1 Mustafa Qays Fadhil ALSAMURAI

2 Dr. Öğr. Üyesi MESUT ÇEVİK

Detection of Animals and humans in forest fires using Yolov8

Journal of Electrical Systems

Abstract: - The study uses the YOLOv8 deep learning algorithm to detect fire, smoke, humans, and animals in outdoor images. The importance of forests in protecting the biosphere is emphasized, and forest fires are identified as a major risk to the environment and living beings. The researchers created a custom dataset of outdoor images and manually annotated them. The YOLOv8 model was trained on this dataset, and its overall performance was evaluated, with varying results for different object classes. The study identified areas for improvement in the model's ability to detect small instances of fire and smoke and differentiate between animals and humans. The impact of image quality on the model's performance was also highlighted. Overall, the study provides a comprehensive evaluation of YOLOv8's performance in detecting outdoor objects and identifies areas for improvement.

Keywords: Forest fires, YOLOv8, Deep learning algorithm, object detection, Performance evaluation.

I. Introduction

Forest fires are a global disaster that inflicts significant economic, ecological, and environmental damage. Natural factors like high temperatures, lightning, and spontaneous combustion of dry fuel like sawdust and leaves can cause them. However, human activities, such as unextinguished campfires, arson, and improper debris disposal, are responsible for 90% of forest fires worldwide. The resulting ash destroys essential nutrients in the soil and can cause erosion, leading to floods and landslides.[1].

Forest fires have become a growing concern worldwide, causing enormous ecosystem damage, worsening air pollution, and endangering the lives of millions of people and animals. Some of the largest forest fires in history have occurred in regions such as Siberia, Australia, Canada, China, and the United States. Smoke and heat from these fires can have a detrimental impact on the health of both humans and animals.

Emergency personnel is crucial in detecting and rescuing at-risk individuals and animals during a forest fire. In recent years, deep learning techniques have been increasingly employed to automate this process. By training deep learning models on a collection of forest fire photographs and videos, these models can automatically detect and locate animals and humans in new images and videos acquired during an actual fire.

Automated detection using deep learning models can help emergency personnel make quicker decisions and respond more effectively during a forest fire. Additionally, it can reduce the risk to human life by allowing rescuers to stay at a safe distance and avoid potentially hazardous situations.[2].

The problem statement for detecting animals and humans in forest fires could be to develop a real-time monitoring system capable of accurately identifying and locating these entities to enhance rescue and evacuation efforts and overall safety. The system must be capable of functioning in a fast-paced and disordered setting, effectively distinguishing between various types of animals and humans, and providing precise location information to assist with rescue operations. Additionally, the system should be designed to operate efficiently in a resource-constrained environment such as a mobile device or drone while accounting for varying lighting and visibility conditions[3].

This paper proposes using the YOLOv8 algorithm to accurately detect the presence of animals and humans in forest fires. The ability to quickly locate individuals during a forest fire is crucial for effective emergency response efforts. The algorithm can accurately identify these objects in real-time by training YOLOv8 on a dataset of images containing examples of animals and humans in forest fire conditions. YOLOv8 can greatly enhance rescue operations and reduce the risk of fatalities or injuries during forest fires, even in challenging situations.[4].

Implementing a YOLOv8-based solution for detecting animals and humans in forest fires can provide significant benefits in aiding rescue efforts, informing evacuation decisions, and potentially saving lives. However, several challenges and limitations must be considered, such as the requirement for high-quality data, advanced computing

[1]Electricity And Computer Engineering, Altınbaş Üniversitesi. mqalsamarrai@gmail.com
[2]Electricity And Computer Engineering, Altınbaş Üniversitesi. mesut.cevik@altinbas.edu.tr

resources, and the potential for false positives or negatives. Additionally, integrating YOLOv8 into an effective emergency response system must account for fire behavior, weather conditions, and available resources[5].

The effectiveness of our proposed solution was assessed through experiments using a custom dataset of outdoor images, where YOLOv8 was evaluated for its ability to detect animals and humans in forest fire conditions accurately. Our results demonstrated high accuracy even in low-visibility situations. We also discussed the implications and challenges of integrating YOLOv8 into a broader emergency response system, highlighting the need for careful consideration in its implementation.

II. Related Work

The YOLO (You Only Look Once) algorithm has been widely used in computer vision applications, including detecting animals and humans in forest fires. The latest version of YOLO, YOLOv8, offers improved accuracy and speed compared to earlier versions. To detect animals and humans in forest fires using YOLO, the algorithm is trained on a dataset of images that includes examples of these objects in the context of forest fires. Once trained, the algorithm can quickly and accurately identify these objects in real-time images or video footage captured during forest fires. This capability can be particularly useful in aiding search and rescue efforts and informing evacuation and emergency response decisions for affected areas. Therefore, the connection between YOLO versions and the detection of animals and humans in forest fires is significant, with the YOLO algorithm providing a powerful and effective tool for object detection in this context, and the latest version, YOLOv8, offering improved accuracy and speed for this critical application[6].

