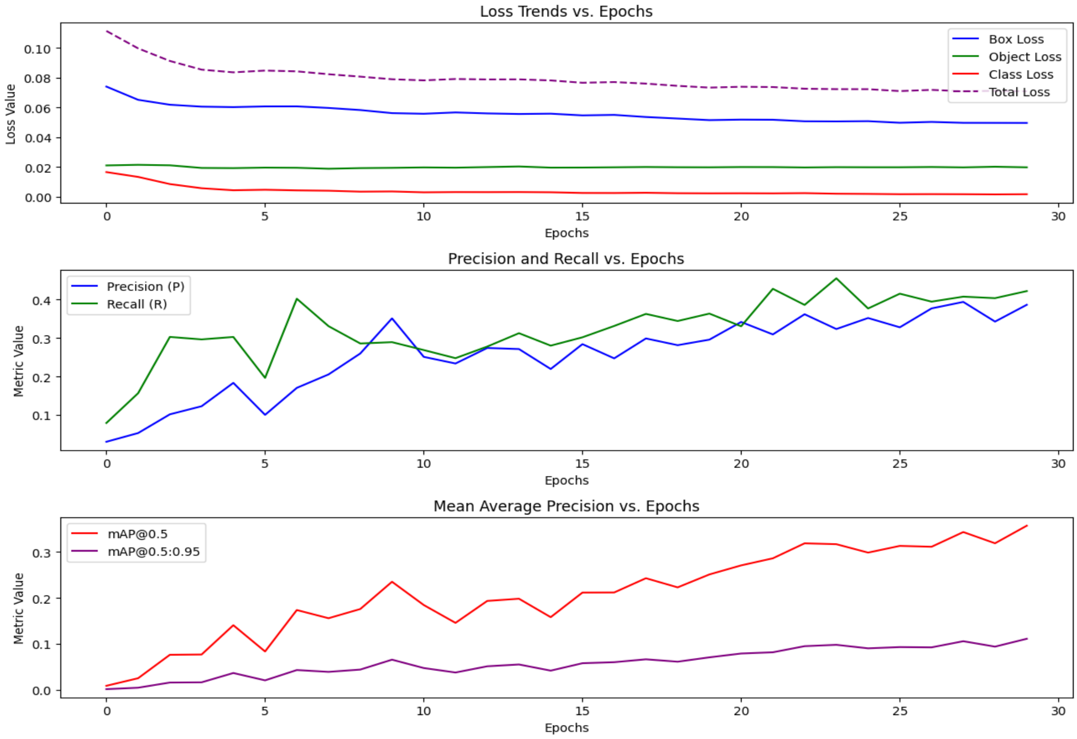


Figure 6. Validation losses, precision - recall and mean average precision curves over the epochs.

In Figure 6, the curve appears to be relatively smooth but closer to the lower-left corner of the plot. This is generally not a positive indication, as it means that the model is not able to achieve high precision and recall. Although the exact AUC (Area Under the Curve) value is not shown in the image, a higher AUC for the PR curve signifies a better model. The displayed curve does not seem to occupy a significant portion of the plot's area, suggesting a relatively small AUC. The curve's trajectory provides insights into the trade-off between precision and recall. A model with a perfect PR curve would reach the top right corner (Precision = 1, Recall = 1). This curve's proximity to the lower point suggests that the model can achieve a moderate balance between minimizing false positives (high precision) and maximizing true positives (high recall).

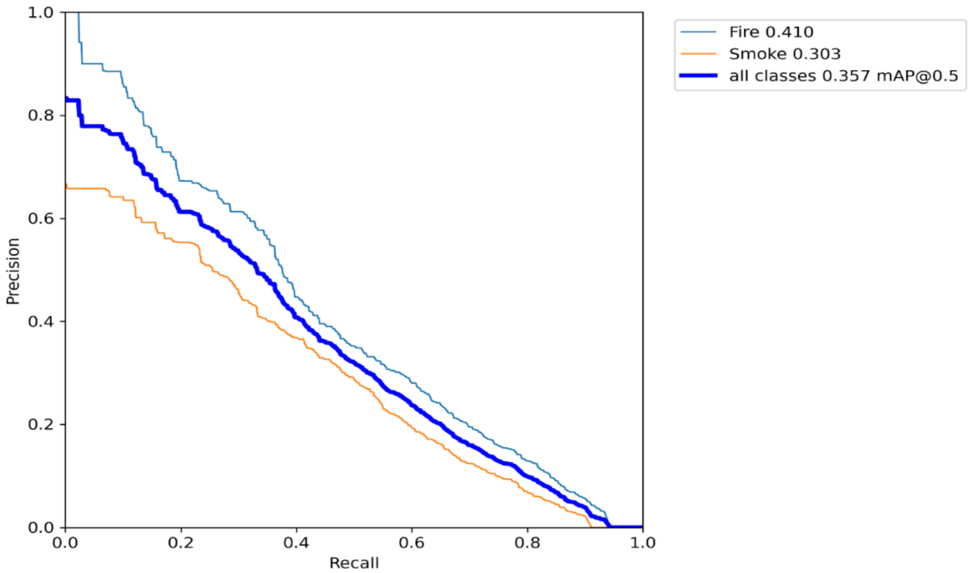


Figure 7. Precision and recall for the prediction of the fire and the smoke of YOLO-v7.

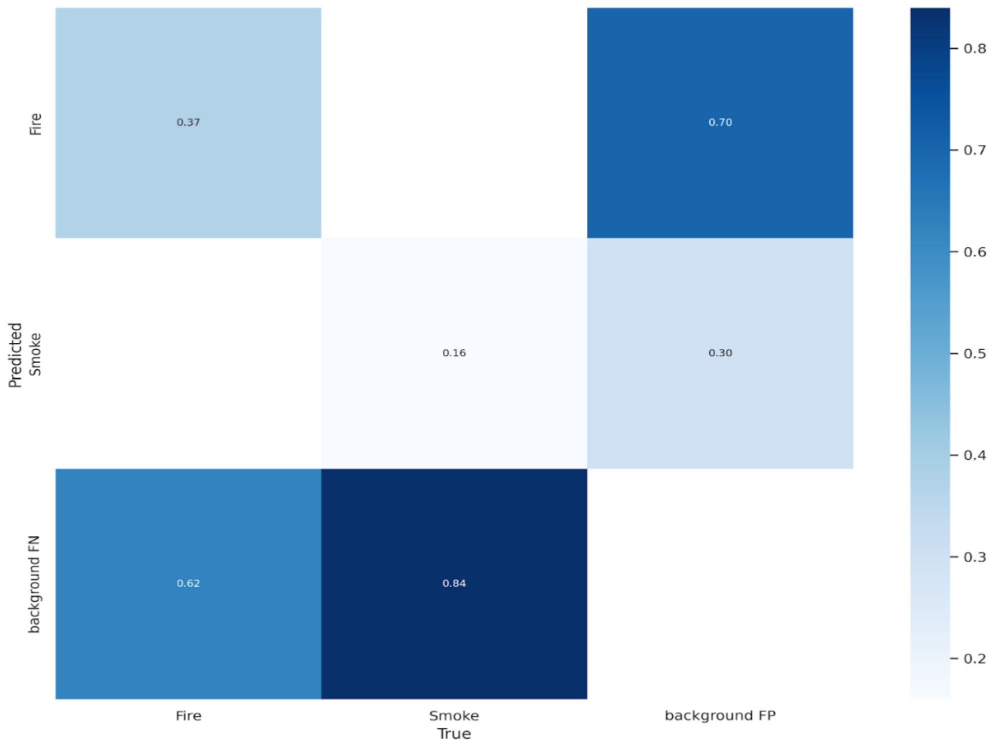


Table 4. Confusion matrix of YOLO-v7 model.

