Fire Detection and Classification System

Travis Dow*, Sujesh Padhi†, Abdelrahman Elnaggar‡,
Aheraz Bin Muslim Mohammad§, Subroto Kumer Deb Nath¶ and Sufiyan Bukhari‖

Department of Electrical and Software Engineering, University of Calgary, Canada

Email: *tdow@ucalgary.ca, †sujesh.padhi1@ucalgary.ca, ‡abdelrahman.elnaggar@ucalgary.ca,
§aherazbinmuslim.moha@ucalgary.ca, ¶subroto.nath@ucalgary.ca, ‖sufiyanahmed.bukhari@ucalgary.ca

***Abstract*—History has proven that fire hazards are among the most dangerous and frequently occurring incidents worldwide. Numerous designs and implementations for fire detection systems have been done for early fire alerts and warnings. In this project, we developed a system to detect and classify fire hazards more precisely and faster from fire hazard images and videos. The system sets its backbone for feature extraction on YOLOv8 (You Only Look Once) model. We implemented transfer learning using this state-of-the-art model to detect fire and classify the input images into correct classes. Several benchmarks have been evaluated while training the system using a large dataset of fire and non-fire images. A reasonable detection accuracy rate has been achieved with low false-positive rates making it suitable for real-time applications. The dataset used for the system includes various scenarios of fire hazards, ranging from small household fires to extensive forest fires, captured from different angles and distances. The system achieved good accuracy in detecting and classifying fire hazards, with a validation accuracy of 64% and detection accuracy of 59%. We extended the usage by detecting and locating fire on a recorded video. The proposed system can be further expanded for real-time fire detection from surveillance cameras, drones, or unmanned aerial vehicles (UAV) and thus can find applications in household premises, various industrial sectors, and in the forest, grassland, and prairies to detect fire in the early stage to reduce the overall losses significantly.**

***Index Terms*—fire detection system, fire image classification, deep learning for fire detection, transfer learning using yolo for fire detection**

I. Introduction

Fire hazard is a primary concern that can substantially damage property, the environment, wildlife, and human life. Three classifications of fires, namely household, commercial, and wildfires, can have a significant impact. Fires can happen spontaneously or as a result of human activity and have the potential to spread rapidly over vast expanses. In recent years, numerous wildfires have occurred in regions worldwide, scorching millions of acres of land. These fires caused extensive environmental and wildlife harm, emitting considerable amounts of carbon dioxide into the atmosphere and wreaking havoc on ecosystems [1]. Early detection of this household, commercial, and large-scale wildfires disaster is crucial to reduce the damages significantly with an immediate and effective response. In recent years, advanced computer vision technologies have been increasingly applied for fire detection, with promising results [2]. In particular, object detection models such as You Only Look Once (YOLO) [3]–[5] have shown great potential in identifying real-time fire detection [6].

This project aims to develop a Fire Hazard Detection and Classification System using highly optimized, lightweight YOLO version 8 (YOLOv8), effectively detecting and providing alerts of any fire hazard. The model does not demand high-performance computing power for training [7]. The proposed project leverages the high accuracy and speed of YOLOv8 to detect fire in images and video streams captured by surveillance cameras, drones, or unmanned aerial vehicles (UAV). And so it has the substantial prospect of being implemented in surveillance cameras, drones, or unmanned aerial vehicles (UAV) to detect fire in real-time. Such a fire detection system is critical in ensuring public, wildlife, environmental, and ecological safety during a devastating disaster. The system relies on image classification to detect fires. It is trained on a dataset of high-quality images that accurately represent fire output class and bounded boxes on the image locating fire precisely.

The development of this system involved several stages, such as collecting and labeling a large dataset of fire and non-fire images to train the YOLOv8. We implemented Transfer-Learning to reduce the training complexities [8]. We evaluated the overall system performance on the unseen datasets, achieving high detection accuracy and low false-positive rates. By developing an accurate and reliable fire detection system, it is possible to stop spreading fires in the early stages and minimize the property and environmental damages and the loss of lives, as proved over the last few decades [9].

II. Related Work

The criticality of fire hazards takes a tremendous toll on humanity, and numerous work has been conducted over the years to detect and handle such a terrific disaster. Advanced technologies and algorithms using machine learning and deep learning for object classification from images and video footage provide proven effective solutions [10].

