International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Applying Multi-YOLO for Enhanced Product and Fire Detection in Image Analysis

Hai Tran Son

Submitted: 21/12/2023 **Revised**: 27/01/2024 **Accepted**: 09/02/2024

Abstract: With its wide range of uses and intense research interest, computer vision presents a challenging problem when it comes to product and fire detection in images. In addition to providing useful applications including improving consumer product information, enabling image-based rapid payments, automating product availability management, and building early fire warning systems, this task entails identifying goods and fire in photographs with diverse backdrops. However, there is a problem with the widely held belief in product detection research, which holds that training data should reflect actual situations. The effectiveness of product detection systems is impacted by the fact that testing data obtained in a variety of contexts does not match training data, which is frequently gathered under perfect conditions. This work presents a deep learning method for image-based product detection in response to these difficulties. To identify products in photos, the suggested model, known as Multi-YOLO, makes use of several YOLO models. Every element operates as a separate YOLO model, and Fusion rules combine their outputs to create a single output. The experimental results show how well the suggested model works, especially when applied to our collection of product photos, and emphasize its potential for reliable product detection in practical settings. Furthermore, the study's integration of the Multi-YOLO model within a comprehensive early fire alert system paves the way for enhanced fire prevention strategies and improved public safety outcomes.

Keywords: *fire detection, product recognition, deep learning, Multi-YOLO, decision fusion*

1. Introduction

The detection of products in photographs is an important task in many real-world applications, especially in the last few years with the widespread use of high-quality cameras in mobile devices. Many companies are looking for apps that can recognize objects in photos and classify them correctly according to labeling. This capacity is the foundation for the creation of a wide range of apps that improve user experiences and corporate operations. Product detection is regarded as essential to the operation of these systems and is the first stage in the process. Determining the product's location in the image and categorizing its label name constitute the two primary tasks that make up the problem of product detection. In real life, this issue can be extremely challenging because of the wide variations in viewpoint, posture, illumination, and most importantly, occlusion. Time processing is another issue that the product detection system must deal with. Real-time processing is required for this technology to be used in actual applications. In recent years, deep learning has been widely used in object detection tasks. There are many models proposed to solve this problem. At present, two main methods of deep learning have been introduced. These two main methods are based on the pipeline of processing. One method of object detection is based on two stages. The first stage identifies the position of the object and the other classifies the label of the object separately. The representative of this method is R-CNN [2], Fast R-CNN [3], and Faster R-CNN [4]. On the other hand, another method uses only one stage to both identify the position and classify the label. The representative of this method is YOLO [5,6,7,8], SSD [11]. YOLO is one of the deep learning models that earn a lot of interest from researchers due to the performance of this model adapting the real-time requirement but in many various contexts, the accuracy of YOLO is less potential. Therefore, in this paper, we proposed a model called Multi-YOLO based on the YOLO model. We also proposed a method for synthesizing the result for each YOLO component which we called a Fusion rule. The experimental result of our proposed model on our dataset has proved the potential of this model.

2. Background and Related Work

In the past decade the development of Convolution Neural Networks (CNN) which is a special architecture of neural networks proposed by Yann Lecun [1]. A lot of research is applied in the computer vision area by using CNN and Deep Learning and receiving a high accuracy result. A Convolutional Neural Network (CNN) consists of convolution layers, leveraging the convolution operator to extract a multitude of features and gather valuable information. Each convolution layer employs a kernel for the convolution operator, allowing the model to autonomously learn the most pertinent information without relying on manually crafted features. In various

HCM City University of Education
Ho Chi Minh City, Vietnam
haits@hcmue.edu.vn

computer vision challenges, particularly in object detection scenarios, the attainment of crucial features directly correlates with improved detection accuracy. Within the realm of Deep Learning and CNN, numerous research endeavors have been put forth, enabling the amalgamation of these studies into two distinct methods based on the overall pipeline. One method separates two main tasks into two stages. In the first stage, the model will identify and propose the region of interest (RoI). In the second stage, the model will perform region classification and location refinement on that RoI in the previous stage. Some common deep learning models for this method are R-CNN [2], Fast R-CNN [3], Faster R-CNN [4], etc. This method has quite good accuracy but time-consuming is hard due to the complexity of the architecture. Even though the Faster R-CNN has proposed a Region Proposal Network (RPN) and combines it into a complete network, the model becomes end-to-end and increases the speed of the model. However, the RoI layer is not sharing the parameters so the detection speed of the model still does not adopt the requirement for the real-time application.