When evaluating YOLO-v8, it's evident that the model undergoes a progressive learning process, resulting in improved performance with each epoch. This is clearly reflected in the decreasing training and validation losses, encompassing box loss, class loss, and total loss. Notably, there are no indications of overfitting, as the validation losses remain stable or decrease while the training losses continue to decline. Additionally, accuracy metrics exhibit a consistent upward trend, reinforcing the model's learning capacity. Turning the attention to the results presented in Table 3, several key insights emerge. For images containing fires, the model correctly predicts 42% of them while missing 58% (false negatives), meaning that a significant proportion of actual fire instances go undetected by the model. Similarly, for images containing smoke, the model correctly identifies 29%, with 71% of smoke instances being missed (false negatives), underscoring a challenge in smoke detection. In images devoid of both smoke and fire, there is a relatively high rate of false positives, with 78% being false positives for fire and 22% for smoke. Moreover, it's worth noting that the model's ability to predict smoke instances is on par with its performance in predicting fires, indicating a balanced approach in detecting both fire and smoke as in Figure 8 the two smooth lines do not abstain much from each other.

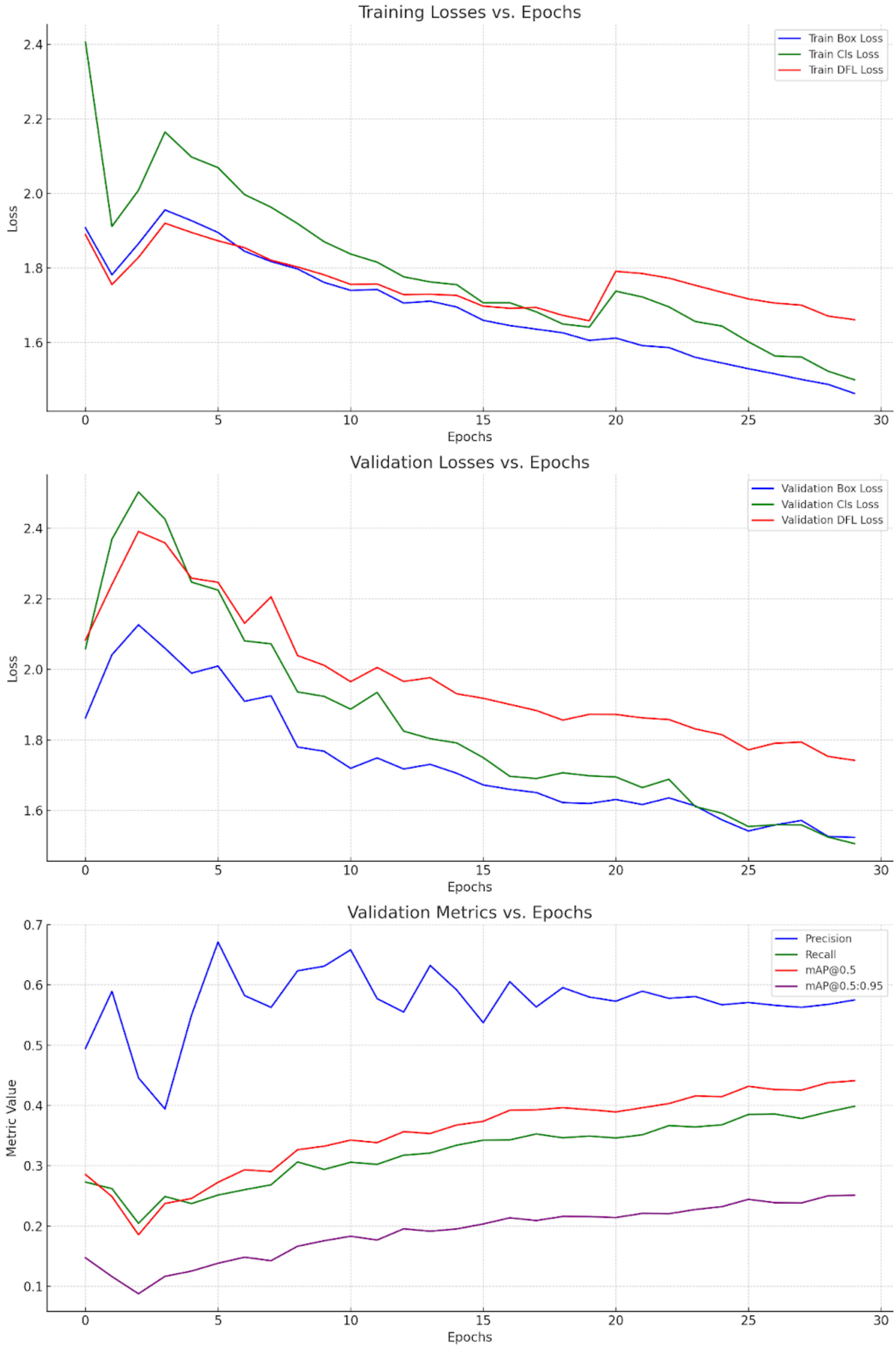


Figure 8. Training losses, validation losses and accuracy metrics over epochs of YOLO-v8 model.

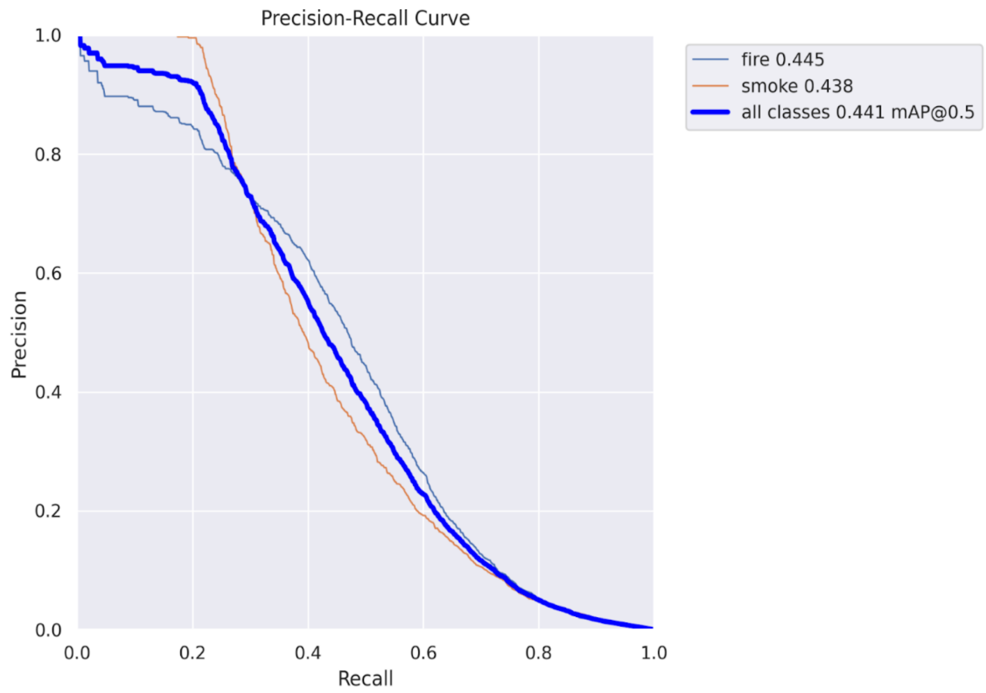


Figure 9. Precision and recall for the prediction of the fire and the smoke of YOLO-v8.

