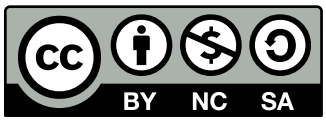
Aalto University
School of Science

Master's programme in Master's Programme in Security and Cloud Computing

Deep Learning based method for Fire Detection

Zixuan Liu

Master's Thesis
2023



Aalto University
School of Science

Author Zixuan Liu

Title Deep Learning based method for Fire Detection

Degree programme Master's Programme in Security and Cloud Computing

Major Security and Cloud Computing

Supervisor Prof. Alexander Jung (Aalto University), Prof. MICHIARDI Pietro (Eurecom)

Advisor Siavash Khajavi

Collaborative partner Detectium

Date 9.2.2023 **Number of pages** 59 **Language** English

Abstract

Fire accidents have become increasingly frequent and have profound effects on today's society, leading to injuries, fatalities, and significant economic losses. It is crucial to develop effective and early fire detection systems that can promptly detect and prevent fire disasters.

Machine learning and computer vision provide a promising solution for the early detection of fires, mitigating potential risks and enhancing safety measures. In this study, we present an extensive and comprehensive fire dataset, surpassing existing datasets in terms of both scale and diversity. This dataset enables robust and thorough training of fire detection models and serves as a benchmark for evaluating future fire detection systems.

The core of our fire detection system is the state-of-the-art Yolov5 model, known for its simplicity, speed, and efficiency in object detection tasks. We demonstrate the effectiveness of our proposed model with promising results, achieving an average F1 score of 0.77 and an mAP@0.5 score of approximately 0.77. These metrics reflect the model's capability to accurately detect fires across various scenarios.

Moreover, we take our research further by focusing on the deployment of the trained model to the cloud. The cloud deployment aspect enhances the practicality and accessibility of our fire detection system, making it more scalable and efficient. Furthermore, it opens up avenues for future advancements and integration with other smart technologies, contributing to the development of smarter and safer environments.

Overall, this work contributes to the advancement of fire detection systems, offering a robust dataset, a powerful detection model, and an efficient cloud deployment approach. With this research, we aim to foster a safer and more secure environment by reducing the risks posed by fire accidents and enabling timely and effective fire prevention measures

Keywords fire detection, yolo, cloud

With the support of the
Erasmus+ Programme
of the European Union

Preface

I want to thank Dr. Siavash Khajavi for giving me this great opportunity to join his company and build this amazing product of fire detection.

I want to thank my mother for helping me collect the fire dataset.

I want to thank Guangkai Jiang for helping me to annotate the dataset.

I want to thank Professor Alexander Jung and Professor MICHIARDI Pietro for their guidance.

Finland, 30 July 2023

Zixuan Liu

Contents

1 Introduction

Fire accidents have a profound impact on individuals, communities, and economies, leading to tragic loss of life, property damage, and financial burden. Several notable fire incidents serve as sobering reminders of the devastating consequences that can result from fires. For instance, the fire disaster at the central hospital in Liaoyuan County, Jilin Province, in 2005 claimed the lives of 37 people and left 95 others injured, causing significant economic losses of approximately 8.22 million yuan [1]. Similarly, the fire outbreak at the Dance King Club in Longgang District of Shenzhen in 2008 resulted in 44 fatalities, 64 injuries, and extensive economic losses of around 70 million yuan [2]. In the United States alone, fire accidents in 2020 led to the tragic loss of nearly 3500 lives, while over 15200 individuals sustained injuries, underscoring the urgent need for effective fire detection and prevention measures [3].

Traditional fire safety systems often rely on smoke detection systems[4] [5], which can detect the presence of smoke particles to trigger alarms. However, these systems have inherent limitations, such as a tendency for false alarms and the inability to provide precise location information for fires. To overcome these challenges and enhance fire detection capabilities, the integration of camera solutions with traditional smoke detectors has gained attention. By leveraging cameras alongside smoke detectors, additional information beyond the presence of smoke can be obtained, including visual confirmation of fire, accurate fire location, and potential insights into the cause of the fire. This combined approach holds the potential for more reliable and accurate fire detection, facilitating prompt response and mitigating the risk of false alarms.

In recent years, computer vision[6] and machine learning algorithms have made significant advancements, offering powerful tools for analyzing visual data and detecting objects in images or video streams. Within the field of computer vision, Convolutional Neural Networks (CNNs) have emerged as a dominant approach for object detection. By training CNN models on large-scale datasets, such as our newly constructed fire dataset, these models can learn to recognize fire patterns and distinguish them from non-fire elements. The application of CNN-based fire detection systems can provide real-time monitoring capabilities, enabling early detection and alerting of fire incidents.

The integration of camera-based solutions, computer vision algorithms, and machine learning techniques represents a promising approach to enhance fire detection systems. By combining the strengths of traditional smoke detectors with the visual information captured by cameras, the accuracy and reliability of fire detection can be significantly improved. Furthermore, the use of sophisticated computer vision algorithms, such as CNNs, enables automated analysis of visual data, facilitating early fire detection, precise fire localization, and potential identification of fire causes. These advancements empower firefighters with valuable information for efficient intervention and enable fire services to gain deeper insights through data analysis, supporting situational awareness and informed decision-making.

In summary, the integration of camera solutions, computer vision algorithms, and machine learning techniques in fire detection systems presents a transformative opportunity to enhance fire safety measures. By leveraging the power of visual data analysis, these systems offer improved accuracy, early detection capabilities, and

valuable insights for effective fire management. Through continuous research and technological advancements, we can strive towards building more robust, efficient, and intelligent fire detection systems to protect lives, safeguard property, and mitigate the devastating impact of fire accidents.

In this research, we present a novel CNN-based fire detection system that demonstrates promising performance. Our work contributes to the field of fire detection in the following key ways:

1. Large-scale Fire Dataset: We introduce a comprehensive and extensive fire dataset, surpassing existing datasets in terms of scale and diversity. Our dataset comprises over 10,000 real-world images and 1,000 real-world videos of fire. It is the largest fire dataset to date, containing more than three times the number of images compared to the previously largest dataset. This dataset enables more robust and comprehensive training of fire detection models.

2. YOLOv5 Model Training: We employ the state-of-the-art YOLOv5 model as the backbone for our fire detection system. Leveraging the large-scale dataset, we train the YOLOv5 model to effectively detect fire in various scenarios and environments. The trained model demonstrates promising results in terms of accuracy, precision, and recall, showcasing its effectiveness in fire detection tasks.

3. Cloud Deployment and IoT Integration: We take a step further by deploying the trained model to the cloud, allowing for seamless integration with Internet of Things (IoT) devices. This deployment approach enhances the scalability and accessibility of our fire detection system. By leveraging the power of cloud computing and IoT connectivity, our system can be easily deployed and utilized in real-world scenarios, providing real-time fire detection capabilities.

Overall, our research contributes a large-scale fire dataset, a well-trained YOLOv5 model, and a cloud-based deployment framework for fire detection. These contributions advance the field of fire detection, offering improved performance, scalability, and practicality for real-world applications.

The remaining section of the thesis is structured as follows:

1. Literature Review: This section provides a comprehensive review of existing fire detection systems in the literature. We explore various approaches, methodologies, and techniques employed in previous studies to address the challenge of fire detection.

2. Methodology: In this section, we present the methodology employed in our research. We provide detailed information about the construction of our collected dataset, including the acquisition process and the types of fire images and videos included. Additionally, we describe the yolov5 models used in our experiments and discuss their performance evaluation. Furthermore, we delve into the deployment and integration of the trained model, highlighting the practical aspects of implementing the fire detection system.

3. Conclusion: The final section of the thesis comprises the concluding remarks and the overall contributions of the research. We summarize the findings, discuss the implications of the study, and suggest potential avenues for future research in the field of fire detection. Additionally, we reflect on the strengths and limitations of our approach and highlight the practical implications and potential applications of our developed fire detection system.

2 Literature review

The pursuit of developing an efficient fire detection system has been an ongoing endeavor since the early stages of research. Initially, researchers relied on handcrafted technologies that focused on identifying specific features unique to fire, such as its characteristic motion and distinct red color.

Phillips et al. [7] presented a novel fire detection system that harnessed the power of motion and color information extracted from video clips to effectively detect and locate fire incidents. To achieve this, they initially developed a Gaussian-smoothed color histogram, which proved instrumental in detecting fire pixels within the video frames [7]. Subsequently, they introduced a temporal variation analysis of the detected possible fire pixels to discern the true fire pixels from the false detections [7].

A major challenge in fire detection lies in dealing with the reflections of fire on objects in close proximity to the fire source. To address this issue and enhance fire detection accuracy, Phillips et al. employed two essential techniques: erosion and region growing. The erosion operation facilitated the automatic removal of spurious fire pixels, which are erroneous detections, thus enhancing the reliability of the system [7]. On the other hand, the region growing method efficiently identified missing fire pixels, thereby ensuring more comprehensive and accurate fire detection [7].

Notably, the experiments conducted by Phillips et al. demonstrated the robustness and versatility of their fire detection system. Even under conditions with limited information, their system could intelligently and automatically determine the presence of fire, highlighting the effectiveness of their approach in various real-world scenarios [7].

By incorporating motion and color information and employing sophisticated algorithms such as Gaussian-smoothed color histograms, erosion, and region growing, the fire detection system proposed by Phillips et al. exhibited notable advancements in fire detection technology.

Another pioneering effort in this direction was made by Thou-Ho et al. [8], who proposed an early fire alarm system based on video processing. In their innovative approach, Thou-Ho et al. leveraged the power of video analysis to extract crucial fire pixels and smoke pixels from input videos. This process involved meticulously measuring the chromaticity and disorder within the RGB model. Specifically, fire pixels were deduced based on the intensity and saturation of the R (red) component [8]. Subsequently, the extracted fire pixels underwent a rigorous verification process, where the dynamics of growth and disorder were monitored throughout the entire video. Additionally, the presence of smoke was also taken into account during this validation step.

The hallmark of their methodology lay in the iterative checking of the growth patterns of the fire, which served as a crucial trigger for the alarm system. Only when all the verification conditions were met did the alarm system become activated, ensuring a low false alarm rate and reliable fire detection. The experiments conducted by Thou-Ho et al[8]. showcased the feasibility of achieving fully automated surveillance of fire accidents, establishing a robust foundation for early fire detection.

In contrast to Thou-Ho et al. [8], Celik et al. [9] pursued a distinct approach

to fire and smoke detection by analyzing the entire fire and smoke datasets. Their investigations revealed that the YCbCr color space proved more effective in segregating illumination information from chrominance when compared to the traditional RGB color space. As a result, the predefined rules designed to detect potential fire pixels in RGB color space were transformed into the YCbCr color space to yield more accurate and reliable outcomes [9].

A noteworthy advantage of this transformation lies in the derivation of a single quantitative measure that indicates the probability of a given pixel being classified as a fire pixel. By encoding the implicit fuzziness or uncertainties associated with the rules obtained from repeated experiments and the impreciseness of the decision variable, Celik et al. adopted a fuzzy representation, allowing them to express the output decision in linguistic terms [9]. This linguistic representation further provided a means to interpret the output decision quantity, ranging from zero to one, as the likelihood that a pixel is either a fire pixel or not [9].

The proposed fire detection model showcased remarkable performance, achieving an impressive 99% correct fire detection rate while maintaining a low false alarm rate of 4.5%[9]. Such an accurate and reliable fire detection model proved to be an invaluable asset and could potentially be employed as a pre-processing stage for more comprehensive fire detection systems. Celik et al.'s innovative work has contributed significantly to the development of advanced fire detection techniques, particularly by exploiting the YCbCr color space and leveraging fuzzy representation to address the challenges associated with fire and smoke pixel detection.

Toreyin et al. [10] presented an innovative approach for fire and flame detection, surpassing conventional methods by incorporating a diverse range of features. In addition to color and motion information, their method involved the detection of the flicker process using a Markov model [10]. By integrating spatial color variations in the moving regions within the same Markov models, the authors achieved a comprehensive analysis to make a final decision [10].

The incorporation of the flicker process in the detection strategy proved to be a key factor in their method's success. This aspect allowed the system to differentiate genuine fire and flame incidents from ordinary motion of flame-colored moving objects, effectively reducing false alarms [10]. Such a robust approach has profound implications for real-time fire detection, where accurate and timely identification of fire events is critical for ensuring prompt and effective response measures.

By employing a Markov model that integrates color, motion, and flicker information, Toreyin et al. succeeded in creating a sophisticated fire and flame detection system that outperforms traditional methods. The utilization of multiple features in the analysis process significantly enhanced the system's ability to discern genuine fire incidents, making it more reliable and suitable for practical applications in real-world scenarios.

Rinsurongkawong et al. [11] presented a novel approach for early fire detection using the Lucas-Kanade optical flow algorithm, providing real-time fire detection capabilities for both indoor and outdoor environments. The most significant advantage of their method lies in its ability to detect fire at the initial stages of the burning process, offering the potential for earlier response and mitigation compared to other methods [11].

The foundation of their approach lies in the Lucas-Kanade optical flow algorithm, which enables the detection of fire in video streams by identifying moving pixels. The authors further refined their method by applying a background subtraction algorithm to distinguish fire-like colors from the moving regions in the video stream [11]. Additionally, they employed growth rate analysis to examine fire contours and used the Lucas-Kanade optical flow pyramid to analyze the motion of optical flow features in the identified fire regions. The combination of growth rate and optical flow analysis significantly enhanced the accuracy of fire detection and reduced false alarms caused by fire-like colors [11].