The other method does not need to generate a region proposal, it uses only one neural network for two missions. By using only one neural network the performance of this method is faster also the flow was much more concise and simpler than the first method. Some common deep learning model of this method is YOLO [5-8, 10], SSD [11], etc. The basic idea of YOLO is to divide the input image into many bounding boxes of size SxS, and each of these bounding boxes will take responsibility for classifying an object if that bounding box contains an object. Each bounding box has its confidence score which is the score that represents the probability that the box contains the object. This prediction can generate a lot of overlapping bounding boxes so an algorithm called Non-Maximal Suppression (NMS) was proposed. In NMS, only the bounding box of the same object with the highest confidence score will be kept.

Object detection is a problem that may be divided into two main categories: locating objects in images and categorizing the labels that belong to these items. To achieve these important goals, a substantial amount of research has been conducted, mostly using deep learning and Convolutional Neural Networks (CNN).

A two-stage pipeline for product object identification and recognition was developed by Ankit Sinha et al. [11]. It consists of an object localizer based on Faster-RCNN that locates object regions in the rack image, and an image encoder based on RestNet-18 that assigns the identified regions to the relevant classes.

In [12] the authors have applied the YOLOv2 [6] to build a model to identify Coca-Cola products on store shelves. Majdi et al. [13] applied the YOLOv3 model [7] in combination with image processing methods to build a system to manage the number of products on the shelf.

Hurtik et al. [14] used the YOLO model for the product identification system in the store but changed the rectangular bounding box with a quadrilateral bounding box to remove the excess background inherent in the YOLO model.

To apply the product identification problem in practice, the model's performance requirements must be met in real-time. At the same time, the system's accuracy requirement is an important consideration. As a result, Therefore, the YOLO model is a deep learning model that is considered more suitable for practical use.

3. Proposed Model

In this paper, we will propose a model called Multi-YOLO for object detection and apply this model to product detection problems. The model is based on the YOLO model because its performance can adapt to the requirement in real time. However, the accuracy of the model seems to rapidly decrease when the context of the testing data is different from the training data. The Multi-YOLO that we proposed enhances this but still guarantees the real-time requirement. C.Multi-YOLO for object detection The Multi-YOLO model is based on the YOLO architecture, each YOLO component gives an independent result about the detection and then we merge the result of those YOLO to provide the final prediction. The rule for merging the result is called the Fusion rule. In general, each YOLO component will be trained on a variant version of the dataset. As a result, these components can provide a different opinion from their point of view. Each result can be considered as the result of different experts. Aggregating those results, we can create a diverse set of opinions that will rapidly increase the efficiency of the accuracy of the model. The basic architecture of the Multi-YOLO model for object detection can be represented as shown in Fig 1.

- Let Y be the input image dataset.
- Let n be the number of YOLO components.
- Let m be the number of classes.

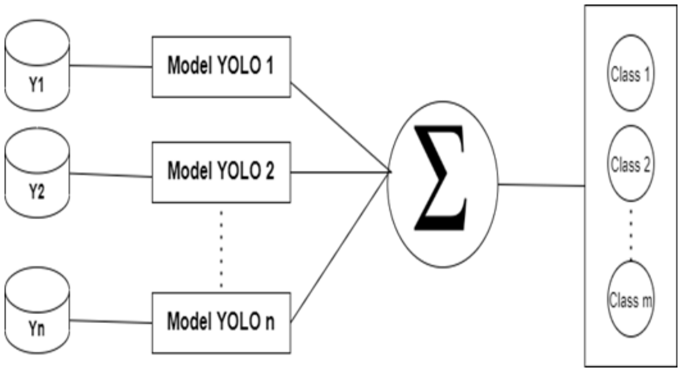

Fig 1. Proposed Multi-YOLO architecture

Multi-YOLO for product detection We proposed the Multi-YOLO architecture to detect a product consisting of two YOLO components as shown below:

●One component is a YOLO that is trained on a high-quality dataset that is collected in good condition. It's referred to as YOLO-Normal.