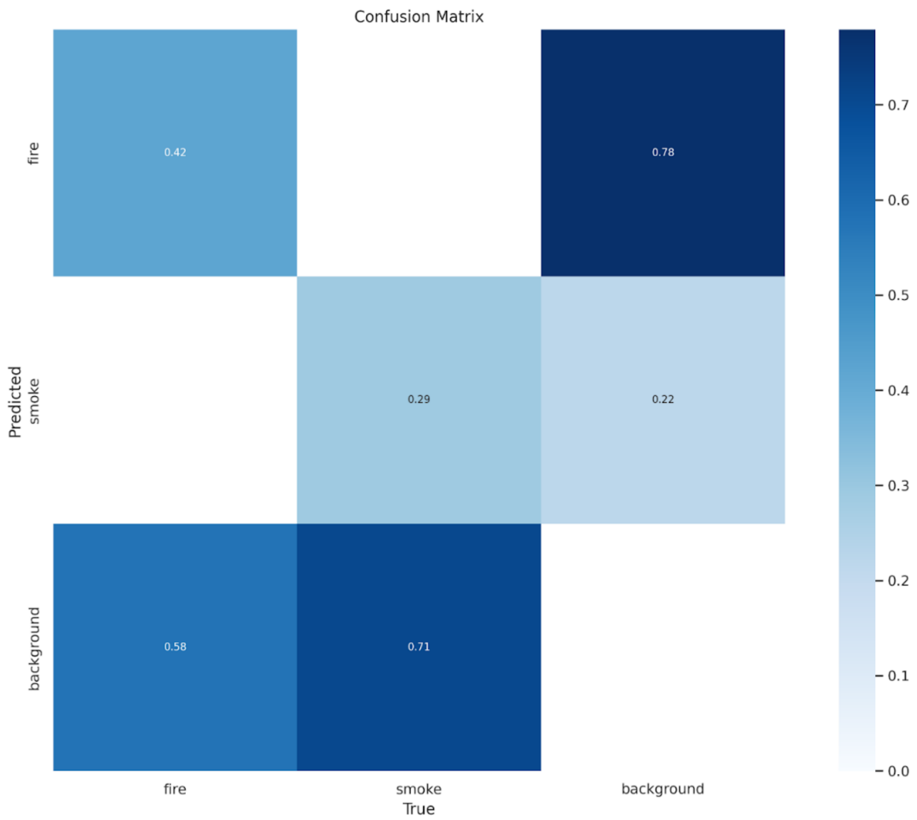


Table 5. Confusion Matrix of YOLO-v8 model.

In comparing YOLO-v7 and YOLO-v8 for forest fire detection, both models exhibit a progressive learning process with decreasing losses and improving accuracy metrics over time. YOLO-v8 holds a slight advantage by correctly identifying 42% of actual fires and 29% of actual smoke, while

YOLO-v7 identifies 37% and 16% respectively. Both models face challenges in smoke detection as the percentages are quite low but YOLO-v8 is slightly outperforming YOLO-v7. In images lacking both smoke and fire, both models exhibit a propensity for false positives, with YOLO-v8 producing a higher rate, but in images with existence of the two classes YOLO-v7 produces higher rate of false negatives which is more serious. Notably, YOLO-v8 showcases a balanced approach in detecting both fire and smoke, while YOLO-v7 displays a more pronounced disparity in performance between these classes. Overall, YOLO-v8 demonstrates a marginally better overall performance, but both models indicate room for improvement, particularly in reducing false negatives and false positives, reinforcing the need for further optimization in forest fire detection scenarios.

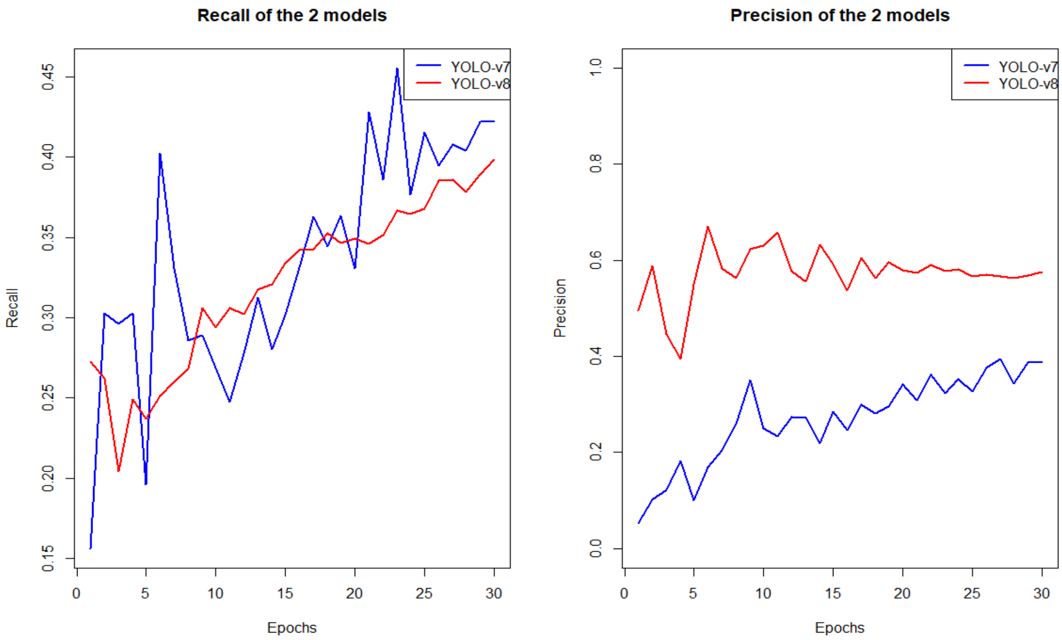


Figure 10. Recall and Precision curves of YOLO-v7 and YOLO-v8.

One more notable advantage of using YOLO-v8 over YOLO-v7 is its significantly reduced processing time. We efficiently performed a test over a subset of 2.750 images in order to further investigate and evaluate this claim. As mentioned, YOLO-v8 outperformed YOLO-v7 in terms of computational efficiency, demonstrating considerably faster execution times. The short timeframe of completing this particulate prediction task is quite important since the alerts that need to be sent as well as the actions that need to take place, have to be immediate.

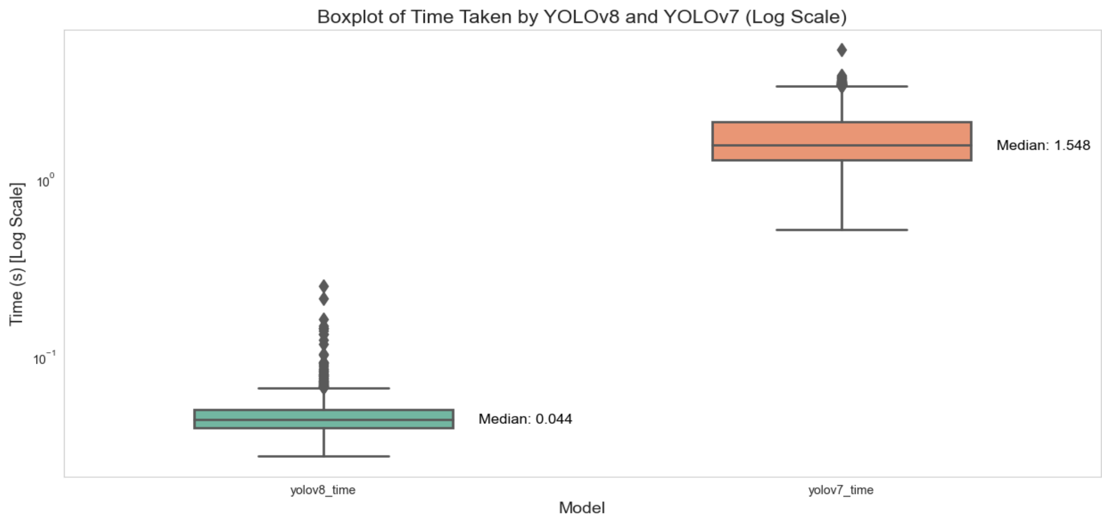


Figure 11. Boxplots of processing timeframe of the models.