While the proposed method demonstrated impressive capabilities in early fire detection, the authors acknowledged its limitations when dealing with images containing fire-like objects in the background. This highlights the importance of further research and refinement to address such challenges and enhance the system's performance in complex and dynamic environments.

In summary, Rinsurongkawong et al.'s work showcased a promising approach for early fire detection using the Lucas-Kanade optical flow algorithm. Their method's capability to detect fire at its inception and the reduction of false alarms make it a valuable contribution to fire safety technology. However, addressing background complexities and improving the system's robustness in various scenarios remain important areas for future research.

To address the challenges in fire motion detection, [12] introduced a set of motion features based on motion estimators. Their approach focused on distinguishing the unique characteristics of turbulent and fast fire motion from the structured and rigid motion of other objects [12]. By considering the absence of smoothness and intensity constancy in fire motion, they carefully designed two optical flow methods tailored specifically for fire detection tasks [12].

The first optical flow method aimed to model fire with dynamic texture, utilizing optimal mass transport techniques [12]. This method effectively captured the dynamic and irregular nature of fire motion, which is distinct from the motion of other objects in the scene. The second optical flow method was specifically designed to model saturated flames, employing a data-driven scheme [12].

Having obtained the optical flow data, the authors computed various features related to flow magnitudes and directions, enabling them to discriminate between fire and non-fire motion effectively [12]. These motion features provided valuable insights into the unique patterns associated with fire, facilitating accurate fire detection.

To evaluate the effectiveness of their proposed methods, [12] conducted thorough testing and demonstrated the practical usefulness of their approach. Additionally, they introduced a novel evaluation method that allowed for a controlled environment to analyze the influence of different parameters on fire detection [12]. This evaluation framework provided valuable insights into the performance and robustness of their methods under varying conditions.

In conclusion, [12] presented a comprehensive approach for fire motion detection, leveraging specially designed optical flow methods and motion features. Their work successfully addressed the limitations of previous methods and showcased practical utility in accurately detecting fire motion. The introduction of a controlled evaluation

environment further strengthened the credibility of their findings and laid the foundation for future advancements in fire detection research.

Mobin et al. [13] presented a pioneering and intelligent self-controlled smart fire detection system called Safe From Fire (SFF). This comprehensive system utilized multiple sensors and actuators, operating under the supervision of a micro-controller unit (MCU). By strategically placing various sensors throughout the monitored area, the SFF system could effectively capture critical data points related to fire occurrences. Leveraging the integrated fuzzy logic approach, the system analyzed the input signals from these sensors to accurately assess the severity of the detected fire incidents.

A key aspect of the SFF system was its utilization of data fusion algorithms, which played a crucial role in reducing false alarms caused by deceptive fire situations, such as smoking or other non-fire-related activities. By integrating and cross-verifying data from multiple sensors, the SFF system demonstrated a higher level of accuracy in distinguishing genuine fire events from potential false alarms.

During a fire event, the SFF system efficiently provided a range of essential fire services. It could promptly send text messages or make telephone calls to alert people and relevant authorities about the detected fire. Furthermore, the system activated a fire alarm, effectively announcing the precise location of the fire and its severity level. These timely and automated response mechanisms enhanced the overall effectiveness and reliability of the SFF system in mitigating fire-related risks.

To validate the performance and reliability of the SFF system, the researchers conducted real-time fire scene simulations. Through rigorous evaluations, the SFF system demonstrated its efficiency in detecting fire incidents accurately and responding swiftly with appropriate actions. The successful outcomes of the evaluations showcased the potential practical applicability of the SFF system in real-world fire detection scenarios.

The early handcrafted technologies in fire detection, while serving as important early attempts, came with several inherent limitations and disadvantages that hindered their effectiveness in real-world applications. One major drawback was their limited adaptability to diverse fire scenarios. These methods heavily relied on predefined rules and heuristics, making them less flexible in handling new and evolving fire patterns that might not fit within the predefined criteria. Moreover, handcrafted approaches often had difficulty in effectively extracting and representing complex fire-related features from the input data, leading to suboptimal performance in distinguishing fire from non-fire elements.

Another significant limitation was the challenge of scaling these handcrafted techniques to handle large datasets and real-time processing requirements. As fire datasets grew in size and complexity, handcrafted methods struggled to cope with the increased computational demands, leading to performance degradation and reduced efficiency. Additionally, handcrafted techniques were often sensitive to noise and variations in the input data, making them less robust in handling diverse environmental conditions and camera perspectives.

Furthermore, handcrafted fire detection systems heavily relied on domain-specific expertise for manual feature engineering and fine-tuning, which made them highly dependent on the knowledge and experience of the developers. This dependency on

expert knowledge limited their applicability to non-expert users and increased the complexity of maintaining and updating the systems as fire patterns evolved.

In contrast, as the field of computer vision and machine learning progressed, researchers recognized the potential of data-driven approaches, particularly Convolutional Neural Networks (CNNs), which offered the capability to automatically learn and extract complex features from vast amounts of data. The transition from handcrafted methods to data-driven CNNs revolutionized fire detection by addressing many of the limitations of earlier techniques. By leveraging large-scale fire datasets, researchers could train CNNs to recognize fire-related features with high precision. These datasets encompassed various fire scenarios, including different types of fires, lighting conditions, and environments. Consequently, the trained models achieved remarkable accuracy in detecting fire instances, even in challenging and diverse real-world settings.

The ability of CNNs to generalize well to new and unseen fire scenarios further enhanced their practical applicability. Unlike traditional handcrafted methods, which often struggled with novel fire patterns, CNN-based models demonstrated robustness and adaptability. This newfound capability made them suitable for deployment in various contexts, ranging from indoor fire detection systems to outdoor surveillance and forest fire monitoring.

Moreover, the advancements in deep learning techniques paved the way for real-time fire detection. The speed and efficiency of CNNs allowed for quick analysis of video streams, enabling timely responses to fire incidents. This real-time capability was crucial for early fire detection, facilitating prompt intervention and mitigating potential disasters.

In response to the growing demand for more robust and accurate fire detection systems, researchers have explored the use of deep-learning techniques. Among them, Zhang et al. [14] introduced a novel approach to fire detection by training both a full image and fine-grained patch classifier within a joint deep convolutional neural network (CNN). The detection process is carried out in a cascaded fashion, where the full input image undergoes evaluation by the global image classifier. If a fire is detected at this stage, the input image is then forwarded to the fine-grained patch classifier for further analysis, providing precise localization of fire patches [14].

In their work, the global image classifier is fine-tuned from the well-known 8-layered AlexNet [15], readily available in the Caffe framework, and pre-trained on the ImageNet dataset. Meanwhile, the fine-grained patch classifier is trained using upsampled Pool-5 features, creating a 2-layered fully connected neural network tailored for fire detection [14]. One key advantage of their proposed architecture is the avoidance of redundant feature computation, as both the global and fine-grained classifiers share the same network. This design ensures an efficient utilization of computational resources and enables faster inference during fire detection tasks.

The performance evaluation of their fire patch detector showcased impressive results, achieving 97% and 90% accuracy on the training and testing datasets, respectively. The high accuracy of their model highlights its potential for real-world fire detection scenarios. Moreover, the precise localization of fire patches using the fine-grained patch classifier provides valuable information for emergency response

teams, facilitating prompt actions to mitigate the fire's impact.

By leveraging the power of deep learning and the cascaded approach in their architecture, Zhang et al. [14] have contributed to the advancement of fire detection systems, offering a promising solution to address the challenges associated with fire detection in complex environments.

However, it should be noted that the fire detection model proposed by Zhang et al. [14] is evaluated on balanced datasets, which might not fully capture the complexities of real-world scenarios where fire incidents are rare. Consequently, the performance of their method could be hindered when applied to more realistic datasets where fire images are scarce.

In response to the limitations of existing fire detection models, Sharma et al. [16] take a different approach and propose the use of deeper Convolutional Neural Networks (CNNs) along with the incorporation of an additional fully connected layer to fine-tune the networks for fire detection. Two widely used pre-trained CNN models, VGG16 [17] and Resnet50 [18], are employed in the development of their fire detection system.

The core innovation of Sharma et al.'s method lies in its ability to tackle imbalanced datasets, which more closely resemble real-world scenarios where fire occurrences are relatively rare compared to non-fire instances. By testing their proposed model on a deliberately unbalanced dataset, where non-fire images substantially outnumber fire images, they demonstrate the model's robustness in handling skewed class distributions. Their approach surpasses the challenges of imbalanced data and achieves a notably enhanced performance on real-world datasets, providing more reliable fire detection capabilities.

The incorporation of additional fully connected layers in their CNN models plays a pivotal role in improving the model's performance. The extra layer facilitates fine-tuning of the networks specifically for fire detection tasks, optimizing the model to learn and discern intricate features indicative of fire in images. Notably, the experimentation results show that Resnet50 slightly outperforms VGG16, reaffirming the advantage of deeper CNN architectures in capturing complex patterns and representations within the data.

The contributions made by Sharma et al. [16] marks a significant step forward in fire detection research, as their model addresses the challenges posed by imbalanced datasets, and exhibits higher adaptability and accuracy in detecting fire incidents, even in real-world scenarios. Their findings underscore the importance of leveraging advanced CNN architectures and specialized fine-tuning techniques for improved fire detection performance, paving the way for more reliable and effective fire safety systems in various applications.

Muhammad et al. have made significant strides in deep learning-based fire detection by fine-tuning four different CNN architectures, namely AlexNet [19], SqueezeNet [20], GoogleNet [21], and MobileNetV2 [22]. Their comprehensive exploration of various CNN models, which encompass some of the most renowned Convolutional Neural Network architectures, has shown promising results in flame detection. Despite these advancements, a key challenge that remains is the relatively high number of false alarms in their fire detection system.

Notably, while previous research has demonstrated impressive performance in

fire detection, certain limitations still persist in their methodologies. One prominent drawback is the evaluation of their methods on limited datasets, characterized by a scarcity of fire images and a restricted diversity of fire types. Consequently, the generalizability and effectiveness of their fire detection models in real-world environments may not be as promising as demonstrated within the limited dataset setting.

Moreover, the employed architecture in their fire detection models might not be optimally suited for the specific task of fire detection. Most of the previous models have utilized two-stage large Convolutional Neural Networks, which are designed for multiple object detection. However, in the context of fire detection, the goal is to detect a single class, which is the presence of fire. Large and complex neural networks might not be the most suitable choice for this particular single-class detection task. Consequently, the use of such architectures may introduce unnecessary computational overhead and might not effectively capture the discriminative features required for precise and efficient fire detection.

To address these limitations, there is a need for further research that focuses on evaluating fire detection models on more diverse and extensive datasets, resembling real-world fire scenarios. Additionally, exploring the design of more specialized architectures tailored specifically for single-class fire detection could lead to enhanced accuracy and a reduction in false alarms, ultimately contributing to the development of more reliable and robust fire detection systems.

3 Method

3.1 Dataset

3.1.1 Previous datasets

In this section, we provide an overview of existing fire datasets commonly used in fire detection research. Understanding the characteristics and limitations of these datasets is crucial for developing effective fire detection models.

The FireNET dataset [23] consists of a total of 502 images, which are further split into 412 images for training and 90 images for validation. Although the dataset provides bounding box annotations in Pascal VOC XML format, it is important to note that most of the images in this dataset are relatively small, with an average size of 275×183 pixels. This small image size limits the availability of textural features for the model to learn from, which can affect the model's performance on larger fire instances. Figure 1 demonstrates the performance of the model on normal cooking fire images, which typically exhibit a large and distinct fire presence in the image. However, the accuracy of correctly detecting these instances as fire is observed to be only 50

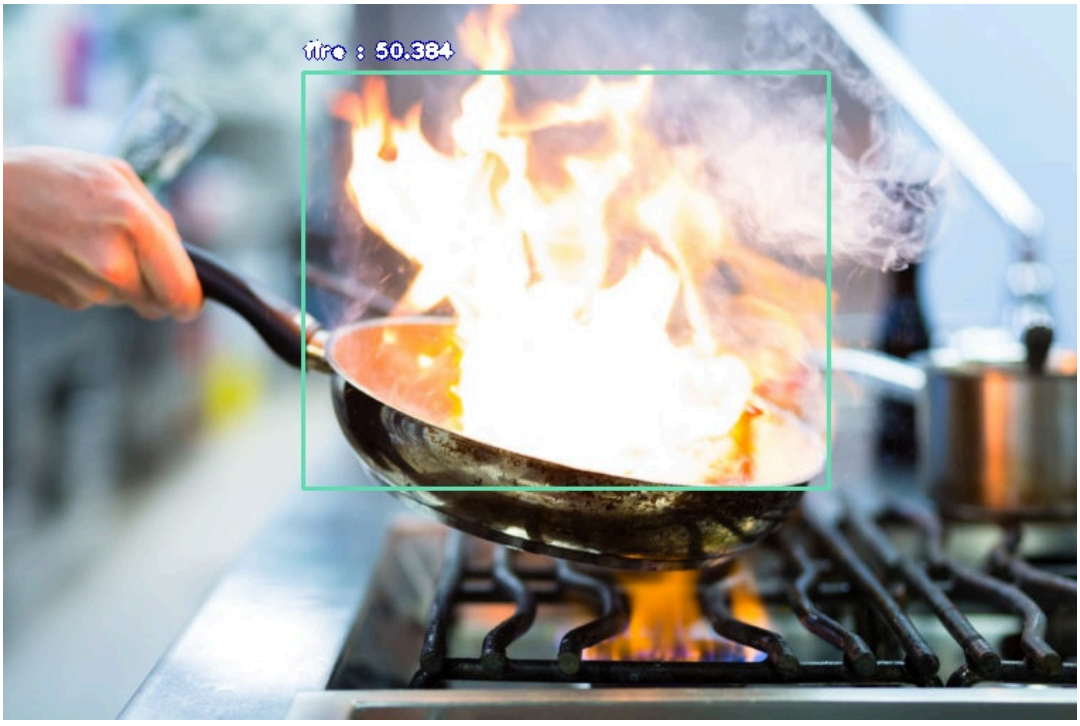


Figure 1: The performance of the model on normal cooking fire images in FireNET dataset.