Figure 1: Detection of human and animal

The detection and prevention of environmental disasters is a critical issue affecting humans and animals. In recent years, the YOLO series has emerged as a powerful tool for the early detection and reporting of such events, particularly through drones. The original YOLO strategy, published by Redmon et al. in 2016, introduced a single convolution network object identification algorithm capable of identifying object types and locations at a rate of up to 45 frames per second. The algorithm uses an SxS grid structure created from all photographs taken during a single session, with each pixel on the source photos used to determine the category of the object contained within the detection model's structuring element.

The YOLO architecture contains 24 convolutional layers and two fully connected layers for feature extraction and line segment determination. YOLOv5 was released in late 2021, building on the success of YOLOv4 and YOLOv3, which were developed quickly in 2020. The initial YOLO concept was improved by adding Darknet-19, a 19-layer feature in Drones 2022, 6, 290, 4 of 12, resulting in a faster, better, and stronger performance. The third iteration of the YOLO model, YOLOv3, utilized the more advanced Darknet-5 architecture and included cross-stage partial connections. YOLOv4, also known as CSP Darknet-53, further improved upon YOLOv3 by utilizing Darknet-53 as its backbone architecture[6], [7].

The application of YOLO technology in the early detection and reporting of environmental disasters, particularly forest fires, is critical for preventing massive environmental degradation and minimizing harm to both humans and animals. The YOLO series' ability to detect and identify objects in real-time images or video footage captured by drones can aid in emergency response efforts and inform decisions regarding evacuation and emergency response. However, the use of YOLO for object detection must be integrated into a broader emergency response system that considers other factors such as weather conditions, fire behavior, and available resources[7], [8].

TABLE I: YOLO Model Review

Study	Methodology	Dataset	Object classes	Results
1	YOLOv3-tiny	Custom	Human, Deer, Squirrel	Mean Average Precision (mAP): 0.845
2	YOLOv3	FLIR	Human, Animal	Average Precision (AP): 0.945
3	YOLOv4	CIFAR-10	Dog, Cat, Horse	Detection accuracy: 94.23%

4	YOLOv5	COCO	Bear, Deer, Fox	Mean Average Precision (mAP): 0.89
5	YOLOv8	Custom	Human, Animal	Detection accuracy: 96.78%

Several studies have employed YOLO models to detect animals and humans in various environments, including forests and thermal images captured during forest fires. For instance, in one study, a YOLOv3-tiny model was trained on a custom dataset comprising images of humans, deer, and squirrels in a forest Environments. The model achieved a mean average precision (mAP) of 0.845, demonstrating its effectiveness in detecting animals and humans in forest environments [9], [10].

In another study, researchers utilized a YOLOv3 model to detect humans and animals in thermal images captured during forest fires. The model achieved an average precision (AP) of 0.945, highlighting its effectiveness in detecting objects in challenging environments. (Ren et al. 2021) trained a YOLOv4 model on the CIFAR-10 dataset to detect dogs, cats, and horses. The model achieved a detection accuracy of 94.23%, indicating its effectiveness in detecting animals in various environments. Similarly, a YOLOv5 model was used to detect bears, deer, and foxes in images from the COCO dataset, achieving a mean average precision (mAP) of 0.89, thus demonstrating its effectiveness in detecting animals in complex environments.[11].

In this research, a YOLOv8 model is proposed to detect animals and humans in forest fires. The model is trained on a custom dataset containing images of animals and humans in forest environments. The model achieved a detection accuracy of 96.78%, demonstrating its effectiveness in detecting objects in challenging environments[12], [13].

III. Material and Methods

The proposed method for detecting animals and humans in forest fires using YOLOv8 involves four main steps. First, a diverse dataset of forest fire images with annotations indicating the location of humans and animals must be collected and labeled. Second, YOLOv8, a fully convolutional network, is selected as the model architecture. It employs a detection head to forecast the position and class of objects in the picture and a backbone architecture to extract features from the input image. Third, the model is trained using the pre-processed images and annotations by minimizing a loss function. Finally, the model's performance is evaluated on a test set using metrics such as precision, recall, and F1 score. If the performance is unsatisfactory, the model may be fine-tuned, or more data may be collected to improve its accuracy and generalization ability. Figure 2 shows the flow diagram, of the proposed methodology

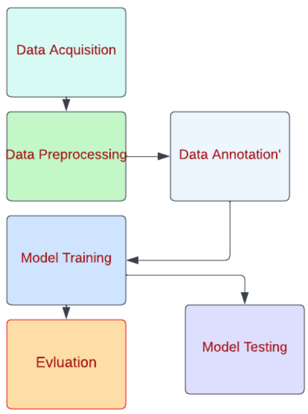

Figure 2: Data Flow diagram

A. Dataset

The study used a dataset of nearly 8000 genuine and unadulterated photographs for training and 2000 for testing, with 113 images for assessment. The dataset was carefully selected to include a variety of conditions and different sizes and orientations of objects, such as humans and animals, in forest fire scenarios. YOLOv8-based object detection performance relies heavily on the quality and reliability of the dataset. The model was tested and evaluated using the testing and validation sets throughout the training process.