Qualitative & Error Analysis

Considering the critical nature of the smoke and fire detection task, with the imperative of promptly notifying authorities, it becomes necessary to conduct a comprehensive error analysis of our YOLOv8 model. This analysis aims to pinpoint situations in which the model generates false alerts, either by mistakenly indicating the presence of smoke or fire when there isn't (false positives) or by failing to recognize their presence when they actually exist (false negatives).
As evidenced by the data presented in Table 3, which showcases the confusion matrix generated during the training phase of YOLOv8, there are notable patterns in the model's performance with regard to fire and smoke detection.
When analyzing images containing fires, the model achieves a correct prediction rate of 42%. However, it's important to note that a substantial 58% of fire instances remain undetected - falsely considered as background (false negatives), indicating a significant room for improvement in capturing actual fire occurrences.
Similarly, in the case of images featuring smoke, the model demonstrates a correct identification rate of 29%. Nevertheless, a substantial 71% of smoke instances go unnoticed (false negatives), highlighting the considerable challenge in effectively detecting smoke within images, and the need for immediate notification about the instance.
For images that do not contain either smoke or fire, there is a noteworthy occurrence of false positives. Specifically, 78% of the model's predictions for fire in such scenarios turn out to be incorrect (false positives). Additionally, for smoke, 22% of the predictions in smoke-absent images are false positives. This suggests that the model may be prone to overestimating the presence of smoke and fire in situations where they are not actually present, warranting further investigation into the causes of these false positive alerts.

Given the scope of this project, the significance of minimizing false negatives cannot be overstated. False negatives represent instances where the model fails to recognize the presence of actual fires or smoke with potential consequences. Therefore, our primary focus should be on enhancing the model's sensitivity to ensure that it captures as many true instances of fires and smoke as possible. While reducing false positives is important to maintain efficiency and minimize unnecessary alerts, the paramount objective remains the swift and accurate identification of these hazardous situations, underscoring the critical nature of mitigating false negatives in our project.
Moreover, it is worth noticing that those percentage statistics regarding false positive and negative occurrences in Table 3, are partly biased due to the inherent complexity and ambiguous shape and form of smoke and fire. Smoke and fire can manifest in various ways, often exhibiting irregular shapes, dispersion patterns, and varying levels of intensity. This inherent variability can pose challenges for the model's bounding box predictions, as it may sometimes encompass a larger or smaller area than the actual fire or smoke, leading to instances of missing or inaccurately capturing the true fire regions within the images. This unpredictability in the appearance of smoke and fire in diverse real-world scenarios contributes significantly to the observed percentages of false positives and negatives, emphasizing the need for robust detection methods that can adapt to these arbitrary shapes and forms.

This observation suggests a need for further optimization, potentially focusing on the geometry and scale of bounding box predictions, to improve the model's sensitivity in real-world scenarios. Given the critical nature of detecting fire and smoke, improving the sensitivity for these classes should be a priority. High false-negative rates could have dangerous implications in our application, although adjusting the confidence intervals of the model should be a good strategy to partially overcome such issues.

Furthermore, once again it's worth considering the potential benefits of combining the results from both YOLOv7 and YOLOv8 models as previously stated in this report. Cross-validating the results from these two models can offer several advantages:

1. **Reducing False Alerts**: By cross-referencing detections from both models, we can potentially reduce false alerts. If both models independently detect a fire or smoke event, it adds confidence to the alert, making it less likely to be a false positive.

2. **Avoiding False Negatives**: The combination of the two models can also help avoid false negatives. If one model misses a fire or smoke event due to its limitations, the other model might still detect it. This redundancy can be critical in ensuring that no genuine fire or smoke incidents are missed. However, it's essential to acknowledge once again that this approach may come at the cost of processing speed. Running two models in parallel or sequentially can increase the computational load, especially on UAVs which have limited resources. Therefore, optimizing the inference pipeline to balance performance and accuracy will be crucial.

   An alternative approach for minimizing false positives could involve directing the Unmanned Aerial Vehicle (UAV) **-** equipped with the model **-** to issue an alert only when it consistently detects a target over a specified duration. For instance, it could be configured to send a notification alert only if it identifies a fire or smoke instance persistently for a continuous period (e.g. 5 to 10 seconds).

   This method would undeniably help in reducing false positive predictions and unnecessary alerts by introducing a temporal criterion for triggering them as follows:

- **Continuous Confirmation**: By requiring that the model detects the target continuously for a set timeframe, it reduces the likelihood of momentary false positive detections causing alerts. This means that a fire or smoke detection must be sustained over a significant duration to trigger an alert.
- **Filtering Out Temporary Anomalies:** Many false positives can occur due to temporary environmental conditions or brief anomalies that may momentarily mimic the presence of a fire or smoke. By mandating a continuous detection period, the system filters out such short-lived events, reducing unnecessary alerts.
- **Increased Confidence:** Continuous detection over a defined time frame increases the confidence that a real fire or smoke event is occurring. This helps in distinguishing between genuine emergencies and potential false alarms, improving the overall reliability of the alerting system.
- **Reduced Alert Fatigue:** Without this temporal criterion, frequent but brief false positives could lead to alert fatigue, causing users to ignore or dismiss alerts. By requiring sustained detection, the system ensures that alerts are only triggered for more likely genuine events, reducing the chance of fatigue.
- **Improved Resource Efficiency**: By reducing unnecessary alerts, this approach optimizes resource utilization, such as storage, and human attention. Unnecessary alerts can strain resources and personnel, which can be costly and inefficient.

In this specific application, which requires immediate alerting, a high false-positive rate for images without fire or smoke is nevertheless, less concerning. In this context, it's more acceptable to have extra false alarms rather than missing actual fire or smoke events. The trade-off between **sensitivity** and **specificity** should be carefully tuned to align with the urgency and consequences of the application.

Discussion & Future Work

In this report, we describe a solution that enables firefighting teams and authorities to efficiently monitor a larger portion of forested areas and collect real-time data when dealing with crises. An early warning system proves invaluable, particularly in firefighting contexts, given the fire's unpredictable nature. Decisions must be made within hours, and without sufficient, reliable data, situations can rapidly spiral out of control. This can often be achieved through the use of Unmanned Aerial Vehicles (UAVs), allowing operators to maintain a safe distance and execute potentially perilous maneuvers that would otherwise endanger a pilot's life.

Several enhancements could further solidify this solution. The addition of thermal cameras, capable of capturing information in the infrared spectrum, would enhance the accuracy of the model and decrease the overall response time by being able to detect sources of fire where the smoke detection would be challenging especially in the early stages when minimal damage has occurred and containment remains feasible. Additionally, equipping each UAV with firefighting capabilities would enable deployment to inaccessible areas. In some cases, this could empower firefighters to contain disasters much earlier, thereby preserving ecosystems, lives, properties, and reducing operational costs.
Furthermore, by implementing specific network capabilities in each UAV and enabling communication with multiple UAVs through peer-to-peer wireless mesh networks, we can establish a fully autonomous or semi-autonomous network of edge devices. These devices can be deployed for monitoring and operation without human intervention. Depending on the situation and predefined operational policies, they can engage even before human operators detect early warnings, thus substantially reducing response times.