The Fire Detection from CCTV on Kaggle dataset [24] primarily focuses on the binary classification task of fire versus no fire. The dataset consists of images extracted from 3-4 videos, resulting in a high level of similarity among the images. However,

this dataset does not provide bounding box annotations, making it less suitable for tasks requiring precise object localization.

The Fire-Detection-Image-Dataset [25] includes a diverse collection of images that capture both fire and non-fire scenarios. This dataset encompasses a wide range of fire situations, including variations in intensity, luminosity, size, and environmental conditions. However, it is worth noting that this dataset is relatively small and highly unbalanced, which may limit its ability to accurately represent real-world fire detection scenarios. Additionally, the absence of bounding box annotations restricts its utility for tasks requiring precise object localization.

The FIRE Dataset on Kaggle [26] was created during the NASA Space Apps Challenge in 2018. This dataset comprises 755 outdoor fire images and 244 non-fire images, primarily intended for classification tasks. Similar to the previously mentioned datasets, the FIRE Dataset does not provide bounding box annotations.

The Wildfire Smoke Dataset [27], released by AI for Mankind in collaboration with HPWREN, is a collection of 737 images with bounding box annotations. This dataset specifically focuses on the detection of wildfire smoke, which is crucial for monitoring and early warning systems in fire-prone areas.

The fire-and-smoke-dataset on Kaggle [28], collected by DataCluster Labs, offers an extensive and challenging set of over 7000 original fire and smoke images. These images were captured and crowdsourced from over 400 urban and rural areas in India and have been carefully reviewed and verified by computer vision professionals at DataCluster Labs. However, it is important to note that this dataset is not publicly accessible, and its image quality and diversity may vary.

The Domestic-Fire-and-Smoke-Dataset [29] consists of early fire and smoke images captured in real-world scenarios using mobile phones. The dataset encompasses a wide variety of lighting conditions, including indoor and outdoor scenes, as well as different weather conditions. It is well-suited for tasks such as early fire and smoke detection. The dataset provides bounding box annotations in multiple formats, including COCO, Pascal VOC, and YOLO formats, facilitating compatibility with different object detection frameworks. Figures 2 and 3 showcase a selection of sample images representing both indoor and outdoor scenes.

The Wildfire-Detection dataset [30] comprises images of size 250×250 pixels, depicting both normal scenes and fire instances. However, similar to the FireNET dataset, the images in this dataset are relatively small, which may limit their ability to effectively detect smaller fire flames, such as those from candles or lighters. Some examples of these images are shown in Figure 4.

The DFireDataset [31] is currently the largest publicly available dataset for fire detection, consisting of over 21,000 images. The dataset includes images with only fire, images with both fire and smoke, and images without fire. All images in the DFireDataset have been annotated according to the YOLO format, resulting in a total of 14,692 fire bounding boxes. However, it is important to note that a significant portion of the dataset contains images captured from videos, which leads to highly similar and repetitive images as shown in Figure 5. After removing duplicate images, approximately half of the original dataset remains.

Overall, the currently available public datasets for fire detection exhibit limitations

Figure 2: Some sample images for indoor fire scenes in Domestic-Fire-and-Smoke-Dataset.

Figure 3: Some sample images for outdoor fire scenes in Domestic-Fire-and-Smoke-Dataset.

in terms of the number of fire images and variations in fire types. Additionally, some datasets may focus on larger fire instances, while smaller fire flames may be underrepresented. Understanding these limitations is essential for designing effective fire detection models that can handle a wide range of fire scenarios.

As for the fire video dataset, the available options are limited. The mivia Fire Detection Dataset [32] is one such dataset that can be utilized. This dataset consists of 31 videos captured in real environments. It can be divided into two main parts: the first 14 videos showcase scenarios with fire incidents, while the remaining 17 videos do not contain any fire-related events. However, the latter part includes challenging situations that may resemble fire, such as moving red objects, smoke, or clouds.

Despite its availability, it is important to note that the mivia Fire Detection Dataset is relatively small in size, which can pose limitations when used for training purposes.

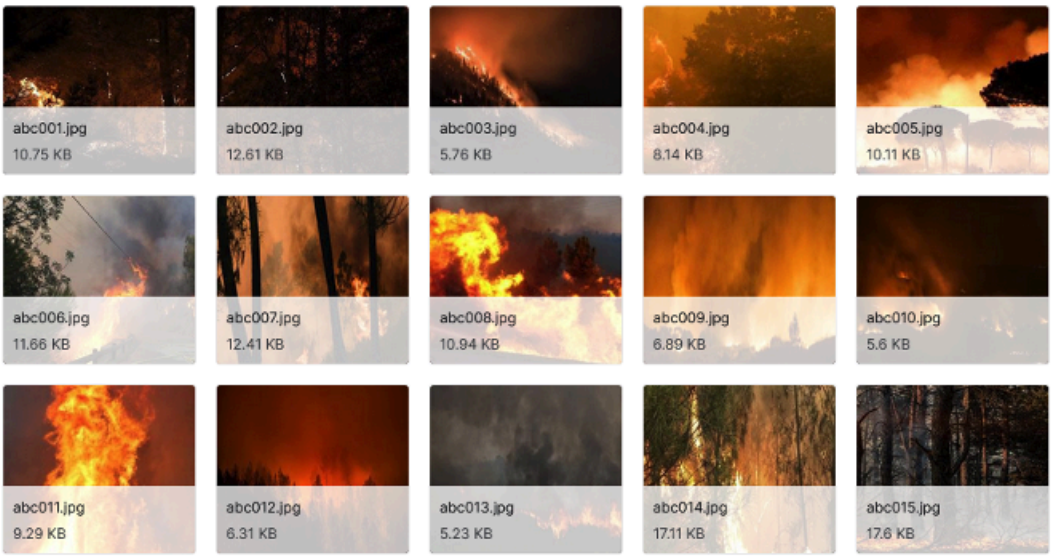


Figure 4: Some examples of fire images in Wildfire-Detection dataset.

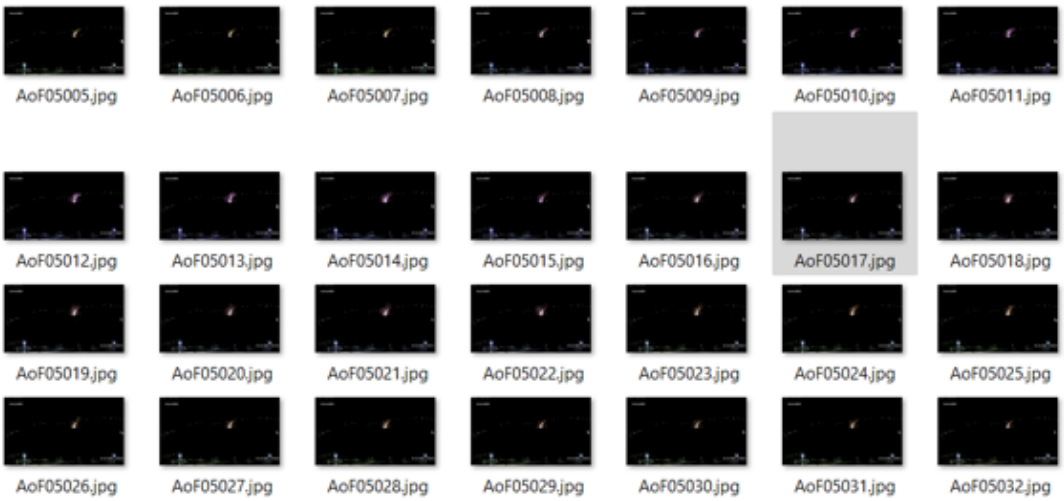


Figure 5: Highly similar fire images in DFireDataset.

Additionally, the quality of the dataset may not be optimal, making it challenging for the model to effectively learn from the provided samples.

3.1.2 Our datasets

Due to the limitations and inadequacies of the existing fire datasets, we recognized the need to construct a new, large-scale dataset tailored specifically for training our fire detection model. Our goal was to create a diverse and comprehensive dataset that encompasses a wide range of fire scenarios and variations. This dataset consists of a total of 10,210 fire images and 1,076 fire videos, offering a substantial and diverse collection of fire-related data.

To curate the image dataset, we employed a systematic approach, searching for relevant images across multiple platforms. Our search efforts extended to popular platforms such as Adobe Stock and Pexels, where we utilized various text search queries to retrieve images related to fire incidents. To ensure the dataset's comprehensiveness, we conducted searches in different languages, including English, Chinese, and French. Additionally, we expanded our search to include videos and short clips from platforms like YouTube and TikTok, extracting frames from these videos to augment our image dataset. Through this rigorous search and extraction process, we gathered a significant number of high-quality fire images, representing diverse fire scenarios and contexts.

During the collection process, we applied strict criteria to ensure the quality and authenticity of the dataset. We carefully reviewed each image, eliminating any that had been manually edited or contained artificially generated fire. Furthermore, we excluded images that raised ethical concerns or lacked sufficient clarity to provide meaningful data. The resulting image dataset comprises a total of 10,210 fire images, representing various fire types and environmental conditions. To enhance the dataset's diversity, we also included 8,983 non-fire images to provide a comprehensive context for fire detection.

Figure 6 showcases a selection of images from our collected dataset, offering a glimpse into the diversity and richness of the fire-related content we have assembled.

Figure 6: One example of our image dataset.

To ensure accurate annotations for the dataset, we employed Roboflow, a powerful annotation tool that significantly simplifies and expedites the annotation process. With Roboflow, we leveraged pre-trained models to automatically apply bounding box annotations to the fire images. This AI-assisted labeling process greatly reduced the manual effort required for annotation, while still ensuring high accuracy and consistency in the annotations. The annotations provided by Roboflow were thoroughly reviewed and verified to guarantee their quality and relevance.

Figure 7 showcases an example of our annotated images, with precise bounding box annotations encapsulating the fire objects within the images. These annotations enable the training and evaluation of fire detection models, facilitating the development of robust and accurate fire detection algorithms.

Figure 7: One example of annotated images with bounding boxes.

While annotating the video dataset, we focused on video-level labels to determine whether a particular video contained instances of fire or not. This information was crucial for training and evaluating fire detection models on video data. However, to provide a more detailed understanding of temporal annotations, we manually reviewed and edited the videos. This involved identifying the precise frames corresponding to the start and end of fire incidents within each video. This temporal annotation enriches the dataset by providing precise temporal information about fire occurrences in the videos.

Creating and finalizing the annotations for our dataset was an extensive and meticulous process that spanned several months. We ensured that the annotations were accurate, comprehensive, and consistent throughout the dataset, enabling the development of robust fire detection models.

In summary, our newly constructed dataset encompasses a vast collection of fire images and videos, carefully curated to capture various fire scenarios and types. This dataset, along with the precise annotations, provides a valuable resource for training, testing, and advancing fire detection algorithms.

In the context of object detection, bounding boxes are commonly used to represent objects by drawing rectangles around them. While bounding boxes provide a general indication of an object's location, they do not capture the exact shape of the object.

Additionally, bounding boxes may include parts of the background or other objects within their boundaries. This is especially true in fire datasets, where distinguishing objects from their surroundings can be challenging.

Segmentation masks, on the other hand, offer a more detailed representation of objects by outlining their precise shapes. These masks provide a pixel-level analysis, allowing for a more accurate understanding of an object's shape, size, and position. Given the variable and dynamic nature of fire, precise identification and delineation are essential for comprehensive analysis.

To facilitate pixel-level analysis and enhance the understanding of fire, we have prepared a small segmentation dataset consisting of nearly 1000 fire images. This dataset includes segmentation masks that outline the boundaries of fire regions, enabling a detailed and accurate assessment of fire characteristics. Figure 8 showcases some examples from our segmentation dataset, highlighting the pixel-level details and the ability to precisely identify and analyze fire regions.

By incorporating segmentation masks, we go beyond the limitations of bounding boxes and achieve a more comprehensive understanding of fire phenomena. This level of detail is crucial in scenarios where precise identification and delineation of fire regions are required, enabling advanced analysis, monitoring, and decision-making in fire-related applications.

During the annotation process, we leveraged Meta AI's Segment Anything Model (SAM) [34] to streamline the generation of segmentation masks. SAM is a powerful and versatile model specifically designed for image segmentation tasks. It excels in producing high-quality object masks for various objects within an image, either through direct input of the entire image or via input prompts such as bounding boxes [34]. Trained on an extensive dataset consisting of 11 million images and 1.1 billion masks, SAM exhibits strong zero-shot performance across a wide range of segmentation tasks.