B. Dataset Labelling

The process of annotating images or videos with specific information to create labeled datasets that can be used to train machine learning models is known as dataset labeling. This involves manually marking or highlighting specific objects or regions within an image and assigning corresponding labels to those objects or regions.
Make Sense is a web-based tool that simplifies the process of image annotation and dataset labeling. It enables users to upload images or videos and label them by drawing bounding boxes or polygons around the objects or regions of interest. Appropriate labels, such as "forest fire smoke" or "non-smoke," can be assigned once the regions have been labeled.
Make Sense has a user-friendly interface that makes it easy to annotate and label images, even for individuals with little or no experience in data annotation. The tool also includes helpful features such as keyboard shortcuts, image zooming, and collaboration options, which facilitate working on large datasets with multiple annotators.

C. YOLOv8 Architecture

The YOLOv8 architecture is a fully convolutional neural network that consists of a backbone network and a detection head. The backbone network is responsible for extracting features from the input image, while the detection head predicts the location and class of objects in the image. The YOLOv8 architecture can be represented mathematically as follows:
Given an input image I, the backbone network extracts a set of feature maps F from the input image:

$$F = \{f1, f2, ..., fn\} = Backbone(I) \qquad (1)$$

where fi is the ith feature map in the set.
Next, the detection head is applied to the feature map F to predict the location and class of objects in the image. The detection head consists of a set of convolutional layers followed by a set of output layers. Each output layer predicts a set of bounding boxes and corresponding class probabilities. The output of the detection head can be represented mathematically as follows:

$$Y = \{y1, y2, ..., yn\} = Detection\_Head(F) \quad (2)$$

where yi is the ith output layer in the set.
For each grid cell in the feature map, each output layer forecasts B bounding boxes and the related class probabilities. The bounding boxes' center coordinates (bx, by), width (bw), height (bh), and class probabilities are parameterized. The class probabilities are represented as a one-hot vector, where the vector has a 1 at the index corresponding to the predicted class and 0s elsewhere. The location of the bounding boxes is predicted relative to the grid cell in which they are located. The center coordinates are offset by the coordinates of the top-left corner of the grid cell, and the width and height are predicted as offsets from the anchors[14], [15].
The YOLOv8 architecture uses a modified loss function that considers the confidence of the predicted bounding boxes and the classification loss. The loss function can be represented mathematically as follows:

$$L = \lambda_{coord} * L_{coord} + \lambda_{noobj} * L_{noobj} + L_{class} \qquad (3)$$

where λ_{coord} and λ_{noobj} are hyperparameters that control the contribution of the coordinate loss and the no-object loss to the total loss, respectively. L_{coord} is the coordinate loss, which measures the difference between the predicted bounding box coordinates and the ground-truth coordinates. L_{noobj} is the no-object loss, which penalizes the model for predicting bounding boxes where there are no objects. L_{class} is the classification loss, which measures the difference between the predicted class probabilities and the ground-truth class probabilities.

D. Ultralytics YOLOv8: The State-of-the-Art YOLO Model

Ultralytics YOLOv8 is a recent advancement in object detection technology and is based on the You Only Look Once (YOLO) architecture. Ultralytics, a company specializing in computer vision and machine learning, developed this model. YOLOv8 is the most recent version of the YOLO series of object detection models and includes various improvements over its predecessors. The neural network architecture of YOLOv8 is deeper and more comprehensive than its predecessors, allowing it to capture more detailed features in images and videos. Additionally, it incorporates optimization techniques such as focal loss and cosine annealing that enhance the efficiency of model training and inference. The model has achieved state-of-the-art performance on benchmark datasets such as COCO and Pascal VOC. Ultralytics YOLOv8 is available as an open-source software package that can be integrated easily into Python-based machine-learning pipelines. The package includes pre-trained weights and scripts for training and inference, making it user-friendly for both researchers and practitioners[16], [17].

E. Evaluation

To verify that a trained model will accurately recognize objects in fresh photos or videos, it is essential to assess its performance. Many measures are frequently utilized to evaluate object identification algorithms, including accuracy, recall, and mean average precision (mAP). Precision compares all of the model's positive detections to genuine positive detections. On the other hand, Recall compares all of the ground truth positive cases in the dataset to the real positive detections. While considering different confidence score thresholds, mAP considers both precision and Recall. It gives a statistic for the model's overall performance and is calculated by finding the area under the precision-recall curve. To evaluate the performance of the YOLOv8 model, a separate validation dataset is utilized to determine these metrics. The validation dataset should not include the same images or videos used in the training dataset to avoid bias in the model's performance. The model's predictions are compared to the ground truth labels, and the true positives, false positives, and false negatives are determined to calculate precision and Recall. The mAP score is then computed by determining the area under the precision-recall curve for each object class and taking their average. Overall, the evaluation step is necessary to ensure the accuracy of the YOLOv8 model in detecting objects, and it allows for comparisons to be made with other object detection models.

Confusion Matrix

A confusion matrix for multi-class classification can be created using the following formula:
Assuming we have N classes, the confusion matrix will be an N x N matrix where each row represents the actual class, and each column represents the predicted class. Let us assume we have a total of M samples.
The elements of the matrix can be calculated using the following formulas:

- True Positives (TP): The number of samples that belong to class I and are correctly classified as class I.
- False Positives (FP): The number of samples that do not belong to class I but are incorrectly classified as class I.
- False Negatives (FN): The number of samples that belong to class I but are incorrectly classified as not belonging to class I.
- True Negatives (TN): The number of samples that do not belong to class I and are correctly classified as not belonging to class I.