Members-Roles

Our project group is comprised of four dedicated members with equal participation in each stage: Vasiliki Terizaki, Christos Pantoleon, Eleftherios Fthenos, Christiana Kolai.

Time Plan

The project timeline is shown in Figure. Our team used a combination of web calls and in-person meetings for communication and alignment on tasks and deadlines. We employed Trello for task management, enabling us to track progress in real-time and adjust our efforts as needed.

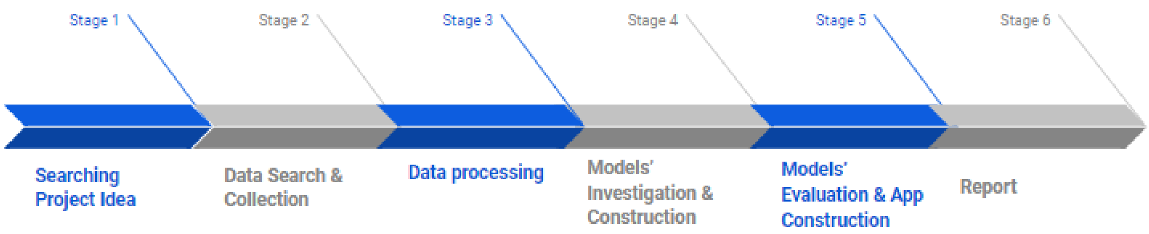


Figure 12. Project Timeline.

Bibliography

- M. Yandouzi et al., "Forest Fires Detection using Deep Transfer Learning," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 8, pp. 268–275, Oct. 2022, doi: 10.14569/IJACSA.2022.0130832.
- https://publications.jrc.ec.europa.eu/repository/handle/JRC133215
- https://effis.jrc.ec.europa.eu/apps/effis.statistics/seasonaltrend
- https://effis-gwis-cms.s3.eu-west-1.amazonaws.com/effis/reports-and-publications/forest-focus-studies/FireDamagesJRC_Finale.pdf
- https://www.unep.org/resources/report/spreading-wildfire-rising-threat-extraordinary-landscape-fires?gclid=CjwKCAjw29ymBhAKEiwAHJbJ8u3l9jo7IaswktTcoTGL9idedzmTChb42S-i87-yfmnLW5d6R5GwJxoCXf4QAvD_BwE
- https://www.copernicus.eu/en/news/news/observer-global-wildfire-watch-copernicus-ems-and-cams-monitor-wildfires-2023
- https://www.who.int/health-topics/wildfires?gclid=CjwKCAjw29ymBhAKEiwAHJbJ8gba-37yEn4rjKfRCi_J1faQdLaLM9pqdBJ_AKMbOkut9dPFVQzvlRoClUMQAvD_BwE#tab=tab_1
- https://civil-protection-knowledge-network.europa.eu/news/success-based-preparation-ercc-ready-2023-wildfire-season
- https://environment.ec.europa.eu/topics/forests/forest-fires_en
- https://ec.europa.eu/research-and-innovation/en/horizon-magazine/swarming-drones-could-help-fight-europes-megafires
- https://effis.jrc.ec.europa.eu/about-effis/technical-background/rapid-damage-assessment
- https://www.theguardian.com/world/2023/sep/01/greek-wildfires-a-visual-guide
- https://effis.jrc.ec.europa.eu/
- http://beyond-eocenter.eu/index.php/web-services/firehub
- https://en.wikipedia.org/wiki/Drones_in_wildfire_management
- https://par.nsf.gov/servlets/purl/10026207
- https://www.aa.com.tr/en/europe/greece-experiences-decline-in-number-of-fires-but-monumental-increase-in-burned-areas/2980616#
- https://www.statista.com/statistics/1264709/area-burned-by-wildfire-in-greece/
- https://www.keeptalkinggreece.com/2023/09/01/greece-2023-20-fewer-big-forest-fires-270-increase-in-burnt-ares/
- https://en.wikipedia.org/wiki/2023_Greece_wildfires
- https://www.businessdaily.gr/english-edition/93567_natl-observatory-11-fires-2017-2023-burned-23-attica-region
- https://www.dimokratiki.gr/28-08-2023/pyrkagies-12-dis-evro-to-synoliko-kostos-gia-tin-ellada-i-katastrofi-kai-oi-apozimioseis/
- https://www.naftemporiki.gr/english/1507655/fires-1-2-billion-the-total-cost-for-greece/
- https://www.datacamp.com/blog/yolo-object-detection-explained
- https://www.google.com/url?q=https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20230913-2&sa=D&source=docs&ust=1694856596499517&usg=AOvVaw2qRgfE0k0UECzF3aK84o_c

Appendix

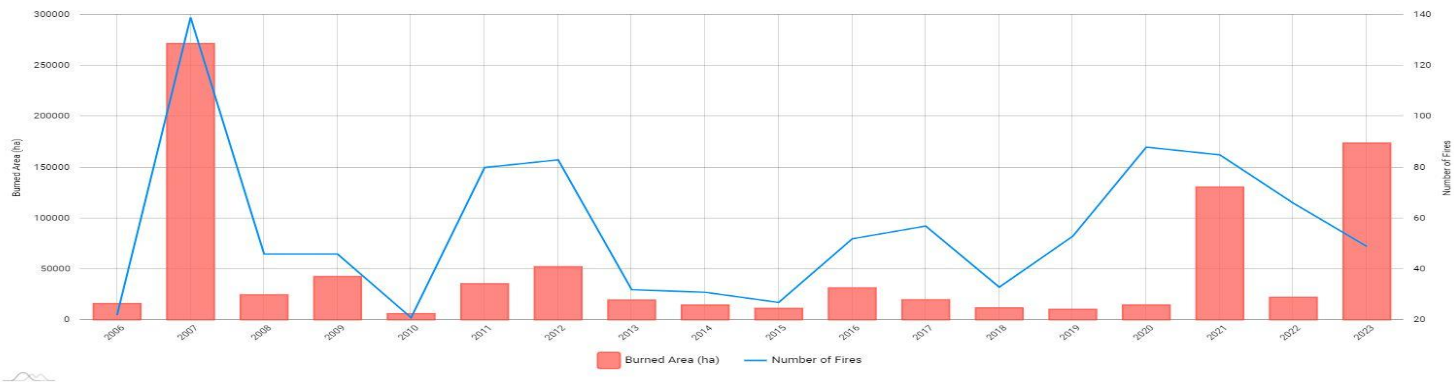
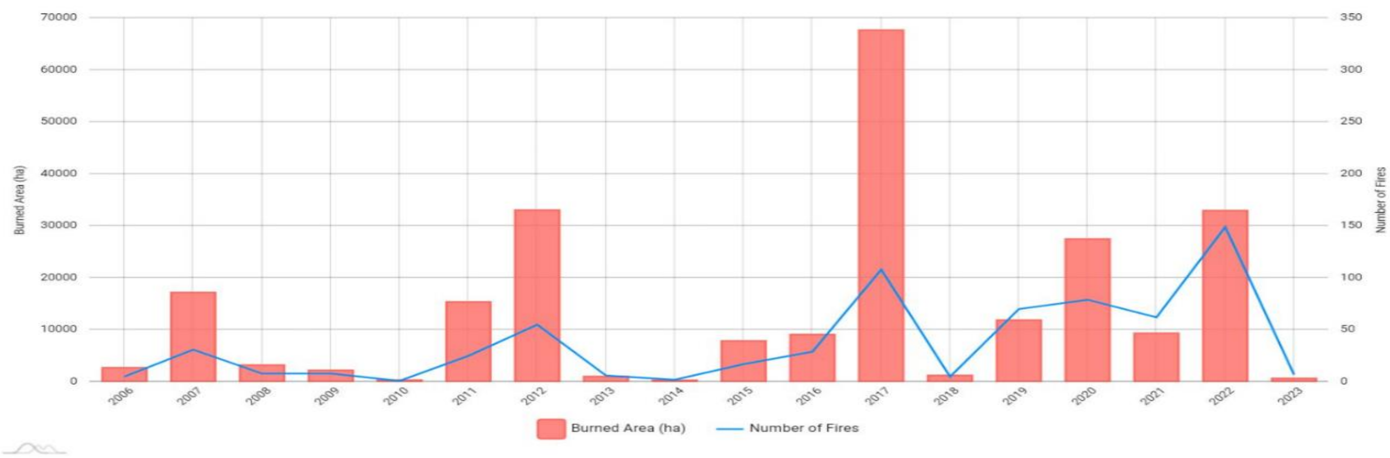