By utilizing SAM, we were able to convert our existing bounding box datasets into a comprehensive segmentation dataset. Initially, the bounding boxes served as a starting point for the segmentation process. SAM takes these bounding box annotations as input and generates precise segmentation masks for each bounding box, accurately outlining the object boundaries within the image. This transformation allowed us to enhance the level of detail and accuracy in our annotation data, providing a more comprehensive understanding of fire regions.

While the initial segmentation masks were generated by SAM, it is important to note that further adjustments and refinements were necessary. Some masks may have contained inaccuracies or required fine-tuning. To ensure the accuracy and quality of the annotations, we performed careful data cleaning and refinement procedures. This involved addressing issues such as removing duplicate detections, merging overlapping polygons, and splitting polygons that covered multiple objects. Through meticulous data cleaning, we achieved a final set of accurate and reliable annotations.

For the annotation refinement process, we utilized the Roboflow [33] platform, which offers convenient tools for editing and validating the data. Roboflow facilitated efficient annotation refinement, allowing us to easily make adjustments, validate the annotations, and ensure their accuracy. The refined annotations were then saved in the yolov8 segmentation format, enabling their seamless integration into the training

pipeline for future use.

The combined utilization of SAM and Roboflow provided an efficient and effective workflow for generating and refining segmentation annotations. This approach enabled us to leverage the strengths of advanced segmentation models while ensuring the accuracy and reliability of the final annotation data.

Figure 8: Examples of our segmentation image dataset.

The division of our dataset into three distinct parts, namely the training set, validation set, and testing set, plays a crucial role in the development and evaluation of our fire detection model. This division allows us to effectively train, fine-tune, and assess the performance of the model using independent and representative subsets of the data.

The training set comprises 70% of the total images and videos in our dataset. This subset is used to train the model by exposing it to a diverse range of fire-related images and videos, enabling it to learn and extract meaningful features and patterns associated with fire occurrences. During the training process, the model optimizes its parameters, adjusting its internal representations to better align with the characteristics of the fire data. The larger size of the training set ensures that the model receives ample exposure to different fire scenarios, enhancing its ability to generalize and make accurate predictions on unseen data.

The validation set, which consists of 10% of the images and videos, serves as a critical component for model evaluation and hyperparameter tuning. After each training iteration or epoch, the model's performance is assessed on the validation set to monitor its progress and identify potential issues such as overfitting or underfitting. By evaluating the model's performance on a separate subset of the data that it has not

been trained on, we can gauge its ability to generalize and make accurate predictions on new, unseen fire images and videos. This evaluation guides us in fine-tuning the model's hyperparameters, such as learning rates or regularization techniques, to optimize its performance and ensure its robustness.

The testing set, representing 20% of the images and videos, serves as the final benchmark for assessing the model's performance. This independent subset of the data is used to evaluate the model's ability to generalize to new, unseen fire scenarios and make accurate predictions. By withholding this portion of the data during the training and validation phases, we obtain an unbiased measure of the model's performance in real-world scenarios. The testing set enables us to gauge the model's effectiveness in detecting fire across a range of challenging scenarios and provides insights into its overall accuracy and reliability.

The division of the dataset into training, validation, and testing sets is essential for several reasons. Firstly, it allows us to objectively evaluate the model's performance by assessing its ability to generalize to unseen data. Secondly, it helps us avoid overfitting, a phenomenon where the model becomes too specialized in the training data and fails to generalize to new examples. By evaluating the model on an independent validation set, we can detect and address overfitting issues. Finally, the testing set provides an unbiased estimate of the model's real-world performance and its generalization capabilities, helping us determine its readiness for deployment in practical applications.

By carefully partitioning our dataset into these three distinct subsets and following rigorous evaluation protocols, we can develop and assess a fire detection model that demonstrates robustness, accuracy, and generalization abilities across various fire scenarios and data distributions.

3.1.3 Data argumentation

Data augmentation plays a crucial role in improving the performance of object detection models. In this work, we employ online image space and color space augmentations during the loading of the training dataset. However, it is important to note that the validation and test datasets are not subjected to data augmentation. Additionally, a novel and unique data augmentation technique called Mosaic augmentation is utilized for each training image, further enhancing the dataset's diversity.

The Mosaic augmentation technique involves the combination of four randomly selected images from the training dataset into a single augmented image consisting of four tiles. To achieve this, the selected images are resized using different ratios to seamlessly integrate them into four grids. Subsequently, a random image patch is cropped from the center, forming the final augmented image. Figure 9 visually presents some examples of the mosaic augmented training images in the context of YOLOv5.

This approach ensures that the images are never encountered in the same way during training, contributing to a more comprehensive learning experience for the model. By exposing the model to various combinations of images, the Mosaic augmentation helps in training the model to detect objects across different localizations and environments. Moreover, it proves particularly beneficial in addressing the common challenge of

accurately detecting small objects. By incorporating diverse image compositions through the Mosaic augmentation, the model learns to overcome the limitations associated with smaller objects and improves its detection capabilities across the entire object size spectrum.

The utilization of Mosaic augmentation, alongside online image space and color space augmentations, adds an essential level of variability and realism to the training data. This augmentation strategy not only enhances the model's ability to generalize to new and unseen scenarios but also fosters robustness and adaptability in object detection.

Figure 9: Examples of mosaic augmented training images.

3.2 Model

The current work utilizes the yolov5 [35] object detection algorithm. Yolov5 belongs to the renowned You Only Look Once (YOLO) [36] family of computer vision

models and is widely recognized for its object detection capabilities. What sets yolov5 apart from other object detection frameworks is its exceptional ease of use, making it highly accessible and convenient for developers aiming to integrate computer vision technologies into their applications. Its user-friendly interface and simplified implementation process enable developers to efficiently leverage the power of object detection for their specific use cases. By utilizing yolov5, the present work benefits from its efficiency, accuracy, and developer-friendly nature, thereby enhancing the overall performance and effectiveness of the implemented computer vision system.

3.2.1 The overall architecture of yolov5

The architecture of yolov5 builds upon the foundational idea of previous YOLO versions and consists of three essential components: the backbone, feature fusion network (neck), and prediction head [37], as illustrated in Figure 10. Initially, the input layer receives the input images and forwards them to the backbone network for robust feature extraction. The backbone network generates multiple feature maps of varying sizes, which are then passed to the feature fusion network. In the yolov5 architecture, the feature fusion network produces three feature maps of sizes 80, 40, and 10, respectively, each targeting the detection of small, medium, and large objects.

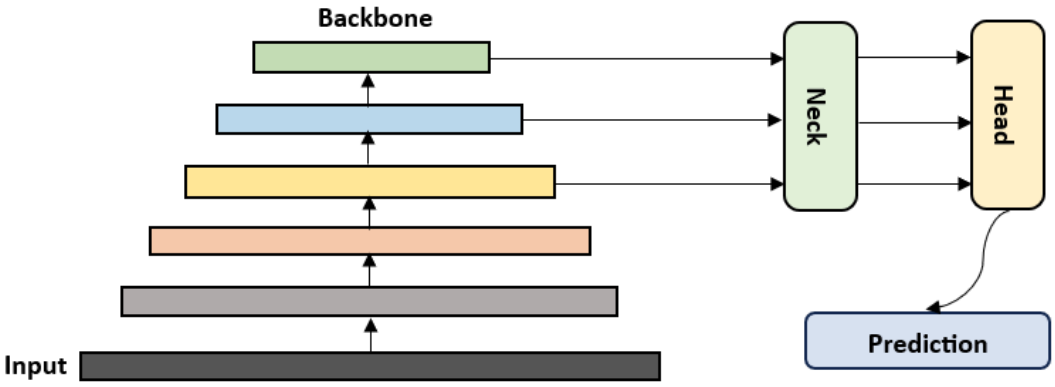


Figure 10: The overall architecture of yolov5.

Subsequently, these three feature maps are processed by the prediction head to obtain the final detection results. These results include the object class, class confidence, and bounding box coordinates, as well as the width and height of the bounding box. To refine the results, certain filtering techniques are applied based on predefined thresholds, such as the class confidence threshold or object threshold. These thresholds help discard irrelevant information. The non-maximum suppression process (NMS) [38] is then employed to further refine the output by eliminating duplicate detections and selecting the most confident and accurate bounding boxes.

By incorporating this architecture, yolov5 effectively leverages the sequential flow of the backbone, feature fusion network, and prediction head to achieve efficient and accurate object detection. The architecture's ability to handle objects of different sizes

and its utilization of filtering techniques, such as NMS, contribute to the production of reliable and precise detection information.

3.2.2 Backbone

The backbone of yolov5 incorporates the powerful CSPDarknet53 (Cross Stage Partial Network) architecture [39], as illustrated in Figure 11. This architecture primarily consists of multiple CBS (Convolutional, BatchNorm, SiLU) modules and C3 (Cross Stage and Cross Block) modules. The stacking of these modules within the backbone facilitates effective feature extraction. The CBS modules work in tandem with the C3 modules, enhancing their capability in extracting informative features. Additionally, a SPPF (Spatial Pyramid Pooling Fusion) module is included at the end of the backbone structure, further enhancing the expression of features within the network.

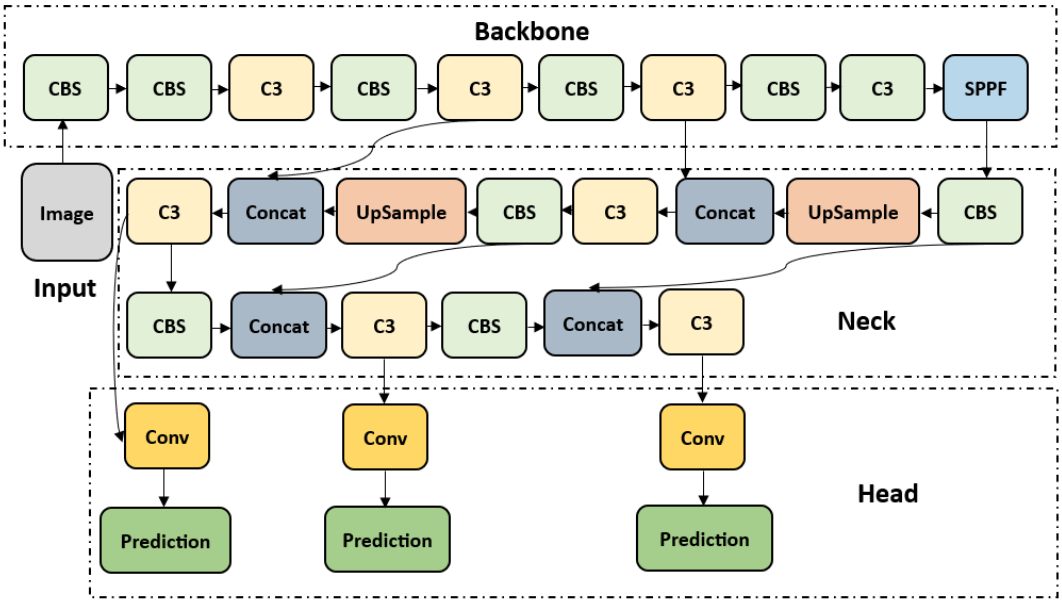


Figure 11: The detail network architecture of yolov5.

Among the various modules utilized, the C3 modules, inspired by the ideas from CSPNet [39], play a pivotal role in the architecture. These modules effectively address the issue of redundant gradient information during network optimization, leading to a significant reduction in complexity while maintaining high accuracy [39]. The success of the C3 modules has resulted in their wide adoption in other prominent networks such as DenseNet [40], ResNeXt [41], and ResNet [18].

By incorporating the CSPDarknet53 backbone, yolov5 harnesses the capabilities of the CBS and C3 modules to achieve superior feature extraction and representation. The innovative design of the C3 modules, coupled with their proven effectiveness in reducing complexity without compromising accuracy, makes them a key component in achieving efficient and accurate object detection.

The architecture of the SPPF (Spatial Pyramid Pooling Fusion) model can be visualized in Figure 12. It consists of multiple pooling layers with different scales,

which are employed to capture features at various levels of detail. In yolov5, a 3-level SPPF is utilized to enhance feature representation.

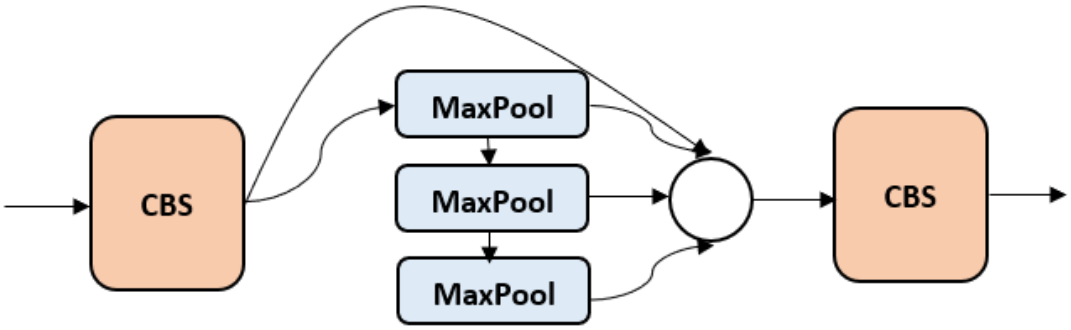


Figure 12: The structure of SPPF.

The SPPF process begins by individually pooling each feature map to produce a single value. Subsequently, each feature map is pooled again to yield four values, and similarly, another pooling operation is performed to obtain 16 values. These pooled vectors from the three levels are then concatenated to generate the final result.