●One component is a YOLO that is trained on a low-quality dataset collected in practice. It's referred to as YOLO-Bad.

The YOLO-Normal model was aimed at building a model that has good generalization knowledge. On the other hand, the YOLO-Bad model aimed to build a model with good knowledge in recognizing the unusual cases of images. Then the results of the two models will be merged to give the final detection result. This architectural diagram is shown in Fig. 2.

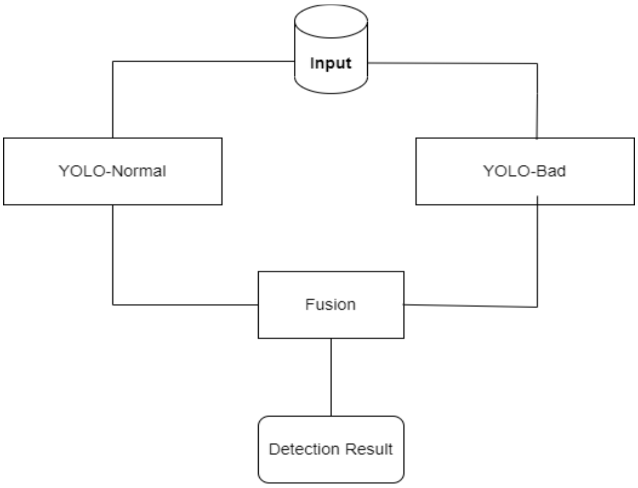

Fig 2. Multi-YOLO architecture for product detection

Fusion rules As mentioned above, in the Multi-YOLO model, we have used an association rule to compute each YOLO component result which we called the Fusion Rule. In this section, we describe it in detail about it.

(i) Fusion rule by max confidence (Multi-YOLO-MC) For each YOLO component output, we get all the bounding boxes and their corresponding confidence score, after that we will use the NMS algorithm to reduce the redundant bounding boxes. Finally, the bounding box with the highest confidence score will be output.

(ii) Fusion rule by Alpha coefficient (Multi-YOLO-AC) For each YOLO component output, we get all the bounding boxes after using the NMS algorithm to reduce all the redundant bounding boxes. Then the final prediction was calculated by using this formula:

$$\alpha * \text{YOLO-Normal} + (1- \alpha) * \text{YOLO-Bad}$$

Where α runs from 0.1 to 0.9 with a jump of 0.1, then we will choose α at which the Multi-YOLO model gives the best results.

4. Experimental and Discussion

In this study, our dataset is the product images of some products in the supermarket that were collected and labeled with the help of the expert who is working in the supermarket. The dataset has 5 labels which are 0 for

Downy, 1 for Safeguard, 2 for Palmolive, 3 for Rejoice, and 4 for Head Shoulder as shown in Fig. 3.

For training purposes, we have 2 datasets one is for training YOLO-Normal and one for training YOLO-Bad. Each dataset contains data about 5 labels and some images that have no object to reduce the false negative. The training dataset is split into a training dataset and a valid dataset with a ratio of 8:2 for choosing the best model for training process purposes. The YOLO-Normal training dataset contains 2975 images with these products, and The YOLO-Bad training dataset contains 1464 images with these products shown in Table I.

Table I. Training dataset for yolo-normal

Label	Number of images for YOLO-NORMAL	Number of Images for YOLO-BAD
Class 0	699	346
Class 1	577	291
Class 2	683	314
Class 3	536	276
Class 4	480	237