A. An early fire-detection method based on image processing

The study suggests a video processing-based technique for the early detection of fires, utilizing a chromatic and disorder measurement approach to identify fire pixels and smoke pixels using an RGB model. The intensity and saturation of the R component determine the decision function for fire pixels. The extracted fire pixels are validated using growth dynamics, disorder, and smoke. Once the alarm-raising condition is met, a fire alarm is generated through an iterative check on the flame's growth ratio. The technique has been shown to achieve automatic surveillance of fire incidents with a reduced false alarm rate and is thus highly applicable in military, social security, and commercial settings. Future work will involve training the raising parameters with a fuzzy neural network, incorporating fire pixels and smoke pixels extracted at intervals to enhance the fire alarm's reliability [11].

B. Convolutional Neural Networks Based Fire Detection in Surveillance Videos

The paper introduces a new approach for detecting fires in surveillance videos using a convolutional neural network (CNN) based on the GoogleNet architecture. The proposed model is fine-tuned to balance detection accuracy and computational efficiency for identifying fires. Compared to existing fire detection methods based on hand-crafted features and the AlexNet architecture, the technique demonstrates superior performance is claimed by the author. However, the rate of false alarms remains high, suggesting the need for further investigation to improve accuracy. The authors emphasize that the proposed framework is particularly suitable for CCTV surveillance systems due to its cost-effectiveness and scalability. The proposed approach could be extended in future research to detect smoke and fire, allowing for more effective surveillance in complex real-time scenarios [12].

C. Image fire detection algorithms based on convolutional neural networks

The research discusses using image fire detection technology to reduce fire losses by detecting fires early. The current detection algorithms, which involve manually or automatically extracting image features, have lower accuracy, delayed detection, and require much computation. In order to improve the performance of the fire-detection system, the paper proposes using advanced object detection CNN models, including Faster-RCNN, R-FCN, SSD, and YOLO v3, to develop novel algorithms for image fire detection. These algorithms automatically extract complex image fire features and can detect fires successfully in different scenes.

The authors emphasize the experiments to show that the proposed algorithms, based on CNNs, are superior to traditional algorithms and can detect fires with higher accuracy and faster speed, especially the algorithm based on YOLO v3, which achieves an accuracy of 83.7% and a detection speed of 28 FPS (Frames Per Second). The proposed algorithms have lower missed detection and false alarm rates, and the differences in average smoke detection precision between the algorithms based on YOLO v3 and Faster-RCNN are insignificant. In summary, the paper demonstrates that using advanced object detection CNN models is a feasible and practical approach to improve the performance of image fire detection technology [13].

D. Wildfire detection for transmission line based on improved lightweight YOLO

This paper addresses the problem of wildfires in transmission line passages, as they pose a severe threat to power security. To tackle this issue, the authors suggest two wildfire detection models based on YOLOv5. However, these models are designed with the limited computing power of embedded terminals in mind, which necessitates the simplification of YOLOv5's network structure. One approach simplifies the neck and head parts, while the other removes backbone modules, significantly reducing the model parameters.

The authors test their models against a dataset of 1993 images and compare them to mainstream models through comprehensive experiments. The results indicate that the proposed models can monitor in real-time on embedded devices while retaining high accuracy and recall rates. The first YOLOv5-based model achieves an average accuracy of 71.5% and a recall rate of 66.2%, surpassing the original YOLOv5 algorithm by 4.2 percentage points on AP. Meanwhile, the second YOLOv5-based model has a smaller size of 0.9 Mb and achieves an average accuracy of 64.2% and a recall rate of 63.0%. Although the accuracy and recall rate decreases slightly, the second model is still capable of real-time monitoring on embedded platforms, providing a valuable reference for detecting wildfires near transmission lines in the future [14].

Recent work shows that convolutional methods provide an efficient manner for fire detection systems. State-of-the-art models such as YOLO have proven to be very effective, where YOLOv3 provides very high precision [15], and YOLOv5 is seen to be faster [4]. This model can fulfill real-time demands more effectively as it is lightweight, requires less computation, and is more easily deployable on systems, thus promoting the practical implementation of the detection model.