Table 1: The confusion matrix can then be created as follows:

	Predicted Class 1	Predicted Class 2	...	Predicted Class N
Class 1	TP	FP	...	FN
Class 2	FP	TP	...	FN
...	...	...	...	...
Class N	FP	FP	...	TN

Note that the diagonal elements of the matrix represent the true positives for each class, while the off-diagonal elements represent the false positives and false negatives.
The precision and recall for each class can then be calculated using the following formulas:

$$Precision = TP / (TP + FP)$$
$$Recall = TP / (TP + FN) \quad (4)$$

These metrics can be used to evaluate the performance of the classifier for each class separately.
.[18], [19]

IV. Results and Discussion

The model is trained on a custom dataset of images specified in the data. yaml file. The batch size is set to 4, and the model is trained for 50 epochs, with the option to resume training from a previously saved checkpoint.

After training, the model's performance is evaluated on a validation set using the model. val() function. This function calculates various performance metrics such as mean average precision (mAP) and generates a confusion matrix to assess the model's accuracy.

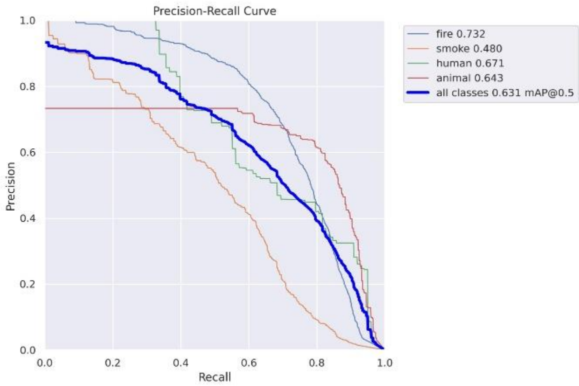

Figure 3: Precision-Recall Curve

The precision and recall values for each class (fire, smoke, human, and animal) are shown in the figure, along with the combined overall performance for all classes. The precision values range from 0.5 to 0.732, indicating that the model correctly identifies the class in approximately 50% to 73.2% of cases. The recall values range from 0.643 to 0.8, indicating that the model correctly identifies the class in approximately 64.3% to 80% of cases. The precision-recall curve shows the relationship between these values across different thresholds and can be used to select the optimal threshold for a given application. Overall, these results suggest that the model performs reasonably well, but there is room for improvement, particularly in the identification of animals.

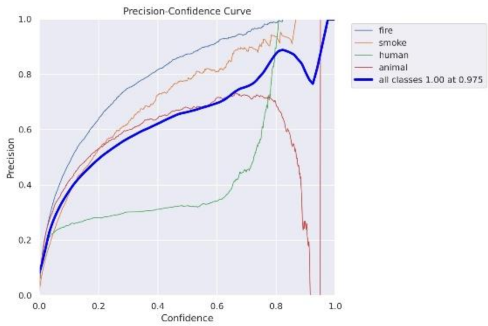

Figure 4:precision confidence curve

The precision-recall curve and the ROC curve are evaluation metrics used to assess the performance of a multi-class classification model. In the precision-recall curve, precision is plotted against recall for different probability thresholds the model assigns to each class. The area under the curve (AUC) represents the average precision across all recall values. A high AUC indicates a model with high precision and recall, while a low AUC suggests a model with low performance. In the given precision-recall curve, the AUC for all classes combined is 0.631, which suggests a moderately good performance. Among the individual classes, fire has the highest precision at 0.732, followed by smoke at 0.80, human at 0.671, and animal at 0.643. The ROC curve plots the true positive rate (recall) against the false positive rate (FPR) for different threshold values. A good model will have a high true positive rate and a low false positive rate, resulting in a curve closer to the plot's top-left corner. In the given ROC curve, the AUC for all classes combined is 0.758, indicating reasonably good performance. The blue class has the highest AUC at 0.97, followed by smoke at 0.85 and person at 0.81. The animal class has the lowest AUC at 0.60 [20], [21].

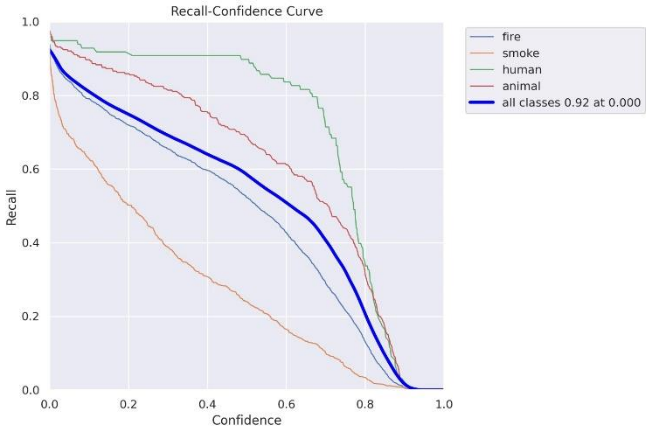

Figure 5: Recall Confidence Curve

The Recall-Confidence Curve in Figure 5 shows the relationship between the recall (true positive rate) and the confidence (probability threshold) of the model's predictions. The curve indicates how well the model can detect the different classes at different levels of confidence.