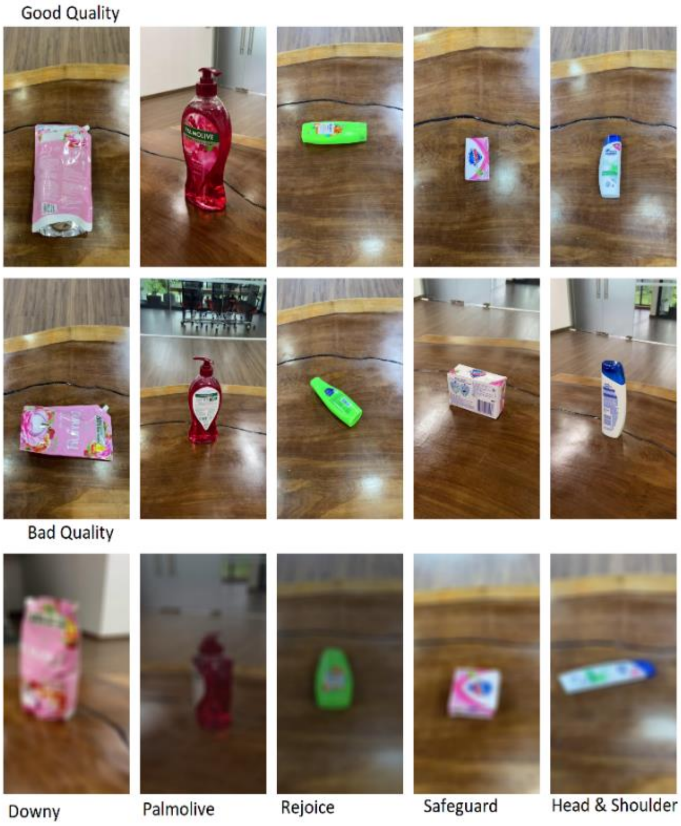

Fig 3. Sample image in Training Dataset

For testing purpose, we will have 5 test set as follow.

● Private-Test-1: this test set contains images of the product under normal conditions for evaluating mAP@0.50 of the model in normal conditions.

●Private-Test-2: this test set contains images of the product with different angles to evaluate mAP@0.50 of the model when the product comes up with different angles in the image.

●Private-Test-3: this test set contains images of products with different brightness levels from dark to bright to evaluate the mAP@0.50 of the model in different lighting conditions.

●Private-Test-4: this test set contains images of products when part of the product is hidden to evaluate the mAP@0.50 of the model when the image contains part of the product.

●Private-Test-5: this test set contains images that do not have a product or contain products that are similar in shape and some product is overlapping for evaluation of the mAP@0.50 of the model in some special context.

Table II. Testing Dataset

Label	**Number of images**
Private-Test-1	200
Private-Test-2	93
Private-Test-3	102
Private-Test-4	116
Private-Test-5	100

To train the model, we train each YOLO component with the appropriate training dataset and then we train a YOLO component with a mixed 2 training dataset we call it YOLO-Mixed. After that, a comparison of the Multi-YOLO model and with each YOLO component on 5 testing datasets will be accurate. For the YOLO architect, we will use YOLOv5 [9] which was written in Pytorch. We apply transfer learning and data augmentation. The model used for transfer learning is a YOLOv5 model that was trained on many products before. The data augmentation method we used is rotated images at 360 degrees.

For evaluation of the effectiveness of the model, in this study, we will use mAP to measure the accuracy of the detection and calculate execution time to measure the performance of the model. To calculate the performance we use the device as shown in table IV.

Table III. Testing Environment

Environment	Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz
Language	Python 3.7
Library	Pytorch 1.12.1+cpu

The mAP measure is calculated by averaging the AP of the classes in the data set. To calculate the AP of a class in the dataset, we need to calculate the Intersection Over Union (IoU), Precision, Recall, and Precision Recall Curve of that class.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

The AP of a class is the Precision-Recall Curve at an IOU threshold. In this study, we use mAP with an IoU threshold is 0.50 so the measure for accuracy is mAP@0.50.

Table IV. Testing Result for Private-Test-1

Model	**mAP@0.50**	**Time excitation (s)**
YOLO-Normal	0.89	0.75
YOLO-Bad	0.97	0.77
YOLO-Mixed	0.94	0.74
Multi-YOLO-MC	**0.99**	1.51

Model	mAP@0.50	Time excitation (s)
Multi-YOLO-AC	0.88	1.85

Table V. Testing Result For Private-Test-2

Model	mAP@0.50	Time excitation (s)
YOLO-Normal	0.53	0.88
YOLO-Bad	0.77	0.87
YOLO-Mixed	0.49	0.93
Multi-YOLO-MC	**0.84**	1.85
Multi-YOLO-AC	0.75	2.17