By employing this approach, the SPPF model effectively avoids redundant operations present in previous SPPNet architectures, such as SPPNet [42]. Instead of repeatedly applying the SPP module, the SPPF model capitalizes on the previously max-pooled features, ensuring a more efficient utilization of computational resources.

3.2.3 Neck

Yolov5 incorporates the use of FPN (Feature Pyramid Network) [43] and PAN (Pyramid Attention Network) [44] methods in the neck network, as illustrated in Figure 11. FPN aims to combine low-resolution, semantically strong features with high-resolution, semantically weak features by establishing a top-down pathway and lateral connections. This feature pyramid exhibits rich semantics at all levels and is rapidly constructed from a single input image scale, without compromising representational power, speed, or memory usage [43].

The top-down pathway in FPN involves upsampling the higher-resolution features using a factor of 2 through nearest-neighbor interpolation. These upsampled features are then merged with semantically stronger feature maps from higher pyramid levels through lateral connections. Specifically, the lateral connections merge feature maps of the same spatial size from both the top-down pathway and the bottom-up pathway, which correspond to the output feature maps generated by the downsampling operations of the backbone. The merging process is achieved through element-wise addition, resulting in a fused feature representation.

Building upon the FPN concept, PAN introduces novel modules that provide global context as guidance to select category localization details in the low-level features. By incorporating PAN, the network can effectively leverage global information to enhance the accuracy and localization capabilities of the lower-level features[44]. The distinct structures of FPN and PAN can be observed in Figure 13.

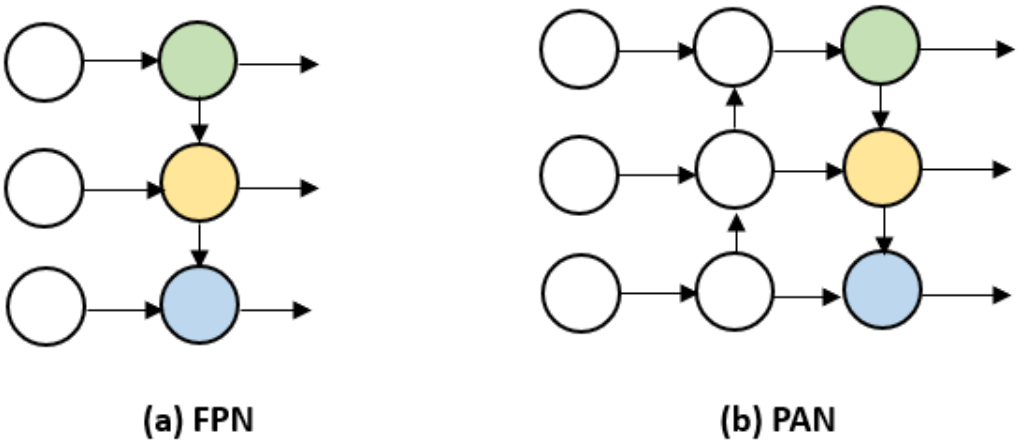


Figure 13: The distinct structures of FPN and PAN.

In summary, Yolov5 utilizes a combination of FPN and PAN techniques in the neck network. FPN enables the fusion of features with different resolutions, enriching the semantic representation at multiple levels. PAN further enhances the localization capabilities of low-level features by incorporating global context guidance. Together, these methods contribute to the overall performance and accuracy of the Yolov5 object detection system.

3.2.4 Head

The head network in Yolov5 plays a crucial role in efficiently computing predictions at different positions within an image. It optimizes the prediction process by considering the positions closest to the centers of ground truth bounding boxes as positive, while designating other positions as negative. This approach ensures that the network focuses on relevant areas for object detection. Figure 14 illustrates all the possible positions for the center of a ground truth box that activate the network's output as positive.

For each positive position, the head network generates regression predictions for the precise position and dimensions of the bounding box. In Yolov5, these predictions are relative to the grid position and anchor size, instead of being relative to the full image as seen in other models like Faster R-CNN [45]. This modification improves the performance of Yolov5 by providing more accurate and efficient bounding box regression.

The bounding box regression equation in Yolov5, as shown in Equation (1), differs from the regression equation used in previous versions of YOLO. Let r_x and r_y denote the unadjusted coordinates of the predicted center point, while g_x, g_y, g_h, and g_w represent the adjusted information of the prediction box. The variables p_h and p_w correspond to the prior anchor's height and width, respectively. Furthermore, s_h and s_w symbolize the offsets calculated by the model. This updated equation takes into account the grid position and anchor size, incorporating specific adjustments that

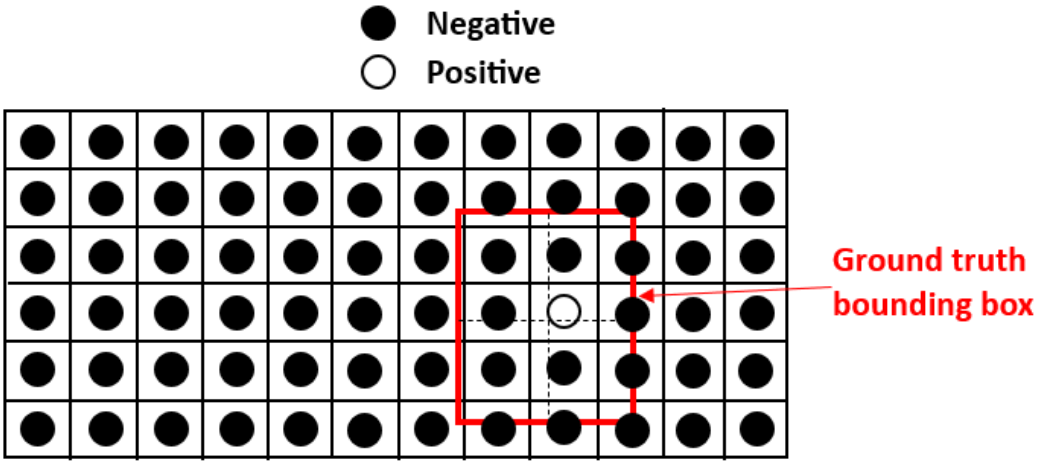


Figure 14: The distinct structures of FPN and PAN.

enhance the precision of bounding box predictions.

$$\begin{aligned} g_x &= 2\sigma(s_x) - 0.5 + r_x \\ g_y &= 2\sigma(s_y) - 0.5 + r_y \\ g_h &= p_h(2\sigma(s_h))^2 \\ g_w &= p_w(2\sigma(s_w))^2 \end{aligned} \tag{1}$$

By optimizing the prediction process through relative regression and incorporating grid positions and anchor sizes, Yolov5 achieves better performance in terms of accuracy and efficiency compared to previous YOLO versions. This innovative approach to bounding box regression ensures that the model can accurately locate and predict objects within the image.

3.2.5 Model Variants

The YOLOv5 architecture comprises four main versions: small (S), medium (M), large (L), and extra large (X), each designed with a different number of parameters and offering progressively higher accuracy rates. A comprehensive overview of these models, including their specific details, can be found in Table 1.

Table 1: Different versions of yolov5.

Model	Size(Pixels)	Params(M)	mAP@0.5	Time CPU b1 (ms)
YOLOv5s	640	7.2	56.8	98
YOLOv5m	640	21.2	64.1	224
YOLOv5l	640	46.5	67.3	430
YOLOv5x	60	86.7	68.9	766

All four YOLOv5 models accept input images with a size of 640 pixels, providing a standardized input resolution for consistent evaluation. The parameter count of the YOLOv5 models increases as the model size grows, ranging from 7.2 million parameters in YOLOv5-S to 86.7 million parameters in YOLOv5-X. This increment in parameter count is reflective of the increased complexity and capacity of the larger models.

With the increase in parameters and model size, there is a corresponding improvement in accuracy rates. The larger models, such as YOLOv5-L and YOLOv5-X, tend to achieve higher accuracy due to their increased capacity to capture intricate details and complex object representations. However, it is important to note that the accuracy gain comes at the expense of longer training times. YOLOv5-S is the fastest to train, while YOLOv5-X requires a longer training duration due to its larger size and higher complexity.

The availability of multiple YOLOv5 variants allows users to choose the model that best suits their specific requirements, striking a balance between accuracy, model size, and training time. The flexibility in model selection empowers practitioners to tailor their object detection solutions to their specific application needs and computational resources. In this work, our objective is to train and evaluate all four versions of YOLOv5, namely small (S), medium (M), large (L), and extra large (X), in order to conduct a comprehensive comparative analysis of their performance. By training and evaluating each variant, we aim to gain insights into their respective strengths and weaknesses, enabling us to make informed decisions about their suitability for different fire scenarios.

3.2.6 Training Parameters

Training the aforementioned YOLOv5 model necessitates significant computational power due to the complexity and demanding nature of the training process. As previously discussed, the efficacy of training is not solely contingent upon the model variant but also relies on the computational capacity employed. In general, higher computational power facilitates faster training and potentially leads to improved results.

In our work, we conducted model training utilizing a pre-trained base model obtained from the official YOLOv5 website [35]. To leverage the necessary computational resources, we employed an Nvidia GeForce RTX 3060 GPU, as indicated in Table 2.

By employing this powerful GPU, we aimed to expedite the training process and enhance the overall efficiency of our model training. The Nvidia GeForce RTX 3060 GPU offers robust computational capabilities, enabling accelerated training iterations and facilitating the exploration of a larger solution space.

Table 2 provides valuable insights into two crucial parameters that play a pivotal role in the training process: batch size and epoch. The batch size refers to the number of training samples processed before updating the model weights [46]. In our work, we carefully selected a batch size of 16, indicating that the model processes 16 training images in each iteration and subsequently updates their weights accordingly.

Choosing an appropriate batch size is a critical consideration in training, as it

Table 2: Some training parameters of the proposed yolov5 model.

Training parameter	Value
Computation Capacity	Nvidia GeForce RTX 3060 GPU
Batch Size	16
Epoch	200

directly impacts the model's performance. If the batch size is too small, it can lead to slower training due to frequent weight updates. However, smaller batch sizes typically yield improved accuracy as more weight updates are performed, allowing the model to learn more fine-grained patterns from the data.

Conversely, if the batch size is too large, the training process can be faster since fewer weight updates are performed. However, using larger batch sizes may sacrifice the model's performance, as the model may not be able to capture nuanced details and variations within the training data.

Therefore, it is crucial to strike a balance and choose an appropriate batch size that aligns with the specific characteristics of the model and the training dataset. This requires careful consideration and experimentation to find the optimal batch size that achieves a balance between training speed and model accuracy.

By selecting a batch size of 16 for our training process, we aimed to strike a balance between efficiency and performance, allowing the model to update its weights based on a reasonable number of training samples at each iteration. This choice reflects a careful consideration of the model's capacity and the characteristics of the training dataset, aiming to achieve a successful training outcome.

Epoch also referred to as training iterations, holds significant importance as a crucial training parameter throughout the training process. An epoch represents a complete iteration where all training samples are processed. To illustrate, suppose our training dataset consists of 1000 images, and the batch size is set to 16. In this scenario, it would take approximately 64 iterations to complete one epoch, as $64 \times 16 \approx 1000$. During each epoch, the model receives the training images in batches, processes them, and updates its weights after handling all images within a batch.

Following the completion of each epoch, an evaluation phase ensues, employing the validation dataset to assess the trained model's performance and identify potential issues such as overfitting or other challenges. This evaluation helps estimate the model's generalization ability and determine if it has learned useful patterns or if it has become overly specialized to the training data.

The selection of the appropriate number of epochs is pivotal in achieving optimal performance. If the number of epochs is insufficient, the model may not be trained adequately, leading to underfitting. Underfitting occurs when the model fails to capture the complexity of the training data and performs poorly even on the training dataset. On the other hand, excessively large numbers of epochs can result in overfitting, where the model becomes too specialized to the training dataset, acquiring detailed patterns and noise, but struggling to generalize to unseen datasets.

Finding the balance between underfitting and overfitting is paramount. It is crucial to choose an appropriate number of epochs that allows the model to converge and learn meaningful representations from the training data without becoming overly specific. This ensures the model's ability to generalize well to unseen datasets and produce reliable predictions.

It is important to note that the selection of the number of epochs may vary depending on the specific dataset, model architecture, and problem domain. Hence, conducting thorough experimentation and validation is crucial to identify the optimal number of epochs for each specific scenario.In our work, we selected a total of 200 epochs for training. This choice was based on careful consideration and experimentation to strike a balance between allowing the model to learn from the data adequately while preventing overfitting. By choosing this optimal number of epochs, we aimed to attain the best performance and generalization capabilities for our trained model.

3.2.7 Evaluation Metric

In this work, the evaluation of the object detection model is based on two primary metrics: mean average precision (mAP) and the F1 Curve. The mAP@.5 metric specifically measures the average accuracy of successfully detecting objects in the test dataset. It quantifies the model's performance by comparing its predictions against the ground truth annotations of the objects in the test dataset, calculating the average accuracy across the entire test dataset [46].

To determine if a model's prediction is correct, an essential aspect is the assessment of the overlay or intersection between the predicted bounding box and the ground truth annotation. This assessment is typically conducted using the Intersection over Union (IoU) threshold. The IoU threshold represents the ratio of overlap between the predicted bounding box and the ground truth annotation of an object within a given image. In this work, we set the IoU threshold to 0.5.