Figure 1. Burned Area and number of fires in Greece from 2006 to 2023.



Data source: EFFIS

Figure 2. EU Burned Area: 2006-2023.

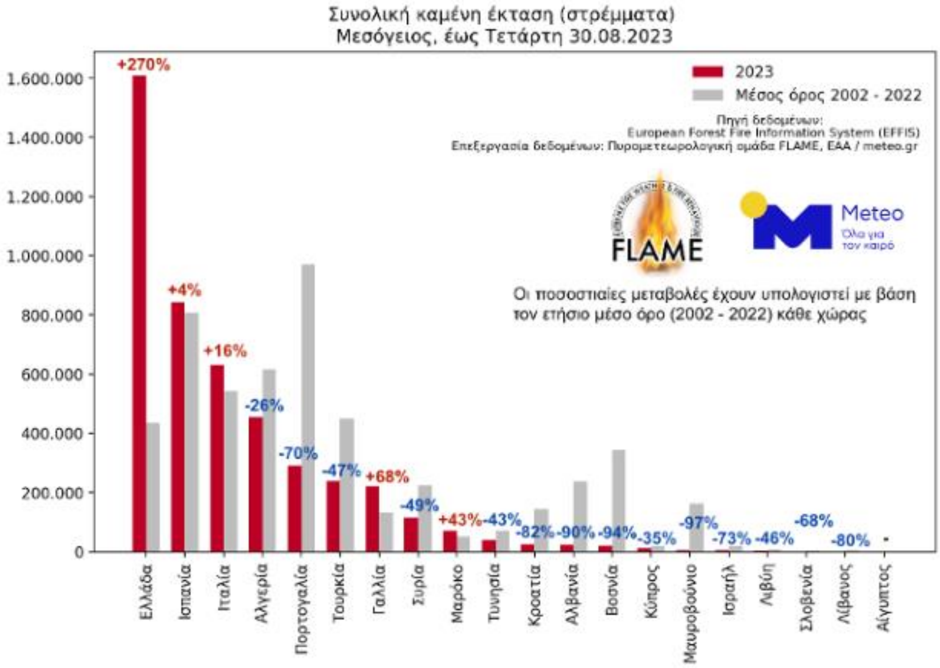


Figure 3. Total burnt area in Mediterranean countries by Wednesday 30 August 2023.

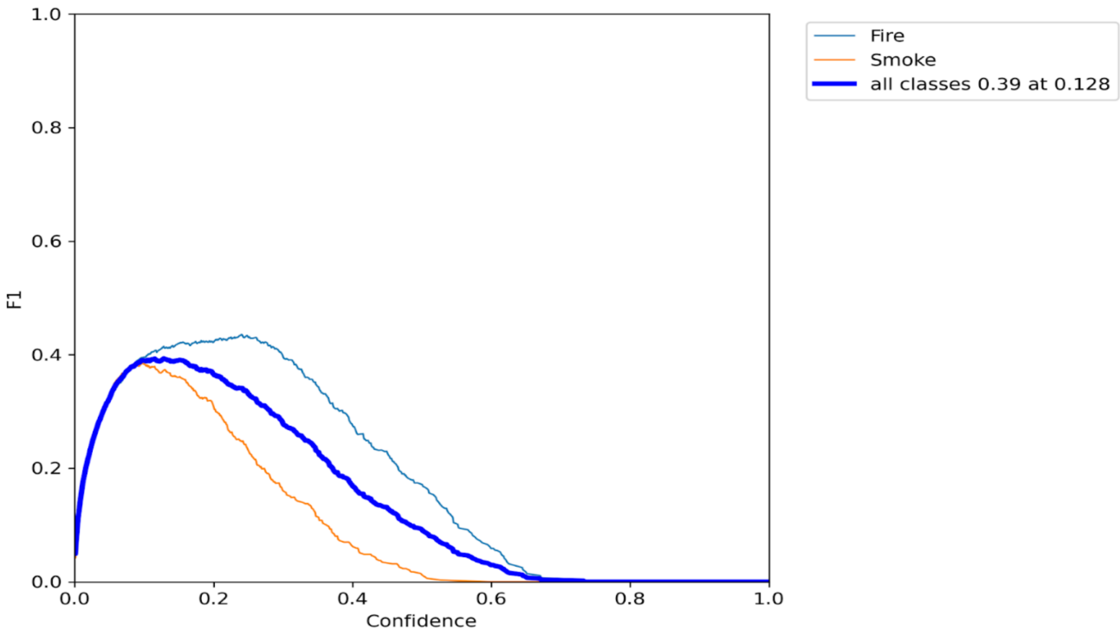


Figure 4. F1 curve for YOLO-v7.

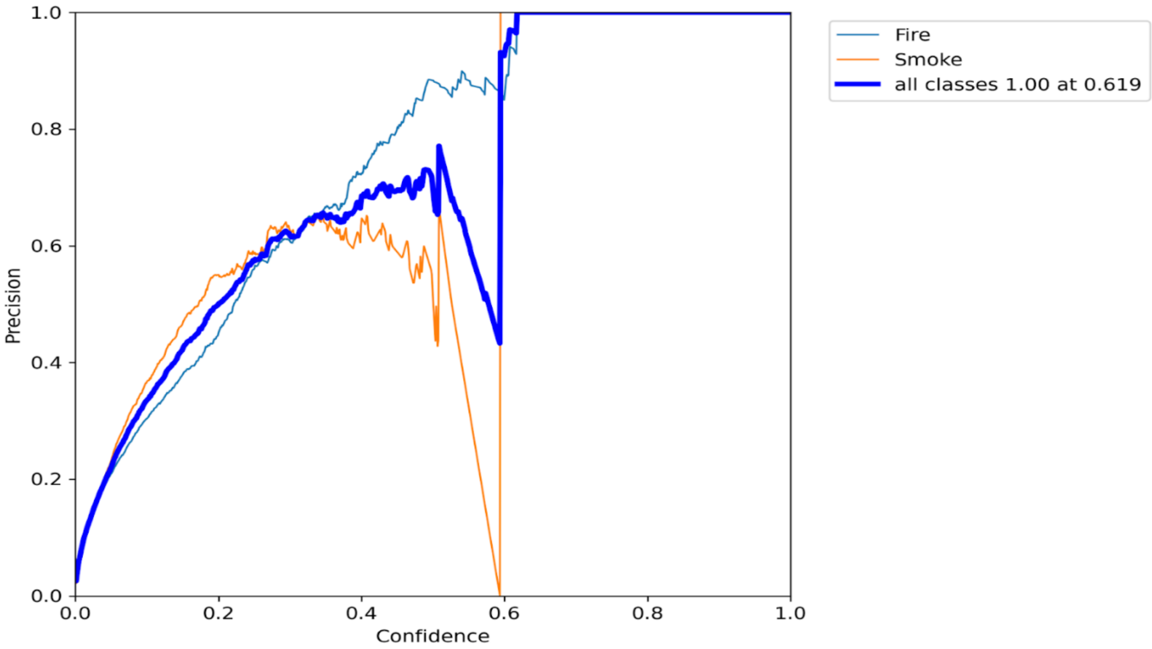


Figure 5. Precision curves for each class of YOLO-v7.