III. Materials and Methods

Based on the previous research that has been conducted and given our requirements of faster and more accurate object detection in image/video data, we elected to use the latest version named YOLOv8 model created by Ultralytics [16] and released in January 2023. Based on the latest advancements in deep learning and computer vision, YOLOv8 provides unmatched performance in both speed and accuracy. Its simplified design enables it to be utilized across various applications and effortlessly tailored to diverse hardware platforms, including edge devices and cloud APIs. The benchmarking results of YOLOv8 as a state-of-the-art real-time object detection have been experimented with and look promising [17] [18]. A brief view of the system is shown in figure 1

We utilized a dataset of 9462 images of fire (small, medium, large sized fires), and it was publicly available [19] and the results of research using them looked convincing [20]. Additionally, it was pre-annotated with bounding boxes and classification labels, demarcating the fire within the images.

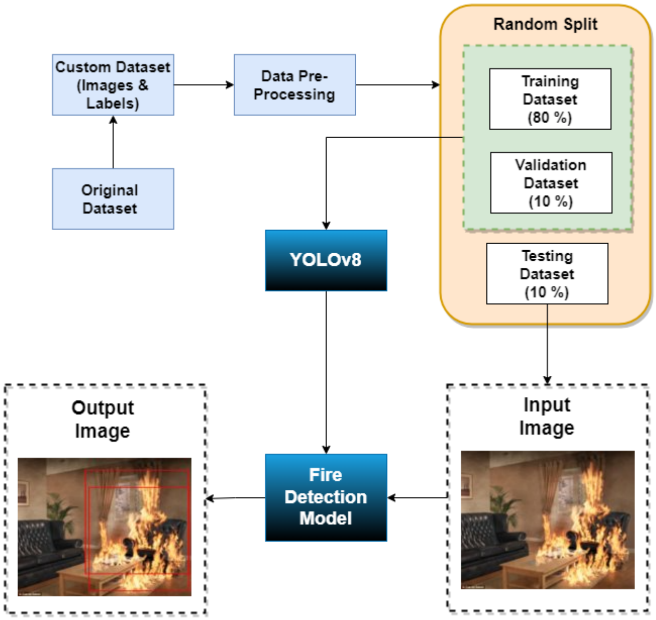

Fig. 1. Block Diagram of the Fire Detection System

For pre-processing, the image labels were converted from their original '.xml' format to '.txt' format for use with the YOLOv8 model. The XML data was parsed, and the bounding boxes and classification label data for all fire objects in the original dataset were extracted. The original bounding box data was in the form of absolute value top-left and bottom-right (x, y) coordinates in image space. In contrast, the requirement for the YOLOv8 model is - normalized centroid (x,y) coordinates, box width, and box length. Each image's bounding boxes demarcating fire were re-calculated and appended to a list to be saved to a '.txt' file. Next, each image was resized and center-cropped to 1280x1280 for uniform input sizes for the model. It is important to note that because the bounding box coordinates were normalized prior to image resizing, the bounding boxes retained their accuracy. Next, a random data split was performed on the dataset for the training, validation, and testing classes using an 80:10:10 split, respectively. Another stratified data split was also performed to perform in-parallel training to compare, understand and identify a more suitable method.

After data pre-processing, the next step was training the model, and we selected the YOLOv8 nano model for increased detection speeds. The model was trained on the Microsoft COCO image dataset [21] and comes with pre-trained weights for object detection. We then trained our YOLOv8 implementation using our customized, pre-processed dataset. The two parallel training was done using the PyTorch deep learning framework, one on an NVIDIA GTX 1060 GPU and another on an NVIDIA RTX 3060 GPU. The training was done in phases, so each time, the best set of weights from the last round was selected and trained for the next round of epochs. The model with random data split was trained for 200 epochs yielding a maximum accuracy of 64% on the training dataset. In contrast, the model with stratified data split was trained for 300 epochs, but the maximum we could achieve was 38% accuracy. Given that the model accuracy did not improve further, we used the best set of weights collected and moved on to the evaluation stage.