When the overlap between the predicted bounding box and the ground truth annotation exceeds the specified IoU threshold, the model prediction is considered correct. Conversely, if the overlap falls below the IoU threshold, the prediction is deemed incorrect. By applying this threshold-based evaluation, we can assess the model's ability to accurately localize and classify objects in the test dataset.

The utilization of the IoU threshold ensures that the model's predictions align closely with the ground truth annotations, as it captures the degree of spatial overlap required for a successful detection. By setting the IoU threshold at 0.5, we strike a balance between precision and recall, considering a significant level of overlap while allowing for some flexibility in the bounding box match criteria.

By employing the mAP@.5 metric and the IoU threshold-based evaluation, we aim to provide a comprehensive and robust assessment of the object detection model's performance. These evaluation measures help quantify the accuracy and localization capabilities of the model, contributing to a more objective and detailed analysis of its effectiveness in detecting objects within the test dataset.

The F1 curve serves as an additional evaluation metric commonly used for classification models. It provides insights into the accuracy and recall of the model

at different thresholds, offering a comprehensive assessment of its performance. Accuracy measures the proportion of correct predictions out of the total predictions, while recall quantifies the proportion of correct predictions over the total number of positive objects in the dataset.

The F1 curve combines both accuracy and recall to strike a balance between the two metrics, ensuring a comprehensive evaluation of the model's classification performance. Equation (2) illustrates the calculation of the F1 score, which considers both precision and recall.

To construct the F1 curve, we begin by selecting various thresholds and calculating the corresponding accuracy and recall values for each threshold [46]. By systematically varying the threshold, we can observe how the accuracy and recall change across different classification settings.

Subsequently, we plot the accuracy and recall values on a graph, enabling us to visualize the variations in accuracy and recall as the threshold is adjusted. This graphical representation of the F1 curve provides valuable insights into the model's performance and aids in understanding the trade-off between accuracy and recall.

By incorporating the F1 curve into our evaluation process, we can gain a more nuanced understanding of the model's classification capabilities across different threshold settings. This analysis helps identify an optimal threshold that balances precision and recall, ensuring robust performance across various classification scenarios.

$$F1 = 2 * \frac{(Accuracy * Recall)}{(Accuracy + Recall)} \tag{2}$$

3.2.8 Results

As mentioned earlier, our research endeavors to comprehensively assess the performance of different variants of the yolov5 model on our collected fire dataset. To achieve this, we trained all four versions of yolov5, namely yolov5s, yolov5m, yolov5l, and yolov5x. By training and evaluating each variant, we aim to gain a comprehensive understanding of their strengths, weaknesses, and suitability for fire detection tasks.

The first model we trained is yolov5s. The yolov5s variant represents the smallest and most lightweight version of the yolov5 model. With fewer parameters compared to the other variants, yolov5s offers faster inference times and is well-suited for resource-constrained environments. By training and evaluating yolov5s on our collected dataset, we can assess its ability to accurately detect fire while maintaining efficient computational performance. Figure 15 presents the confusion matrix of the trained model, which provides valuable insights into its performance. The true positive (TP) value for fire is 0.67, indicating that the model has successfully detected a significant portion of actual fire instances. Additionally, the true negative (TN) value for fire is 0.90, indicating high accuracy in correctly identifying non-fire instances. These results suggest that the model exhibits a good ability to detect fire while minimizing false detections.

To further evaluate the model's performance, we generated precision and recall curves based on confidence values, as depicted in Figure 16 and Figure 17, respectively.

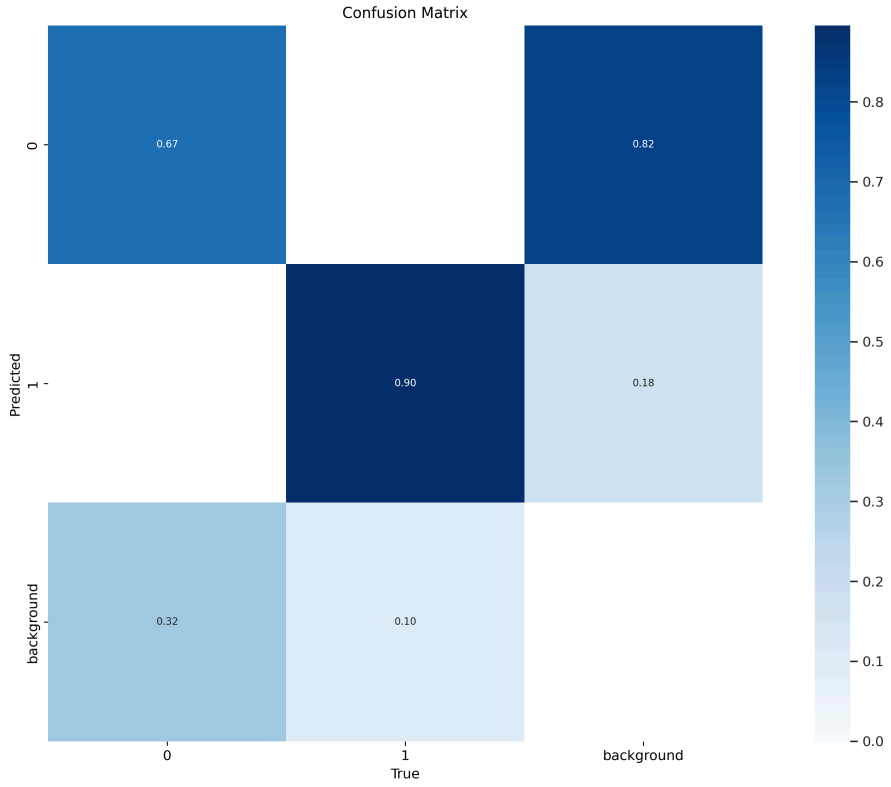


Figure 15: The confusion matrix of yolov5s model.

At a confidence threshold of 0.957, all classes achieve a precision of 1, implying a high level of confidence in the model's predictions. Similarly, at a confidence threshold of 0, the recall reaches 0.90, indicating that the model has successfully captured a significant portion of the fire instances.

Combining precision and recall, we obtained the precision-recall curve, as shown in Figure 19. The curve reveals that the model achieved an mAP@.5 (mean average precision at IoU threshold of 0.5) value of 0.772. This means that approximately 77.2% of the predictions on the test dataset were correct, further highlighting the model's effectiveness in fire detection.

Finally, the F1 curve, displayed in Figure 19, demonstrates the model's performance across different confidence thresholds. With a confidence threshold of 0.514, the model attained an F1 score of 0.77, indicating a balance between precision and recall. These results collectively indicate the model's strong performance in fire detection tasks.

The second model we trained is yolov5m. Yolov5m is a medium-sized variant that strikes a balance between model complexity and computational efficiency. With an increased number of parameters compared to yolov5s, yolov5m offers improved detection accuracy while still maintaining a reasonable inference time. Evaluating yolov5m on our dataset enables us to explore its trade-off between accuracy and efficiency. Figure 20 presents the confusion matrix of the trained model, which provides valuable insights into its performance. The true positive (TP) value for fire is

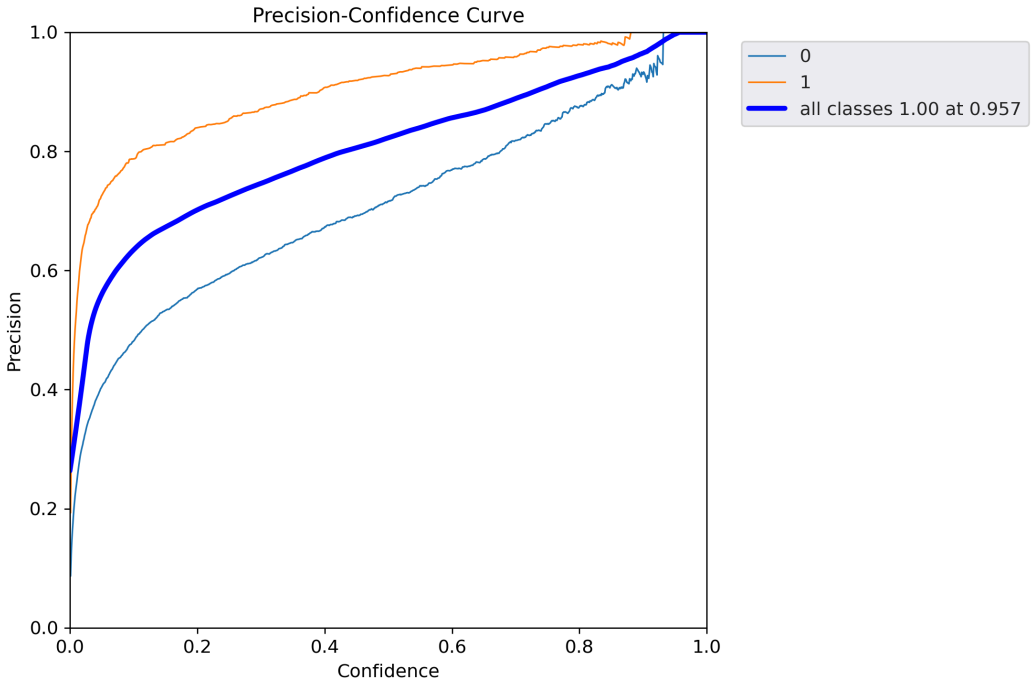


Figure 16: The precision curve of yolov5s model.

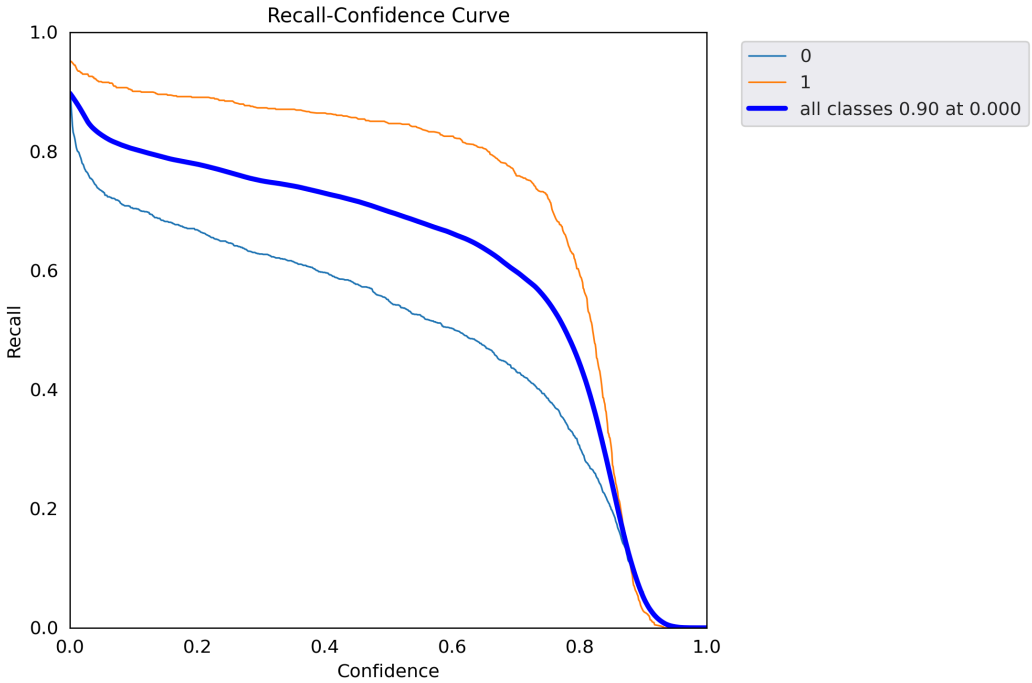


Figure 17: The recall curve of yolov5s model.

0.68, indicating that the model has successfully detected a significant portion of actual fire instances. Additionally, the true negative (TN) value for fire is 0.92, indicating

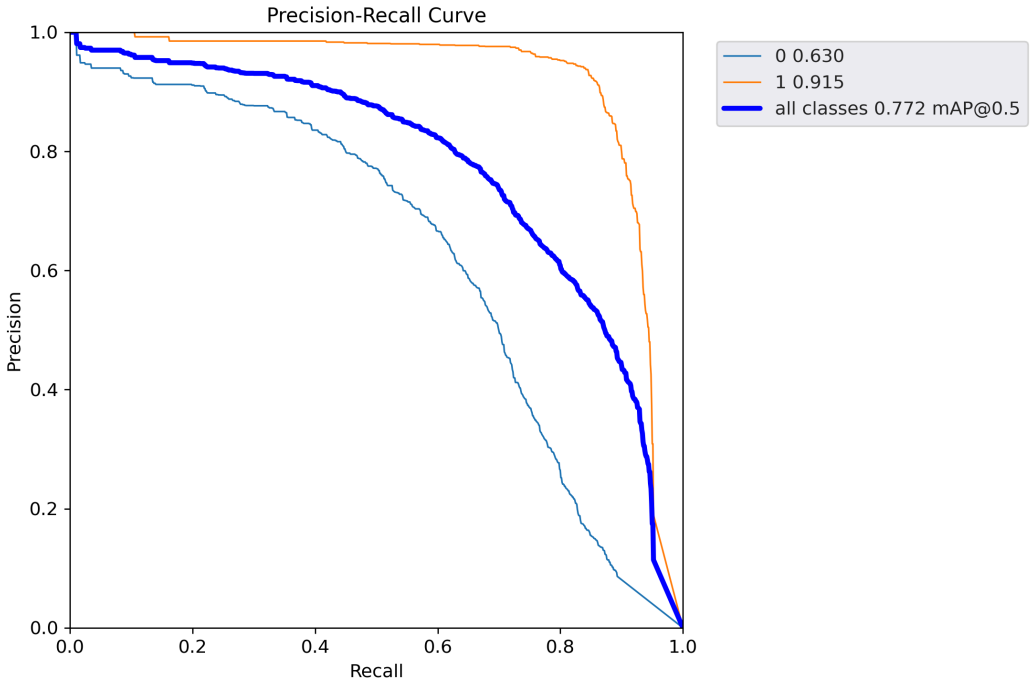


Figure 18: The precision and recall curve of yolov5s model.