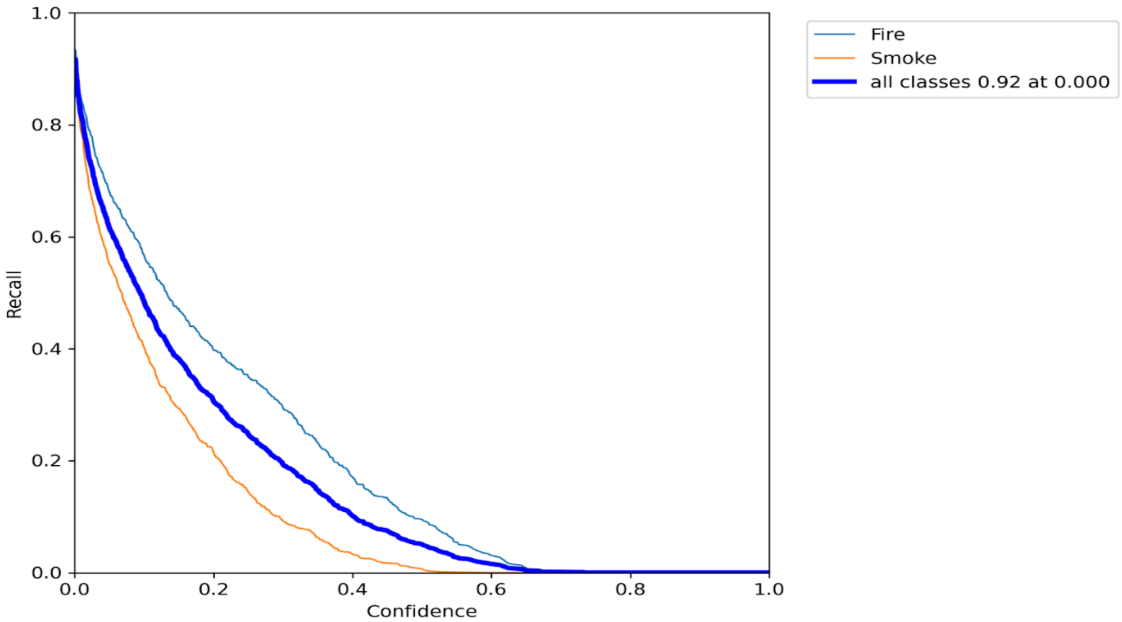


Figure 6. Recall curves for each class of YOLO-v7.

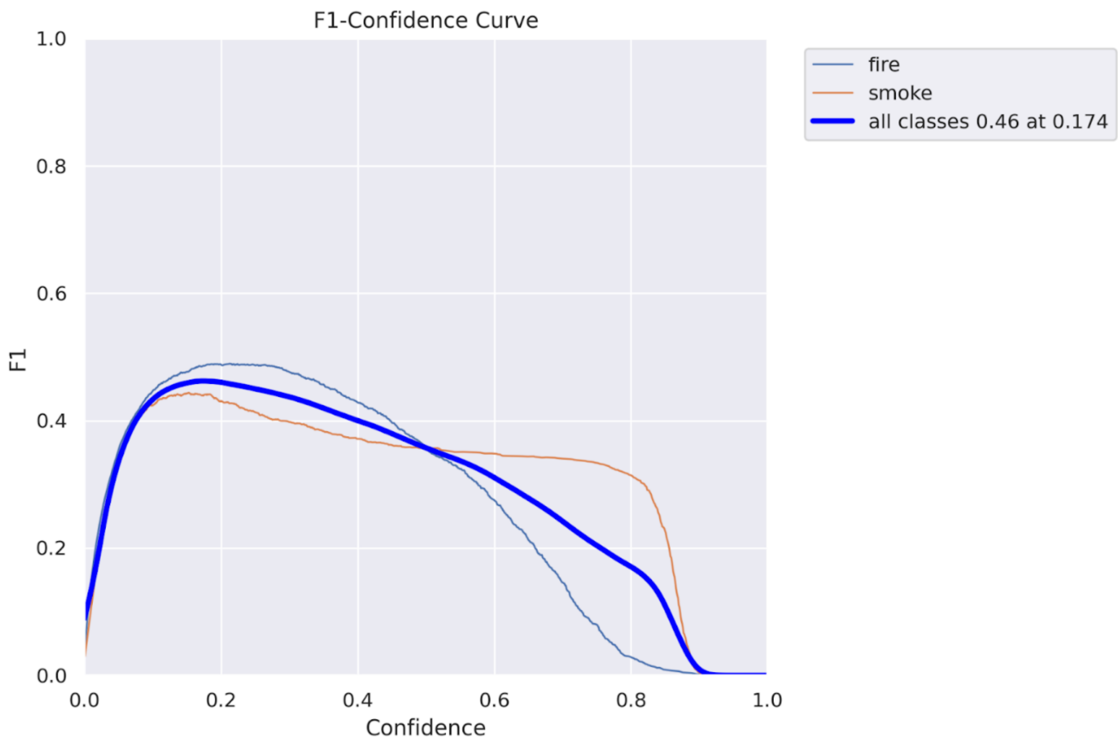


Figure 7. F1 curve for YOLO-v8.