Out of both the training methodologies, the random data split combination worked well with better accuracy after 200 epochs of training compared to the stratified data split, which was run for 300 epochs, as shown in table I. Comparing the results, we went ahead with the random split model, which was trained for 200 epochs and achieved 64% accuracy. It is important to note that the accuracy metric here represents the intersection over union (IoU) accuracy of the bounding boxes as shown in figure 2 and not the classification. Therefore 64% represents a high confidence level similar to the benchmark model's achievement results on large datasets [22].

TABLE I
FIRE DETECTION ACCURACY COMPARISON

No. of Epochs	Accuracy	
	Random Split	*Stratified Split*
200	0.64	0.34
300	x	0.38

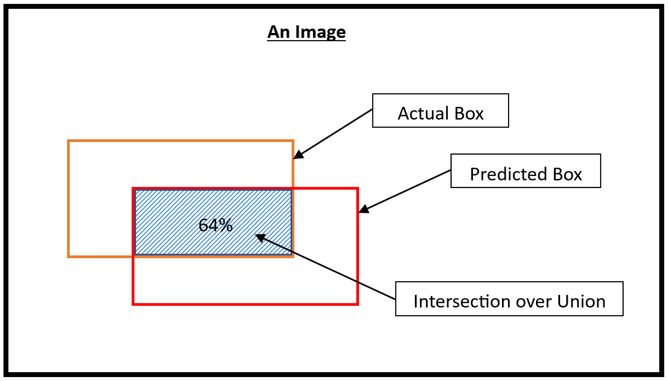

Fig. 2. Accuracy Measurement using Intersection over Union

For evaluation, we conducted validation tests using our testing dataset, which the model had just seen. The model achieved an accuracy metric of 59% on the testing dataset.

IV. RESULTS AND DISCUSSION

In evaluating the performance of our object detection model, we used three metrics: precision, recall, and accuracy. Accuracy measures the correct rate of predictions of the target class in the data set. The results are displayed in Figures 2 and 4 for the training and testing data sets, respectively. Precision measures the rate of correct predictions of the target class given all of the model's predictions and is, therefore, inversely proportional to the false-positive rate. Our model achieved a precision value of 1.0 at a confidence rate of 0.868, meaning no false-positive predictions were created above that confidence level. Recall measures the rate of correct predictions of the target class given the actual representation of the target class in the data set. It is, therefore, inversely proportional to the false-negative rate. Unfortunately, we could only achieve a recall value greater than 0.90 at a negligible confidence value, meaning our model was always making some false-negative predictions. The precision-recall curve generated for the test set (Figure 5) illustrates the trade-off between these two metrics with a calculated mean average precision (mAP) of 0.562 at a confidence level of 0.50 in the testing dataset and mAP of 0.590 and 0.264 at confidence levels of 0.50 and 0.95 respectively in the training dataset. Our calculated F1-score was 0.576 for object detection. Given the costs associated with potentially not predicting a fire versus the costs of falsely predicting a fire, we reason the false-negative rate, and therefore the recall value, is likely the more important metric when considering our results.

A. Training Results

The prediction images show accurate bounded boxes around the fire located within them. The training data was conducted on 80% of the total dataset, and the validation dataset gives quick feedback to the model about the training performance. We validated 10% of the total dataset, and the accuracy for fire detection was 64% from the confusion matrix shown in Figure 3. The Precision-Recall score or also known as PR Score, covers more significantly than 60% of the graph as shown in Figure 4 and symbolizes the ability of the model to majorly detect the location of the fire in the image under evaluation for fire detection setting an expectation or benchmark for the model to perform similarly with a completely unknown testing dataset.