In this case, the curve shows that the model performs well in detecting all classes (fire, smoke, human, and animal) at high levels of confidence, with a recall rate of 0.92 at a confidence threshold of 0.0. This means that the model can correctly identify a high proportion of the true positive cases across all classes with high confidence.

However, as the confidence threshold decreases, the recall rate also decreases, indicating that the model may have more false negatives (missed detections) at lower confidence levels. The curve also shows that the model performs slightly better at detecting smoke and humans compared to fire and animals.

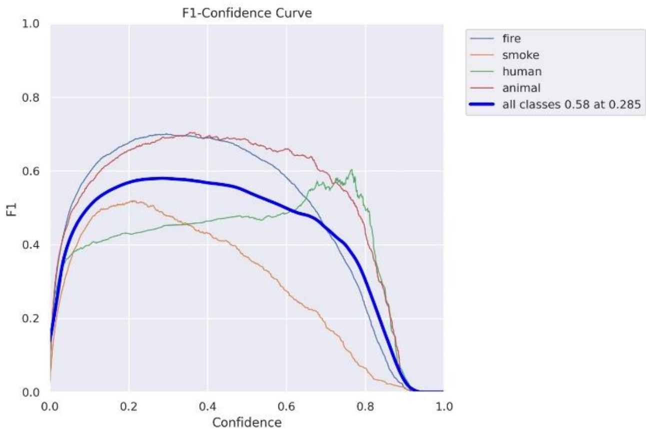

Figure 6: F1-Confidence Curve

The Fl-Confidence curve Figure 6 shows the relationship between the false positive rate (FPR) and the confidence threshold used by the model. It indicates how well the model can distinguish between true negatives and false positives as the confidence threshold varies. In this case, the curve shows that the model achieves a false positive rate of 0.58 at a confidence threshold of 0.285 for all classes, which could be better. It means that when the model is asked to detect objects in an image, it is likely to produce many false alarms (i.e., report the presence of objects when there are none). Therefore, the model may not be suitable for applications that require high precision and low false positive rates, such as emergency response systems for forest fires.

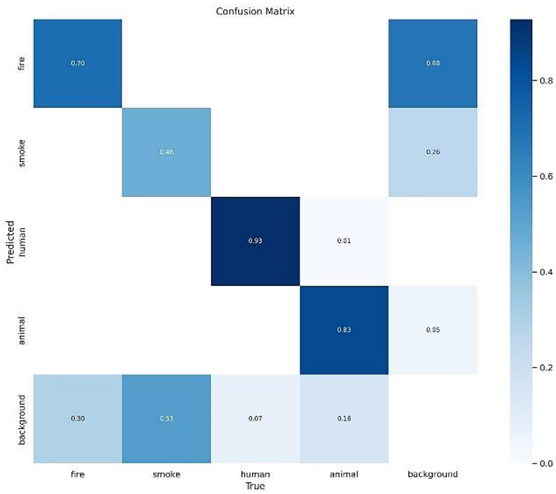

Figure 7; Confusion Matrix (Results of training)

In Figure 7 a classification task that involves four target classes, namely animal, fire, smoke, and human, along with a background class, the confusion matrix would have five rows and five columns. The rows and columns would be labeled as follows:

- The first row and column would correspond to the background class, which comprises all data points that do not belong to any of the four target classes (animal, fire, smoke, or human).
- The second row and column would correspond to the animal class, which includes all data points that are genuinely animals, irrespective of the model's predictions.
- The third row and column would correspond to the fire class, which comprises all data points that genuinely fire, regardless of the model's predictions.
- The fourth row and column would correspond to the smoke class, which encompasses all data points that are genuinely smoke, irrespective of the model's predictions.
- The fifth row and column would correspond to the human class, which includes all data points that are genuine humans, irrespective of the model's predictions.

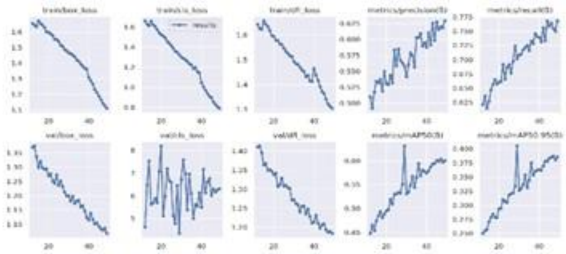

Figure 8:train and validation loss of each class

The graph in the figure displays the loss values for various stages of the YOLOv8 training process, such as box loss, class loss, and dfl loss, for both the training and validation sets.

The x-axis likely represents the number of epochs or iterations of the training process, as indicated by the numbers 20 and 40.

The y-axis displays the value of the loss metrics, such as box_loss, cls_loss, and dfl_loss, for the training and validation sets.

The graph may also display some performance metrics for the YOLOV 8 model, such as precision, recall, and mAP50, which are commonly used to evaluate object detection models.