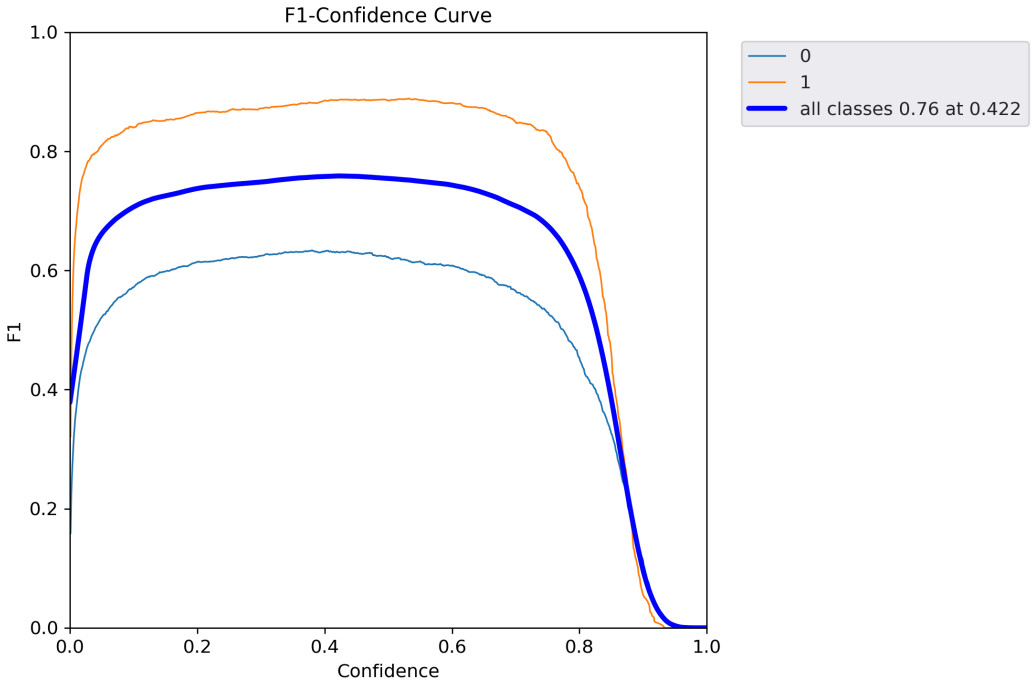


Figure 19: The F1 curve of yolov5s model.

high accuracy in correctly identifying non-fire instances. These results suggest that the model exhibits a good ability to detect fire while minimizing false detections.

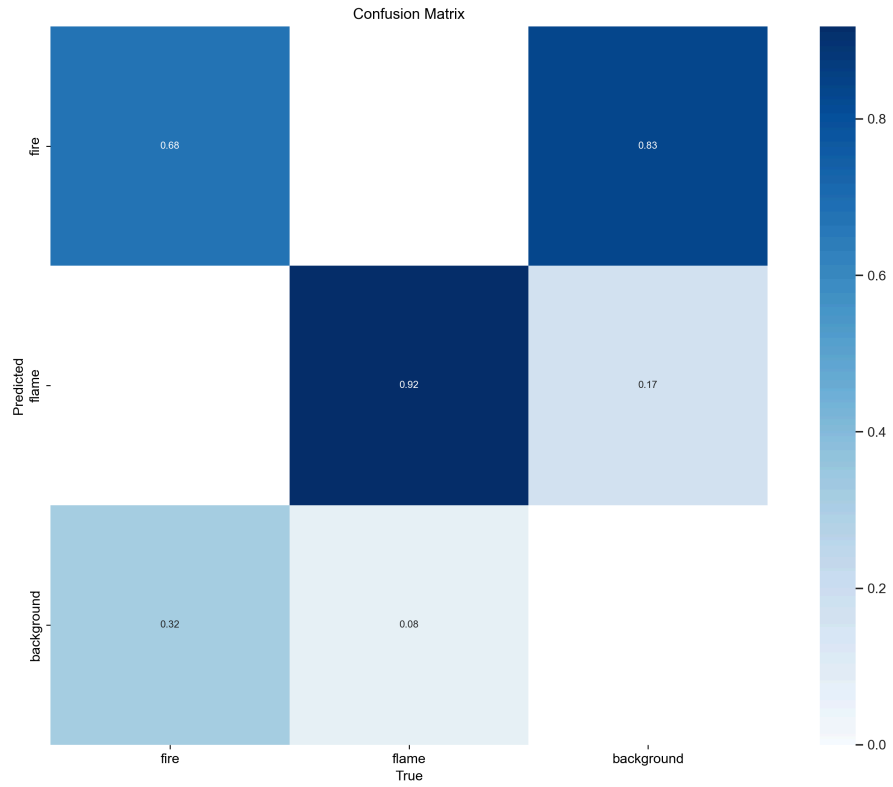


Figure 20: The confusion matrix of yolov5m model.

To further evaluate the model's performance, we generated precision and recall curves based on confidence values, as depicted in Figure 21 and Figure 22, respectively. At a confidence threshold of 0.957, all classes achieve a precision of 1, implying a high level of confidence in the model's predictions. Similarly, at a confidence threshold of 0, the recall reaches 0.89, indicating that the model has successfully captured a significant portion of the fire instances.

Combining precision and recall, we obtained the precision-recall curve, as shown in Figure 24. The curve reveals that the model achieved an mAP@.5 (mean average precision at IoU threshold of 0.5) value of 0.777. This means that approximately 77.7% of the predictions on the test dataset were correct, further highlighting the model's effectiveness in fire detection.

Finally, the F1 curve, displayed in Figure 24, demonstrates the model's performance across different confidence thresholds. With a confidence threshold of 0.514, the model attained an F1 score of 0.77, indicating a balance between precision and recall. These results collectively indicate the model's strong performance in fire detection tasks.

The third model we trained is yolov5l. Yolov5l represents a larger and more powerful variant of the yolov5 model. With an even higher number of parameters, yolov5l aims to achieve superior detection accuracy, especially for challenging fire scenarios. By training and evaluating yolov5l on our dataset, we can assess its ability to capture fine-grained details and handle complex fire detection tasks. Figure 25 presents

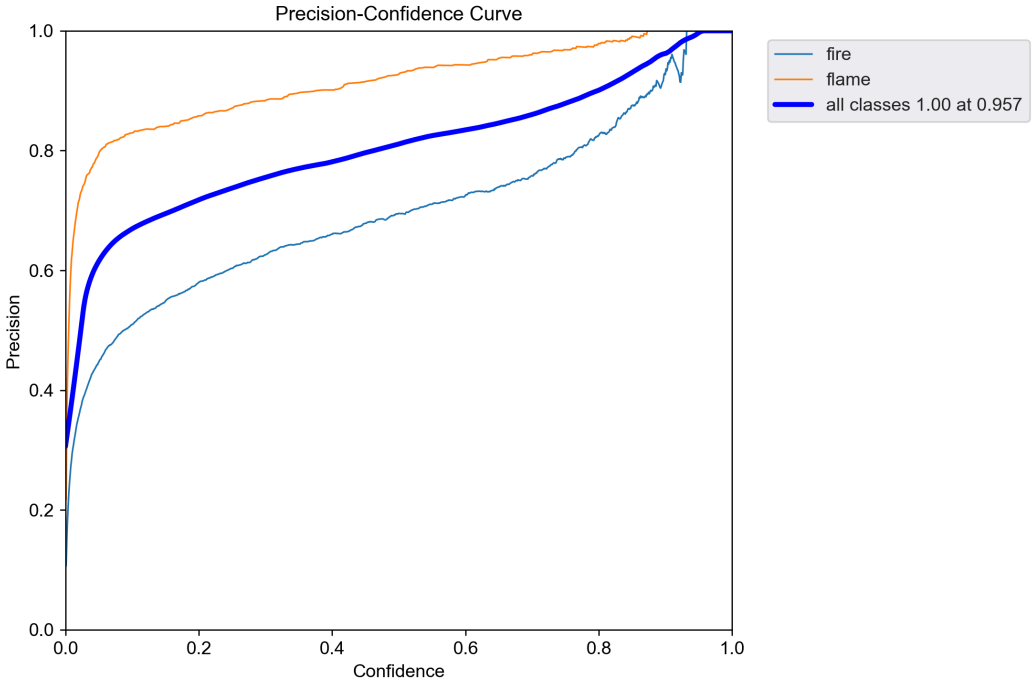


Figure 21: The precision curve of yolov5m model.

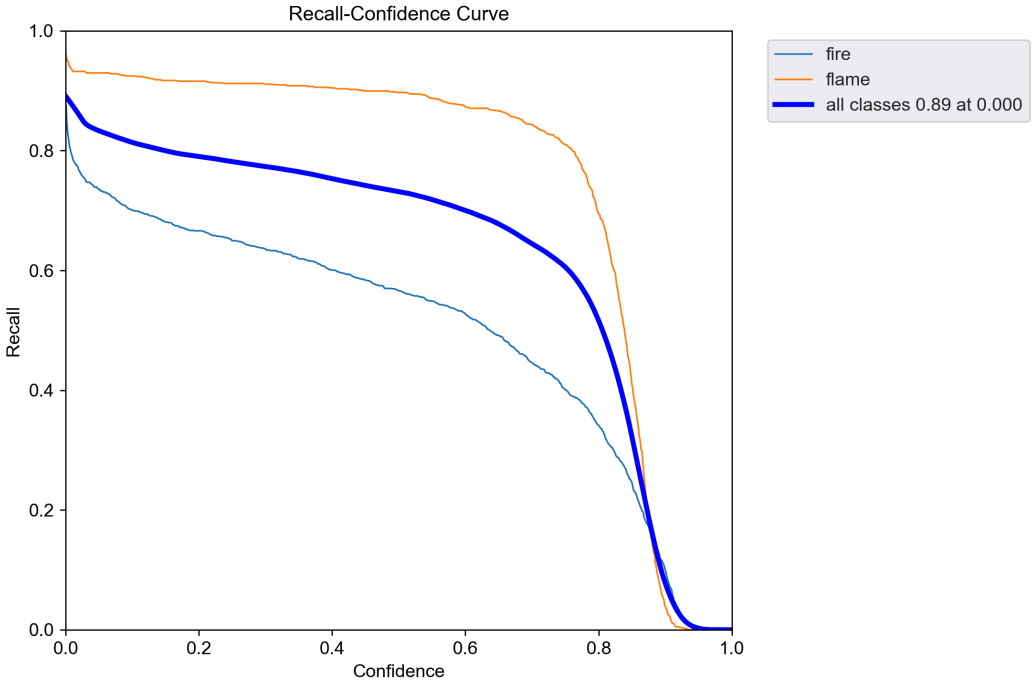


Figure 22: The recall curve of yolov5m model.

the confusion matrix of the trained model, which provides valuable insights into its performance. The true positive (TP) value for fire is 0.64, indicating that the model

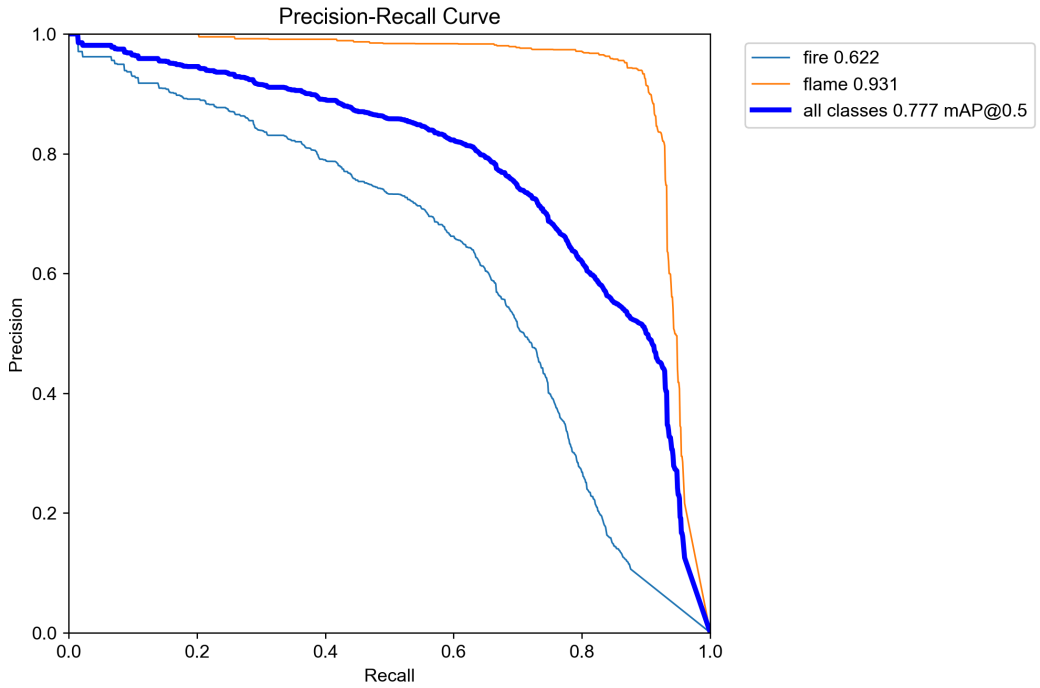


Figure 23: The precision and recall curve of yolov5m model.