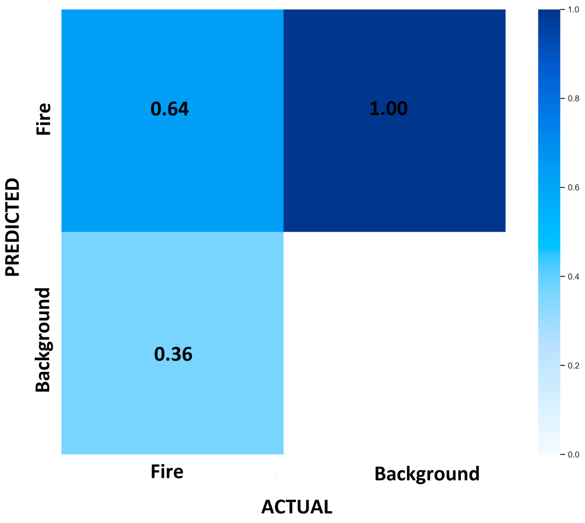

Fig. 3. Confusion Matrix from Validation dataset

B. Testing Results

The testing dataset was the remaining 10% of the whole dataset, and an accuracy of 59% was observed, which is very close to the validation accuracy of 64%. The confusion matrix can be seen in Figure 5. The Precision-Recall score for the testing dataset looks similar to the validation score shown in Figure 6. This confidence symbolizes the model's ability to detect the fire's location and draw boxes around it in the unknown dataset images under evaluation for fire detection.

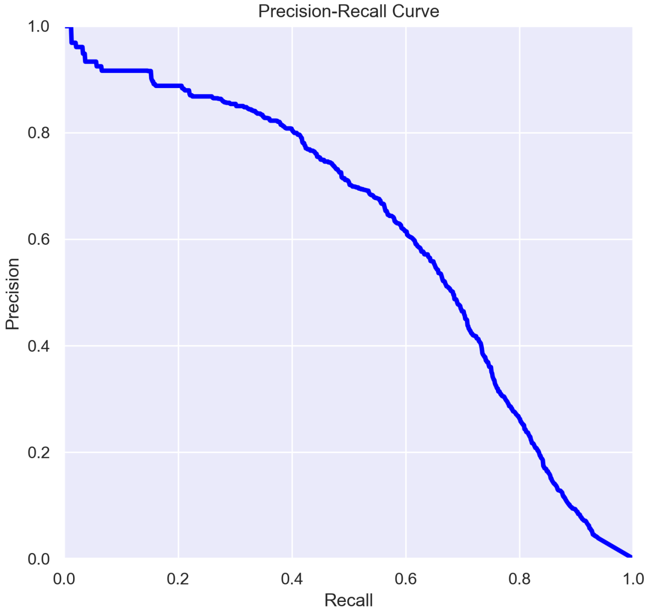

Fig. 4. Precision-Recall Curve from Validation dataset

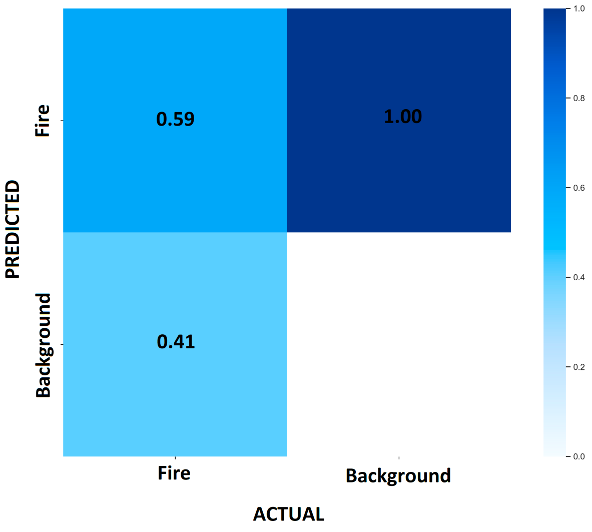

Fig. 5. Confusion Matrix from Testing dataset

C. Analysis

Compared to the model benchmark from the validation score of 64%, as observed in the Figure 3, the completely unknown testing dataset scores 59%, which is closely accurate.