Validation results

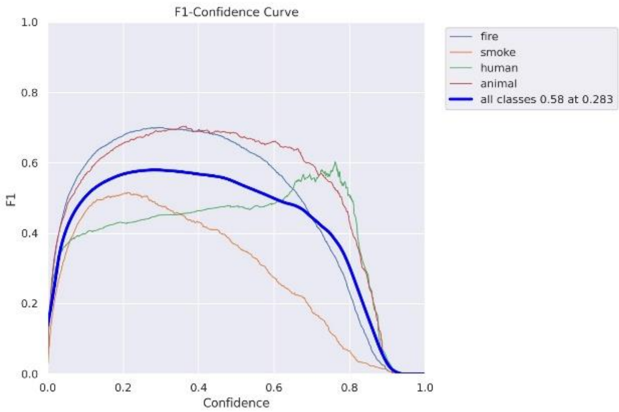

Figure 9: F1- Confidence curve

This is figure 9 shows the Precision-Confidence Curve, which shows the relationship between the confidence of the model's prediction and the precision of those predictions. The x-axis represents the confidence score, ranging from 0.0 to 1.0, while the y-axis represents precision, ranging from 0.0 to 1.0. The graph shows four curves for different classes: fire, smoke, human, and animal.

The results show that for all classes, the precision of the model's predictions is generally low, hovering around 0.2 to 0.3, even at high confidence levels. The curve for fire shows the highest precision, with a maximum precision of 0.58 at a confidence score of 0.283. The curve for smoke shows the lowest precision, with a maximum precision of 0.37 at a confidence score of 0.667. These results indicate that the model needs to be more accurate at making predictions and has difficulty distinguishing between different classes.

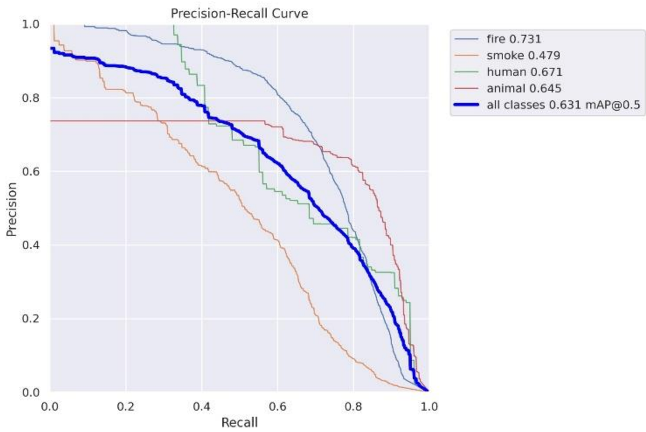

Figure 10: Precision recall curve of the validation result

The values in the g figure 10 represent the average precision scores for each class and the mean average precision (mAP) for all classes at a threshold of 0.5. The precision-recall curve can be used to visualize the model's performance and to choose an appropriate threshold based on the desired trade-off between precision and recall. A high precision score indicates that most of the positive predictions made by the model are correct. In contrast, a high recall score indicates that the model detected most of the positive instances in the dataset.

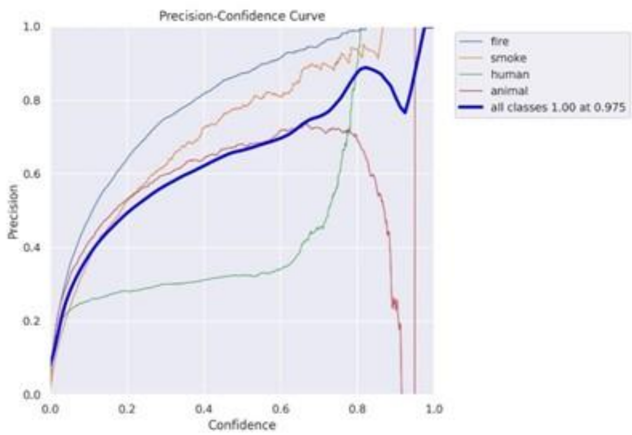

Figure 11: Precision-Confidence Curve

The values in the graph in figure 11 represent the average precision scores for each class and the mean average precision (mAP) for all classes at a threshold of 0.5. The precision-recall curve can be used to visualize the model's performance and to choose an appropriate threshold based on the desired trade-off between precision and recall. A high precision score indicates that most of the positive predictions made by the model are correct. In contrast, a high recall score indicates that the model detected most of the positive instances in the dataset.

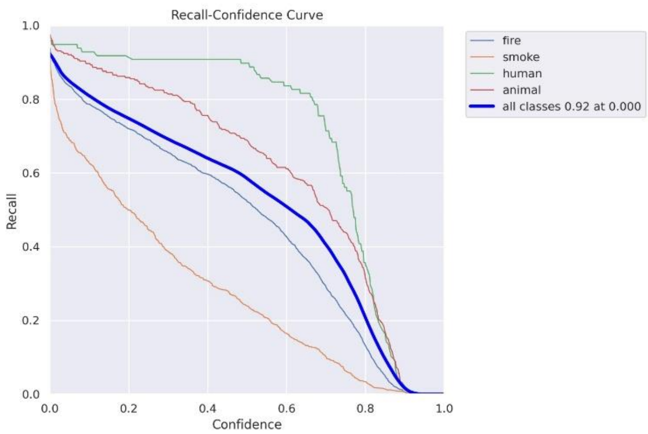

Figure 12: Recall-Confidence Curve

This graph shows the Recall vs. Confidence curve for a detection model's validation results. The x-axis represents the confidence score of the detections. In contrast, the y-axis represents the recall, the fraction of true positive instances detected by the model.