Table VI. Testing Result For Private-Test-3

Model	mAP@0.50	Time excitation (s)
YOLO-Normal	0.58	0.98
YOLO-Bad	0.68	0.96
YOLO-Mixed	0.54	0.99
Multi-YOLO-MC	**0.83**	1.97
Multi-YOLO-AC	0.73	2.29

Table VII. Testing Result for Private-Test-4

Model	mAP@0.50	Time excitation (s)
YOLO-Normal	0.80	0.84
YOLO-Bad	0.73	0.89
YOLO-Mixed	0.63	0.87
Multi-YOLO-MC	**0.85**	1.77
Multi-YOLO-AC	0.84	2.82

Table VIII. Testing Result For Private-Test-5

Model	mAP@0.50	Time excitation (s)
YOLO-Normal	0.55	0.92
YOLO-Bad	0.61	0.97

Model	mAP@0.50	Time excitation (s)
YOLO-Mixed	0.39	0.92
Multi-YOLO-MC	**0.76**	1.83
Multi-YOLO-AC	0.69	2.43

The result for the 5 test set is given in Table IV to Table VIII. Based on those results, we can draw some conclusions about the Multi-YOLO as below:

•The result in the Private-Test-1 shows that in normal context all the modes predict well, but the Multi-YOLO gives the best result.

•For the Private-Test-2 when the product has various angles, the mapping result of the 2 single models is reduced and the model mixed gives the worse result. The Multi-YOLO still gives a good result in this case.

•In the Private-Test-3, we want to see how the model inference in different lighting situations. When changing light, the accuracy of the 3 single models is worse but the Multi-YOLO has a good result.

•For the Private-Test-4 we want to test the result of the models when images contain only part of the product. The result in the Multi-YOLO model is better than the 3 single models. • In the Private-Test-5 when the images have difficult contexts the mapping result of the model is hard but the Multi-YOLO model still gives the best result when compared.

In summary, the incorporation of the Max-Confidence algorithm within the Multi-YOLO model enhances accuracy without imposing a substantial reduction in the overall processing performance of the YOLO model. Conversely, introducing data augmentation during training leads to a skewed learning pattern, ultimately resulting in the overall prediction performance of the model being inferior to that of the individual YOLO components. Figure 5. Sample Predict of Multi-YOLO model We applied the Multi-YOLO model for Fire detection via cameras as below diagram and got the feasibility result in the user acceptance testing (UAT) environment:

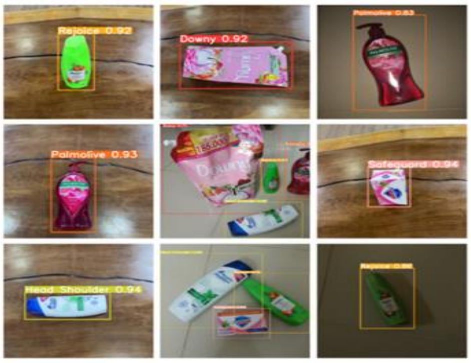

Fig 4. Sample image in applications

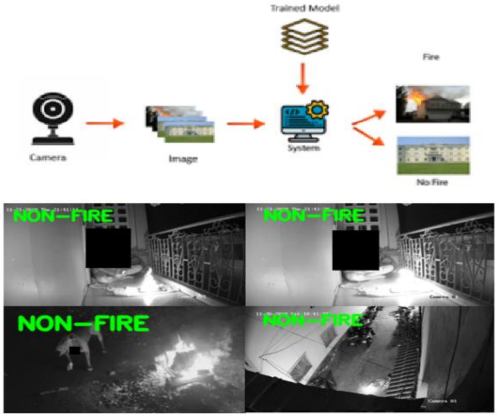

Fig 5. Fire detection system

To conclude, integrating the Max-Confidence algorithm into the Multi-YOLO model proves effective in improving accuracy without causing a significant decline in the overall processing performance of the YOLO model. Conversely, the introduction of data augmentation during training induces a skewed learning pattern, leading to an overall prediction performance that falls short of the capabilities demonstrated by the individual YOLO components.