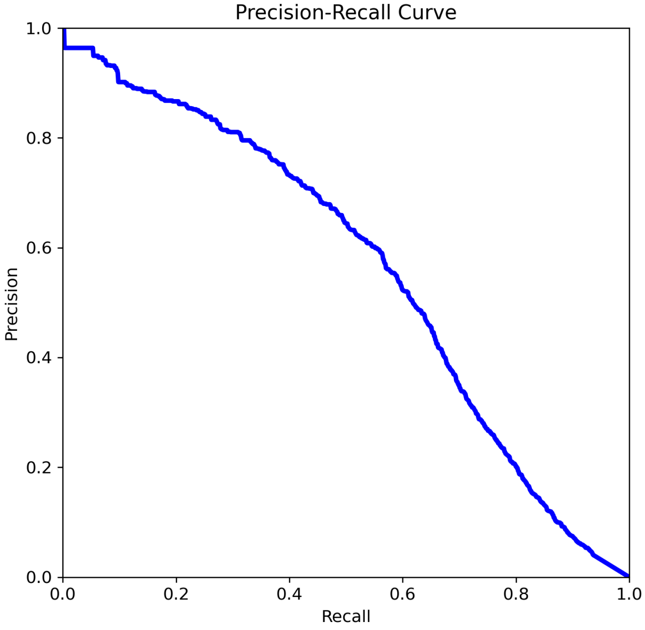

Fig. 6. Precision-Recall Curve from Testing dataset

This model also suggests over-fitting from the training and validation sets (i.e., an accuracy loss of 5% between training and testing). Again, however, given that our problem space is object detection, 59% represents high confidence.

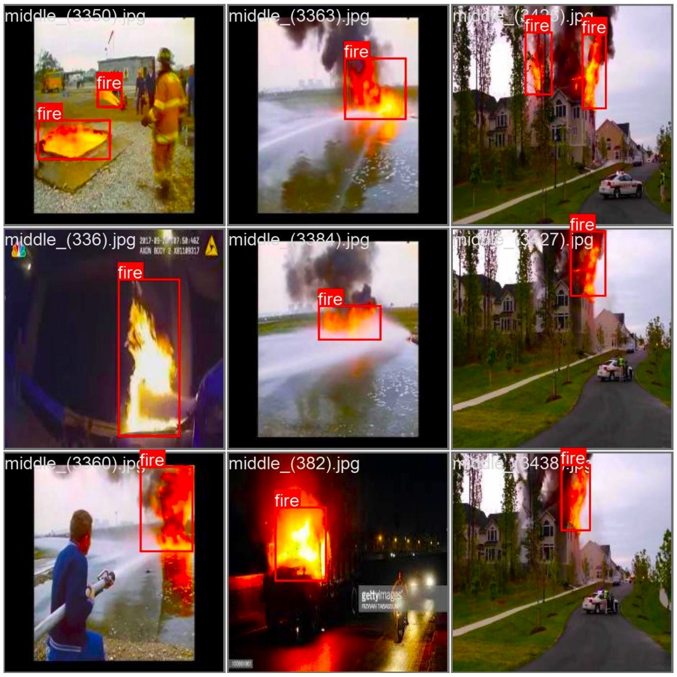

Fig. 7. Validation Dataset - Actual Images

An example of an actual validation image is shown in figure 7 and the corresponding predicted images in figure 8. A comparison of both images shows that the fire detection was very accurate, with a confidence score greater than 0.5 up to 0.8, which can be seen on the predicted images for each bounded box of fire. Some of the images contain multiple fire locations with separate bounded boxes. Similar behavior is observed with the testing dataset as observed in the figure 9 for the actual co-ordinate of the bounded box and figure 10 for the corresponding values of the predicted bounded box location. The model measures the overlapping area of the actual and predicted bounding boxes to calculate the accuracy/confidence score in the confusion matrix observed in Fig 3 and Figure 5. Thus, as observed in the above figures, a medium accuracy of 64% for validation and 59% for testing can still detect fire precisely on an image. We collected numerous metrics from our training and testing iterations,