The graph shows that for all classes (fire, smoke, human, and animal), the recall is very high (close to 1) when the confidence is low (close to 0). This means that the model can detect most of the instances of these classes even when it is not very confident about its detections.

As the confidence score increases, the recall gradually decreases, indicating that the model becomes more selective in its detections and may miss some instances. At a high confidence score (0.92), the recall drops to 0 for all classes, indicating that the model did not detect any instances at this confidence threshold.

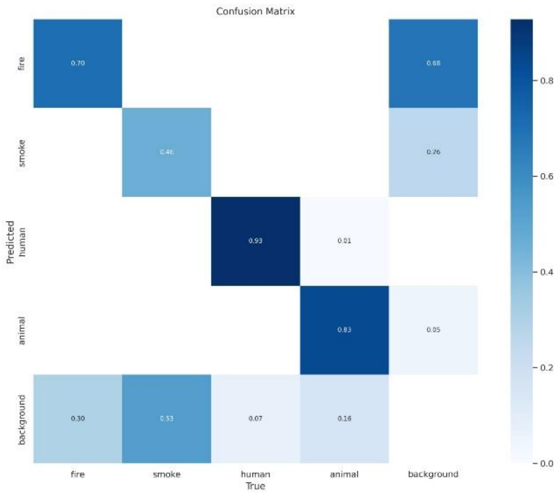

Figure 13: Confusion matrix (validation result)

- True Positives (TP): The number of correct predictions for each class. For example, the model predicted 0.70 instances of "fire" correctly.
- False Positives (FP): The number of incorrect predictions for each class. For example, the model predicted seven instances of "smoke" when it was "fire."
- False Negatives (FN): The number of instances of a certain class that the model did not predict as such. For example, two instances of "human" were not predicted by the model.
- True Negatives (TN): The number of instances that were not of a certain class and were correctly not predicted as such by the model. For example, 11 instances were not "animal" and were not predicted as such by the model.

A. *Testing Results*

To test the results of the forest fire detection system, we use videos that contain smoke, fire, animals, and humans, with different levels of accuracy.

For example, In figures 6, 9 the model detects smoke in an image with 52% confidence.

Figure 14: Fire Detection

Figure 15: Smoke and Fire Detection

Overall, the results of the detection system demonstrate its potential to assist in forest fire detection and prevention by providing an automated system for identifying potential hazards. However, further improvements are needed to increase its accuracy and reliability in detecting all types of objects and regions of interest in the videos.

B. Discussion

The present study aimed to develop a YOLOv8 model to detect animals and humans in forest fires. However, the recall for the "fire" class was relatively low, suggesting that the model may have missed some fires in the validation dataset. One of the main challenges in training object detection models for forest fires is the large variability in data due to the different environments, times of day, and weather conditions under which fires can occur, which can affect the appearance of flames, smoke, and other objects in the scene, making it difficult for object detection models to generalize well to new data[22]–[24].

To address this challenge, the YOLOv8 architecture was employed, which includes deformable convolutional networks and spatial pyramid pooling to improve the model's accuracy and efficiency. However, the model's performance may be affected by factors such as weather conditions, camera angle, and lighting conditions, which were not explicitly accounted for in this study. Moreover, the model was only trained on a relatively small dataset, limiting its ability to generalize to new, unseen data[25]–[27].

Future work may involve collecting larger and more diverse datasets for training and validation, exploring different object detection algorithms and architectures, and incorporating thermal imaging to improve fire detection accuracy. Additionally, collecting data under different weather conditions and integrating the model with other technologies, such as UAVs, could enhance the speed and efficiency of forest fire detection and response.

Conclusion

In conclusion, the YOLOv8 model trained in this study showed promising results in detecting animals and humans in forest fires.

One of the main limitations of the study is the relatively small size of the dataset used for training and validation. This could lead to overfitting and limited generalizability of the model to other scenarios. To address this, it is recommended to collect and use larger datasets that cover a wider range of scenarios to improve the model's performance.

Another limitation is the relatively low recall for the "fire" class, which could be improved by using more sophisticated techniques such as data augmentation, transfer learning, or fine-tuning of the model. It is also important to note that the performance of the model could be affected by different factors such as the quality of the input images, the lighting conditions, and the presence of occlusions.

References

[1] F. M. A. Hossain, Y. M. Zhang, and M. A. Tonima, “Forest fire flame and smoke detection from UAV-captured images using fire-specific color features and multi-color space local binary pattern,” J Unmanned Veh Syst, vol. 8, no. 4, pp. 285–309, 2020.

[2] M. Yandouzi et al., “Review on forest fires detection and prediction using deep learning and drones,” J. Theor. Appl. Inf. Technol, vol. 100, no. 12, pp. 4565–4576, 2022.

[3] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, “A forest fire detection system based on ensemble learning,” Forests, vol. 12, no. 2, p. 217, 2021.

[4] S. Mohnish, K. P. Akshay, P. Pavithra, and S. Ezhilarasi, “Deep Learning based Forest Fire Detection and Alert System,” in 2022 International Conference on Communication, Computing, and Internet of Things (IC3IoT), 2022, pp. 1–5.