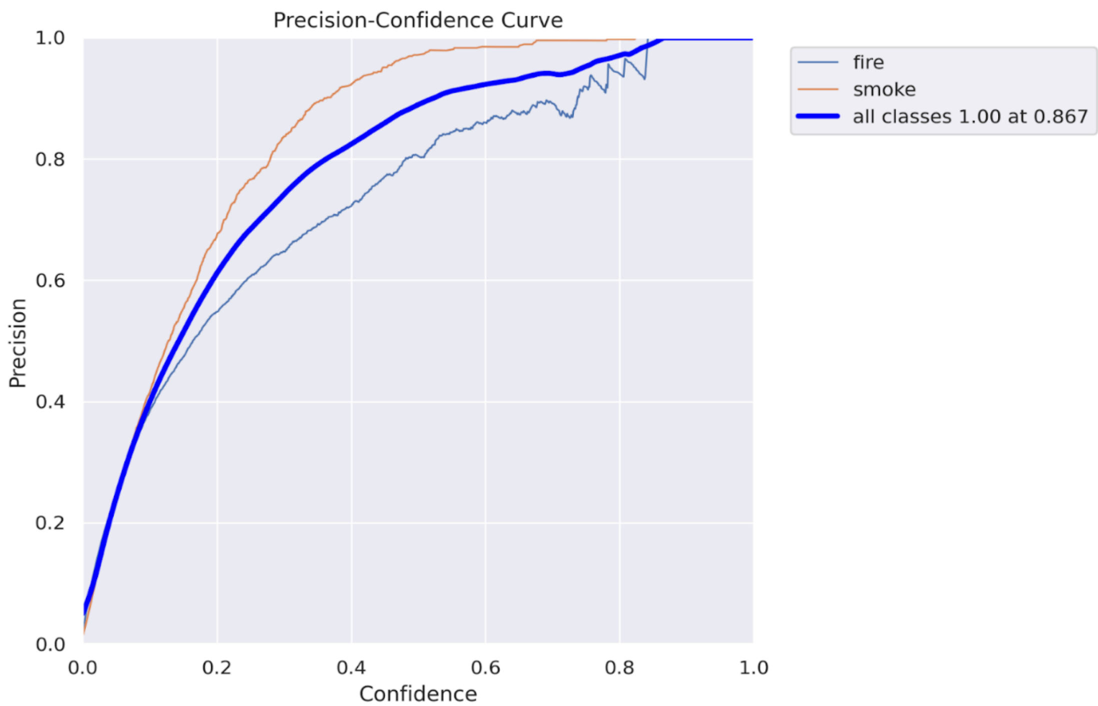


Figure 8. Precision curves of the two classes of YOLO-v8.

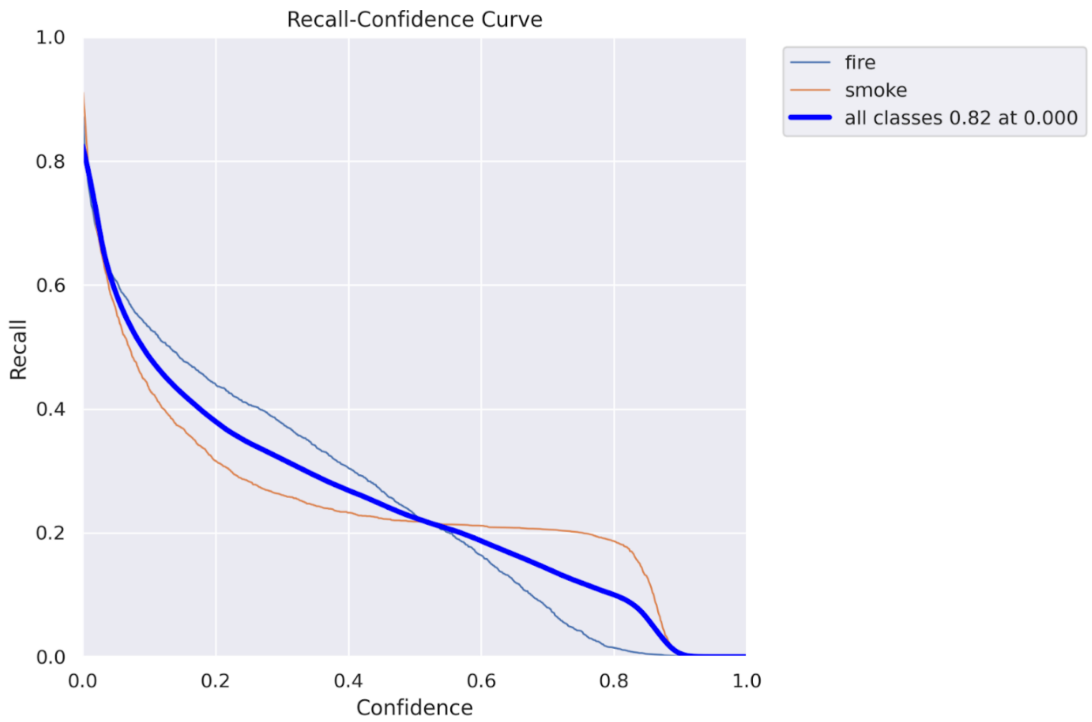


Figure 9. Recall curves of the two classes of YOLO-v8.

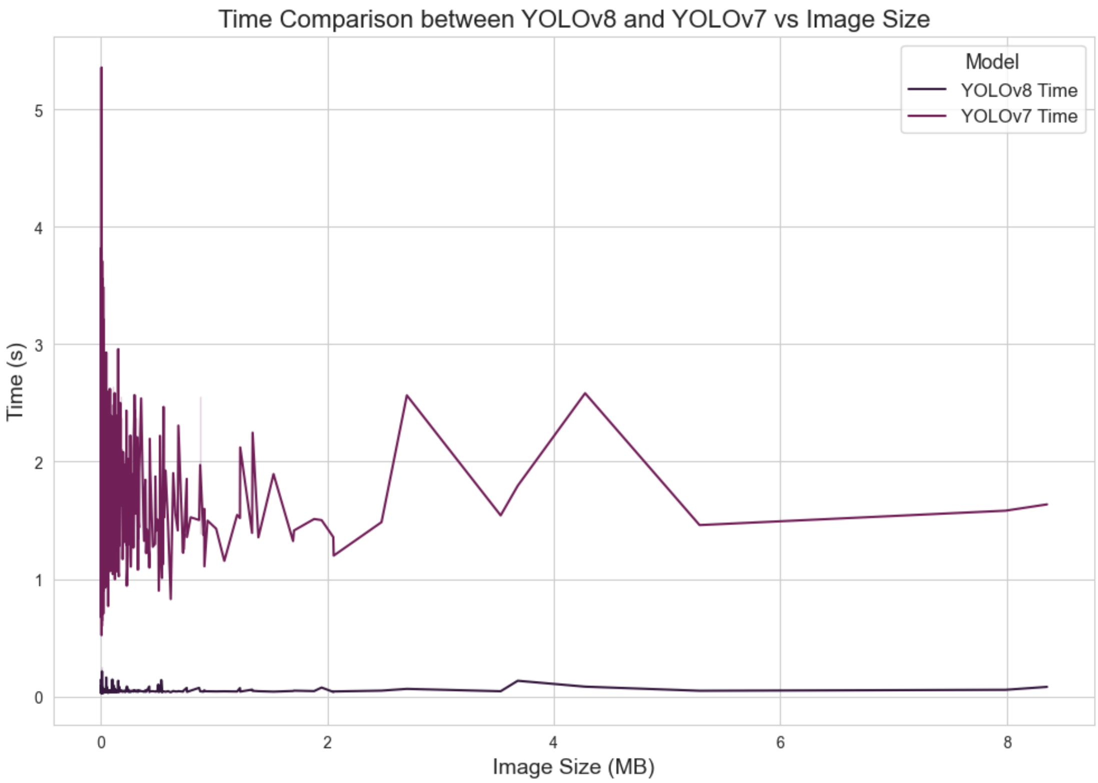


Figure 10. Processing time of images of YOLO-v7 and YOLO-v8 models.

Yolov8 Medium Model Characteristics.

▼ Detection

Model	size (pixels)	mAP^{val} 50-95	Speed CPU (ms)	Speed T4 GPU (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	-	-	3.2	8.7
YOLOv8s	640	44.9	-	-	11.2	28.6
YOLOv8m	640	50.2	-	-	25.9	78.9
YOLOv8l	640	52.9	-	-	43.7	165.2
YOLOv8x	640	53.9	-	-	68.2	257.8

- **mAP^{val}** values are for single-model single-scale on COCO val2017 dataset. Reproduce by `yolo mode=val task=detect data=coco.yaml device=0`
- **Speed** averaged over COCO val images using an Amazon EC2 P4d instance. Reproduce by `yolo mode=val task=detect data=coco128.yaml batch=1 device=0/cpu`