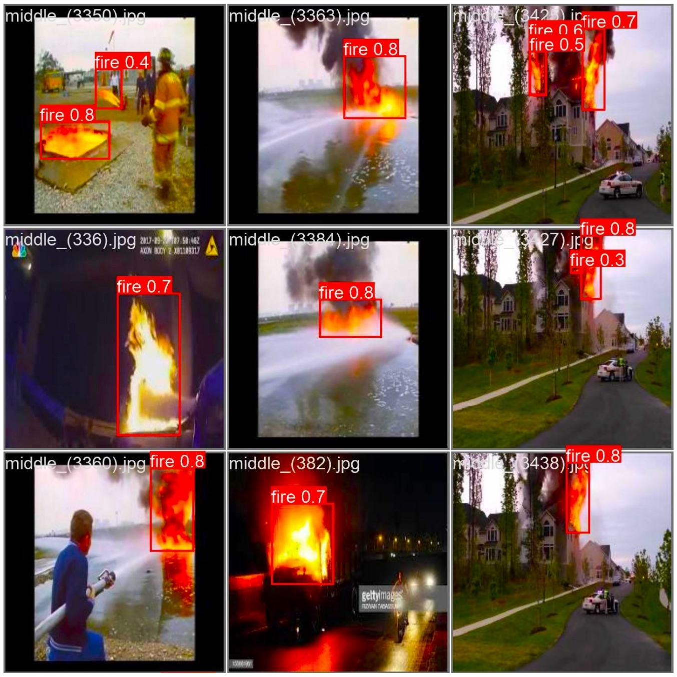

Fig. 8. Validation Dataset - Predicted Images

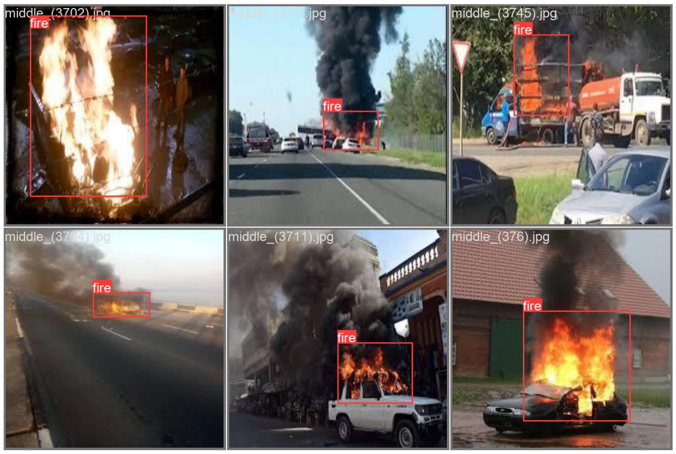

Fig. 9. Testing Dataset - Actual Images

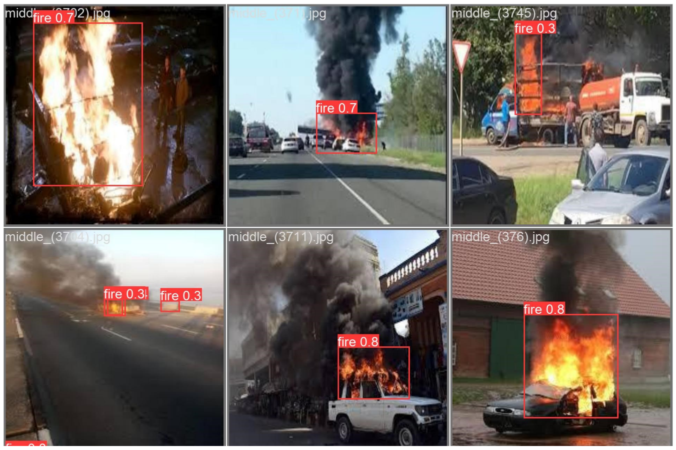

Fig. 10. Testing Dataset - Predicted Images

and all the results are located in the GitHub repository (https://github.com/ttdow/ENEL645/tree/main/runs/detect). Given that we chose object detection versus classification, we believe a testing accuracy value of 0.59 represents a passable fire detection system. Additionally, we achieved an average prediction speed of 23.47ms per image using a GTX 1060 GPU and 501.4ms per image using an i5-1135G7 @ 2.4GHz CPU. While we were interested in testing our model on real-time video data, we needed help locating a publicly-available, pre-annotated fire video data set. Due to time constraints for this project, we could not produce our own annotated fire video data set. However, we tested our model using real-time unlabelled video data, which performed well qualitatively. Our model achieved speeds of 23.47ms per image using a GTX 1060 GPU, which corresponds to 42.6 frames-per-second for video data - more than adequate for a real-time fire video detection system. Unfortunately, no additional quantitative metrics are available to report due to the absence of labels. Further research could be conducted by creating a novel data set of videos of fire labeled with bounding box data for our model to be evaluated against. This research would likely be a welcome contribution given the sparsity of publicly available fire video data sets.