5. Conclusion

With a focus on product and fire detection, this research's exploration of the Multi-YOLO framework's adaptability opens up possibilities for its application in other industries and image processing jobs. Further investigation into these options may result in significant progress toward automated visual inspection and item detection.

This study not only lays the groundwork for future research into refining and expanding the Multi-YOLO model, but it also presents a novel approach to product and fire detection. Subsequent investigations could focus on optimizing for certain fields and achieving smooth integration with current systems for practical application.

In this research work, we introduced a Multi-YOLO model, termed Fusion rules, which extends the components of the YOLO model. The proposed Multi-YOLO model, designed for product detection, is comprised of two distinct components: YOLO-Normal and YOLO-Bad. The performance of the Multi-YOLO model was evaluated in the detection of product images categorized into 5 labels. The accuracy assessments revealed that the Multi-YOLO - MC model, incorporating fusion rules with maximum confidence, yielded the most favorable results across all 5 testing datasets. This model was further applied successfully in the context of a fire detection application.

Acknowledgements

This research is funded by Ho Chi Minh City University of Education Foundation of Science and Technology under grant number 2022.19.06.

References

[1] LeCun, Y., & Bengio, Y, “Convolutional networks for images, speech, and time series” The Handbook of brain theory and neural networks, 1995.

[2] Girshick, R., Donahue, J., Darrell, T., and Malik, J, “Rich feature hierarchies for accurate object detection and semantic segmentation,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[3] Girshick, R, “Fast r-CNN,” In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448, 2015.

[4] Ren, S., He, K., Girshick, R., & Sun, J, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems, vol. 28, 2015.

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., “You only look once: Unified, real-time object detection,” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2015.

[6] Redmon, J., & Farhadi, A., “YOLO9000: better, faster, stronger,” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271, 2015.

[7] Redmon, J., & Farhadi, A, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

[8] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.

[9] Jocher, G., Nishimura, K., Mineeva, T., and Vilariño, R. YOLOv5. GitHub repository: https://github.com/ultralytics/yolov5. Last accessed on 10/10/2022

[10] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, “Ssd: Single shot multibox detector,” In European conference on computer vision, pp. 21-37. Springer, Cham, 2016.

[11] Sinha, Ankit, Soham Banerjee, and Pratik Chattopadhyay. "An improved deep learning approach for product recognition on racks in retail stores." arXiv preprint arXiv:2202.13081, 2022.

[12] Melek, Ceren Gulra, Elena Battini Sonmez, and Songul Albayrak, “Object detection in shelf images with YOLO,” In IEEE EUROCON 2019-18th International Conference on Smart Technologies, pp. 1-5, 2019.

[13] Majdi, M. A., Dewantara, B. S. B., and Bachtíar, M. M, “Product Stock Management Using Computer Vision”, International Electronics Symposium (IES), pp. 424-429.

[14] Hurtik, Petr, Vojtech Molek, and Pavel Vlasanek, “YOLO-ASC: you only look once and see contours,” International Joint Conference on Neural Networks (IJCNN). IEEE, 2020.

[15] Talaat, Fatma M., and Hanaa ZainEldin. "An improved fire detection approach based on YOLO-

v8 for smart cities." Neural Computing and Applications 35.28 (2023): 20939-20954.

[16] Zhao, H., Jin, J., Liu, Y., Guo, Y., & Shen, Y. (2024). FSDF: A high-performance fire detection framework. Expert Systems with Applications, 238, 121665.

[17] Saleh, A., Zulkifley, M. A., Harun, H. H., Gaudreault, F., Davison, I., & Spraggon, M. (2024). Forest fire surveillance systems: A review of deep learning methods. Heliyon.

[18] Georgiev, A. P. G. AN EVALUATION OF FIRE DETECTION METHODS: COMPARATIVE ANALYSIS AND PERFORMANCE ASSESSMENT 16.

[19] Tao, H. (2024). A label-relevance multi-direction interaction network with enhanced deformable convolution for forest smoke recognition. Expert Systems with Applications, 236, 121383.

[20] Zhou, Y. (2024). A yolo-nl object detector for real-time detection. *Expert Systems with Applications*, *238*, 122256.