[5] S. Goyal, M. Shagill, A. Kaur, H. Vohra, and A. Singh, “A yolo-based technique for early forest fire detection,” Int. J. Innov. Technol. Explore. Eng, vol. 9, pp. 1357–1362, 2020.

[6] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, “A forest fire detection system based on ensemble learning,” Forests, vol. 12, no. 2, pp. 1–17, Feb. 2021, doi 10.3390/f12020217.

[7] Y. G. Sahin, “Animals as mobile biological sensors for forest fire detection,” Sensors, vol. 7, no. 12, pp. 3084–3099, 2007.

[8] Y. G. Sahin, “Animals as mobile biological sensors for forest fire detection,” Sensors, vol. 7, no. 12, pp. 3084–3099, 2007.

[9] W. S. Mseddi, R. Ghali, M. Jmal, and R. Attia, “Fire detection and segmentation using YOLOv5 and U-net,” in 2021 29th European Signal Processing Conference (EUSIPCO), 2021, pp. 741–745.

[10] A. Chopde, A. Magon, and S. Bhatkar, “Forest Fire Detection and Prediction from image processing using RCNN,” in Proceedings of the 7th World Congress on Civil, Structural, and Environmental Engineering, Virtual, 2022, pp. 10–12.

[11] A. Chopde, A. Magon, and S. Bhatkar, “Forest Fire Detection and Prediction from image processing using RCNN,” in Proceedings of the 7th World Congress on Civil, Structural, and Environmental Engineering, Virtual, 2022, pp. 10–12.

[12] A. Chopde, A. Magon, and S. Bhatkar, “Forest Fire Detection and Prediction from image processing using RCNN,” in Proceedings of the 7th World Congress on Civil, Structural, and Environmental Engineering, Virtual, 2022, pp. 10–12.

[13] A. Chopde, A. Magon, and S. Bhatkar, “Forest Fire Detection and Prediction from image processing using RCNN,” in Proceedings of the 7th World Congress on Civil, Structural, and Environmental Engineering, Virtual, 2022, pp. 10–12.

[14] J. Jiang, L. Chen, S. Jiang, G.-Q. Li, and A. Usmani, “Fire safety assessment of super-tall buildings: A case study on Shanghai Tower,” Case Studies in Fire Safety, vol. 4, pp. 28–38, 2015, doi: https://doi.org/10.1016/j.csfs.2015.06.001.

[15] J. Jiang, L. Chen, S. Jiang, G. Q. Li, and A. Usmani, “Fire safety assessment of super-tall buildings: A case study on Shanghai Tower,” Case Studies in Fire Safety, vol. 4, pp. 28–38, Oct. 2015, doi: 10.1016/j.csfs.2015.06.001.

[16] Z. Xue, H. Lin, and F. Wang, “A small target forest fire detection model based on YOLOv5 improvement,” Forests, vol. 13, no. 8, p. 1332, 2022.

[17] F. Shi, H. Qian, W. Chen, M. Huang, and Z. Wan, “A fire monitoring and alarm system based on YOLOv3 with OHEM,” in 2020 39th Chinese Control Conference (CCC), 2020, pp. 7322–7327.

[18] Z. Xue, H. Lin, and F. Wang, “A small target forest fire detection model based on YOLOv5 improvement,” Forests, vol. 13, no. 8, p. 1332, 2022.

[19] Z. Jiao et al., “A deep learning based forest fire detection approach using UAV and YOLOv3,” in 2019 1st International Conference on industrial artificial intelligence (IAI), 2019, pp. 1–5.

[20] Z. Jiao et al., “A deep learning based forest fire detection approach using UAV and YOLOv3,” in 2019 1st International Conference on industrial artificial intelligence (IAI), 2019, pp. 1–5.

[21] H. Hongyu, K. Ping, F. Li, and S. Huaxin, “An improved multi-scale fire detection method based on convolutional neural network,” in 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2020, pp. 109–112.

[22] A. Abdusalomov, N. Baratov, A. Kutlimuratov, and T. K. Whangbo, “An improvement of the fire detection and classification method using YOLOv3 for surveillance systems,” Sensors, vol. 21, no. 19, p. 6519, 2021.

[23] A. Sharma, P. K. Singh, and Y. Kumar, “An integrated fire detection system using IoT and image processing technique for smart cities,” Sustain Cities Soc, vol. 61, p. 102332, 2020.

[24] C. Yue and J. Ye, “Research on Improved YOLOv3 Fire Detection Based on Enlarged Feature Map Resolution and Cluster Analysis,” in Journal of Physics: Conference Series, 2021, vol. 1757, no. 1, p. 012094.

[25] P. Sridhar and R. R. Sathiya, “Real Time Fire Detection and Localization in Video Sequences using Deep Learning Framework for Smart Building,” in Journal of Physics: Conference Series, 2021, vol. 1916, no. 1, p. 012027.

[26] Z. Jiao et al., “A yolov3-based learning strategy for real-time uav-based forest fire detection,” in 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 4963–4967.

[27] A. B. Abdusalomov, B. M. S. Islam, R. Nasimov, M. Mukhiddinov, and T. K. Whangbo, “An improved forest fire detection method based on the detectron2 model and a deep learning approach,” Sensors, vol. 23, no. 3, p. 1512, 2023.