Conclusions

Through this project, we proposed a Fire Detection and Classification System using YOLOv8, which could be trained and evaluated on a non-GPU system to predict fire on images and videos precisely and accurately. The system is lightweight and faster and has proved to be an effective solution for a timely response and damage mitigation. Overall, the proposed system provides an effective solution for quick fire detection and rapid response, potentially saving lives and reducing property damage significantly.

Future research could focus on further training and optimization with a much larger dataset for increasing the bounded box accuracy to go beyond 90%. Additionally, the system's capabilities could be enhanced for fire detection on a live video, under different lighting conditions, or detecting fires in crowded environments. The system's implementation in real-world scenarios could be further explored to ensure its reliability and usability.

References

[1] E. Newburger, "Australia fires kill half a billion animals as crisis mounts," Jan. 2020. [Online]. Available: https://www.cnbc.com/2020/01/03/australia-fires-nearly-half-a-billion-animals-killed-as-crisis-mounts.html

[2] N. Grammalidis, E. Çetin, K. Dimitropoulos, F. Tsalakanidou, K. Kose, O. Gunay, B. Gouverneur, D. Torri, E. Kuruoglu, S. Tozzi, A. Benazza, F. Chaabane, B. Kosucu, and C. Ersoy, "A multi-sensor network for the protection of cultural heritage," in *2011 19th European Signal Processing Conference*, 2011, pp. 889–893.

[3] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922001363

[4] K. Liu, H. Tang, S. He, Q. Yu, Y. Xiong, and N. Wang, "Performance validation of yolo variants for object detection," in *Proceedings of the 2021 International Conference on Bioinformatics and Intelligent Computing*, ser. BIC 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 239–243. [Online]. Available: https://doi.org/10.1145/3448748.3448786

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[6] S. Wu and L. Zhang, "Using popular object detection methods for real time forest fire detection," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 01, 2018, pp. 280–284.

[7] R. Huang, J. Pedoeem, and C. Chen, "Yolo-lite: A real-time object detection algorithm optimized for non-gpu computers," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 2503–2510.

[8] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020.

[9] P. Jain, S. C. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan, "A review of machine learning applications in wildfire science and management," *Environmental Reviews*, vol. 28, no. 4, pp. 478–505, Dec. 2020, publisher: NRC Research Press. [Online]. Available: https://cdnsciencepub.com/doi/full/10.1139/er-2020-0019

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[11] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *2004 International Conference on Image Processing, 2004. ICIP '04.*, vol. 3, 2004, pp. 1707–1710 Vol. 3.

[12] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18 174–18 183, 2018.

[13] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," *Case Studies in Thermal Engineering*, vol. 19, p. 100625, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214157X2030085X

[14] H. He, Z. Zhang, Q. Jia, L. Huang, Y. Cheng, and B. Chen, "Wildfire detection for transmission line based on improved lightweight YOLO," *Energy Reports*, vol. 9, pp. 512–520, Mar. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484722023708

[15] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[16] "YOLOv8 Docs." [Online]. Available: https://docs.ultralytics.com/

[17] "YOLOv8 Ultralytics: State-of-the-Art YOLO Models," Jan. 2023. [Online]. Available: https://learnopencv.com/ultralytics-yolov8/

[18] C. Bhalerao, "YOLO v8! The real state-of-the-art?" Jan. 2023. [Online]. Available: https://medium.com/mlearning-ai/yolo-v8-the-real-state-of-the-art-eda6c86a1b90

[19] S. Wu, X. Zhang, R. Liu, and B. Li, "A dataset for fire and smoke object detection," *Multimedia Tools and Applications*, pp. 1–20, 2022.

[20] R. Liu, S. Wu, and X. Lu, "Real-time fire detection network for intelligent surveillance systems," in *2nd International Conference on Computer Vision, Image, and Deep Learning*, vol. 11911. SPIE, 2021, pp. 225–238.

[21] J. Solawetz, F. JAN 11, and . . M. Read, “What is YOLOv8? The Ultimate Guide.” Jan. 2023. [Online]. Available: https://blog.roboflow.com/whats-new-in-yolov8/

[22] “Papers with Code - COCO Benchmark (Real-Time Object Detection).” [Online]. Available: https://paperswithcode.com/sota/real-time-object-detection-on-coco