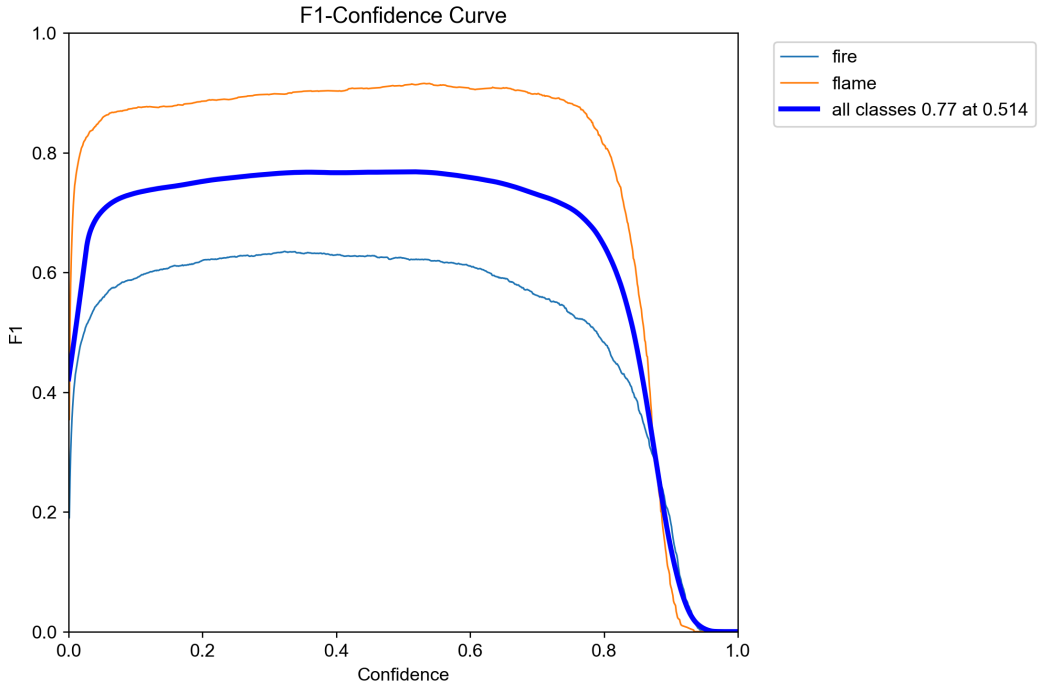


Figure 24: The F1 curve of yolov5m model.

has successfully detected a significant portion of actual fire instances. Additionally, the true negative (TN) value for fire is 0.89, indicating high accuracy in correctly

identifying non-fire instances. These results suggest that the model exhibits a good ability to detect fire while minimizing false detections.

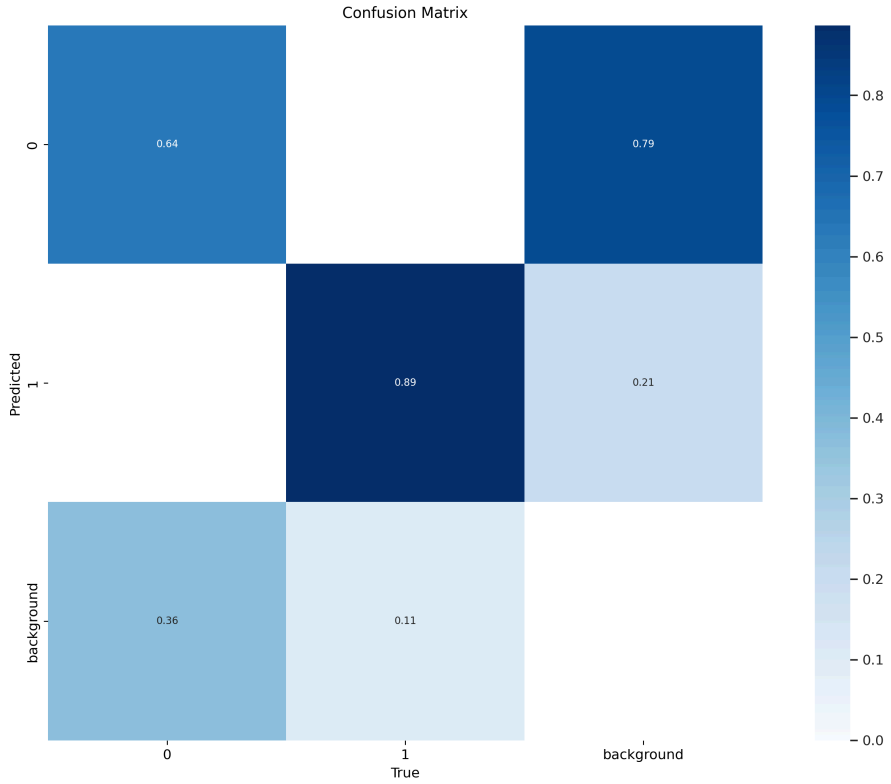


Figure 25: The confusion matrix of yolov5l model.

To further evaluate the model's performance, we generated precision and recall curves based on confidence values, as depicted in Figure 26 and Figure 27, respectively. At a confidence threshold of 0.997, all classes achieve a precision of 1, implying a high level of confidence in the model's predictions. Similarly, at a confidence threshold of 0, the recall reaches 0.83, indicating that the model has successfully captured a significant portion of the fire instances.

Combining precision and recall, we obtained the precision-recall curve, as shown in Figure 29. The curve reveals that the model achieved an mAP@.5 (mean average precision at IoU threshold of 0.5) value of 0.773. This means that approximately 77.3% of the predictions on the test dataset were correct, further highlighting the model's effectiveness in fire detection.

Finally, the F1 curve, displayed in Figure 29, demonstrates the model's performance across different confidence thresholds. With a confidence threshold of 0.463, the model attained an F1 score of 0.77, indicating a balance between precision and recall. These results collectively indicate the model's strong performance in fire detection tasks.

The last model we trained is yolov5x. Yolov5x represents the largest and most computationally intensive variant of the yolov5 model. With significantly more

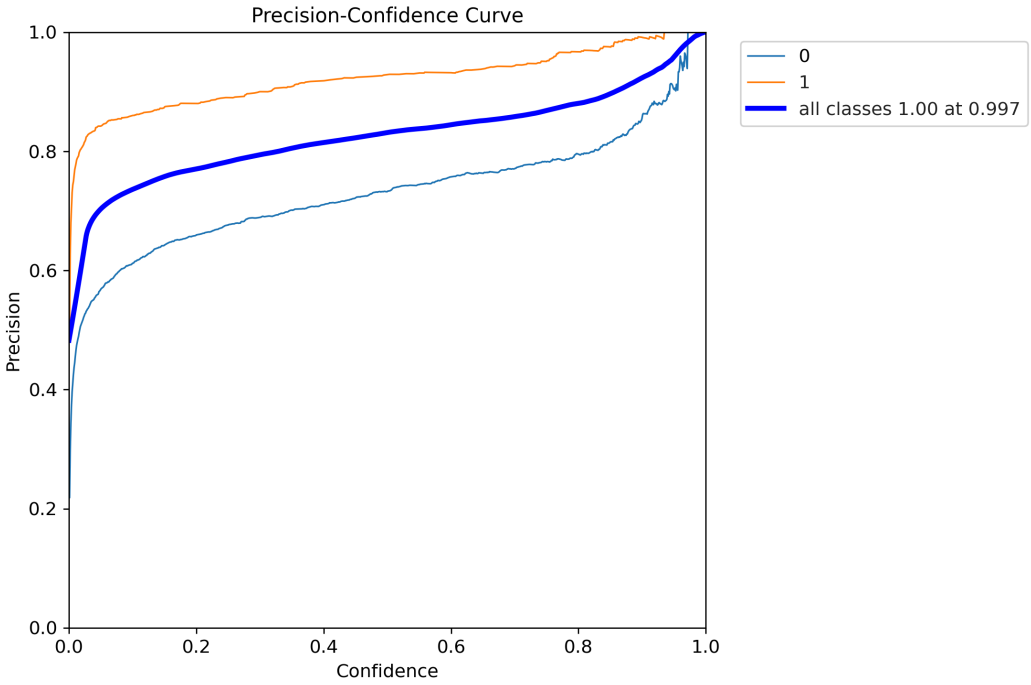


Figure 26: The precision curve of yolov5l model.

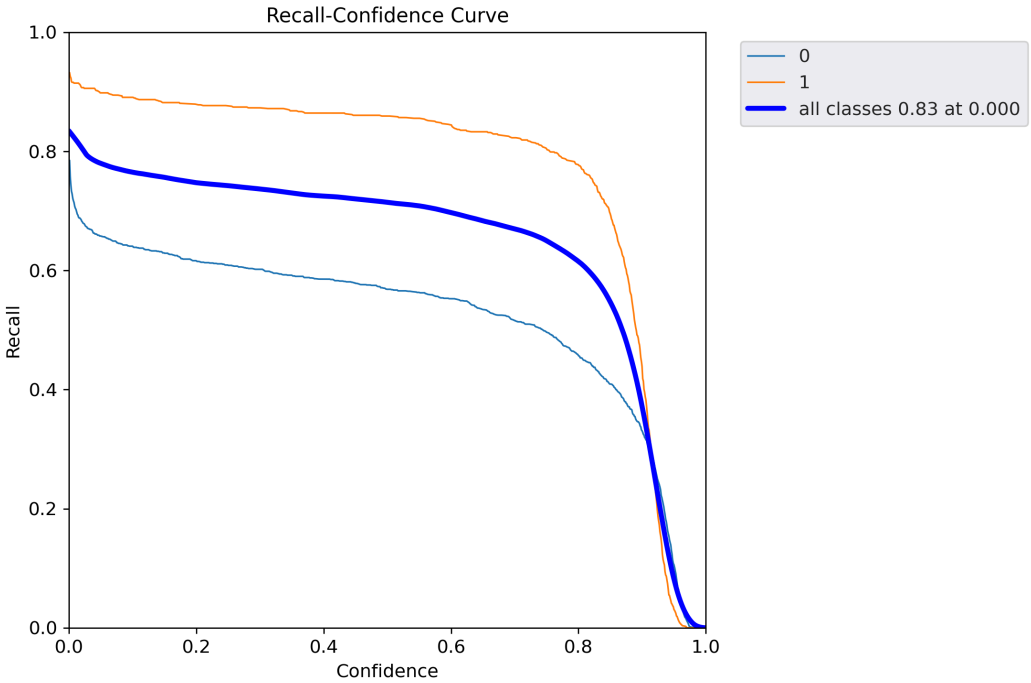


Figure 27: The recall curve of yolov5l model.

parameters, yolov5x offers the potential for even higher detection accuracy and the ability to handle more diverse fire scenarios. However, this variant may come at

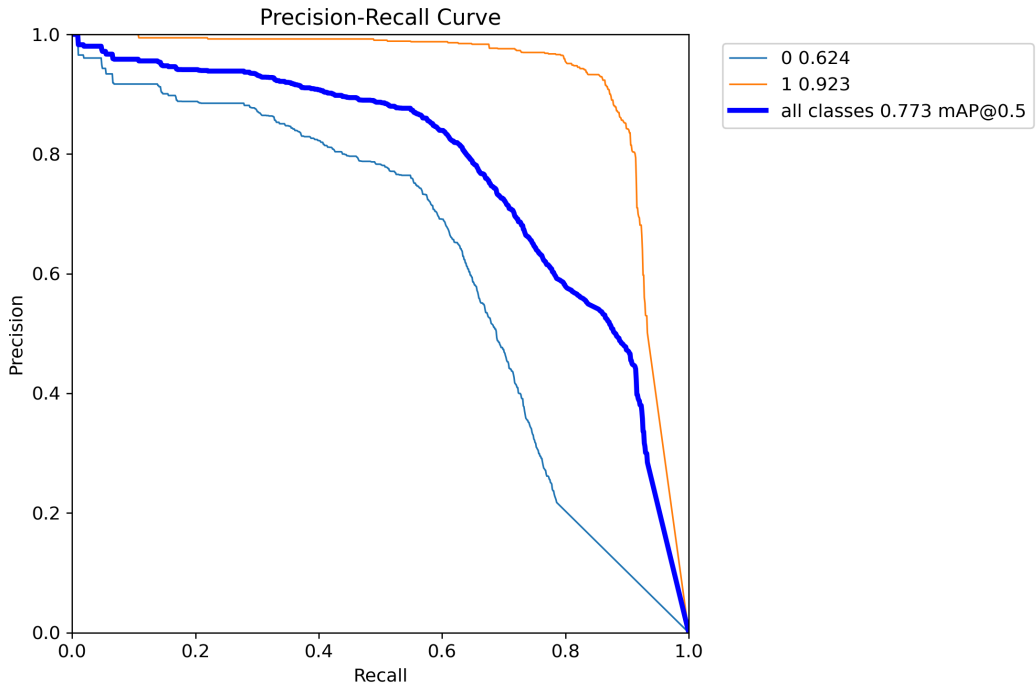


Figure 28: The precision and recall curve of yolov5l model.

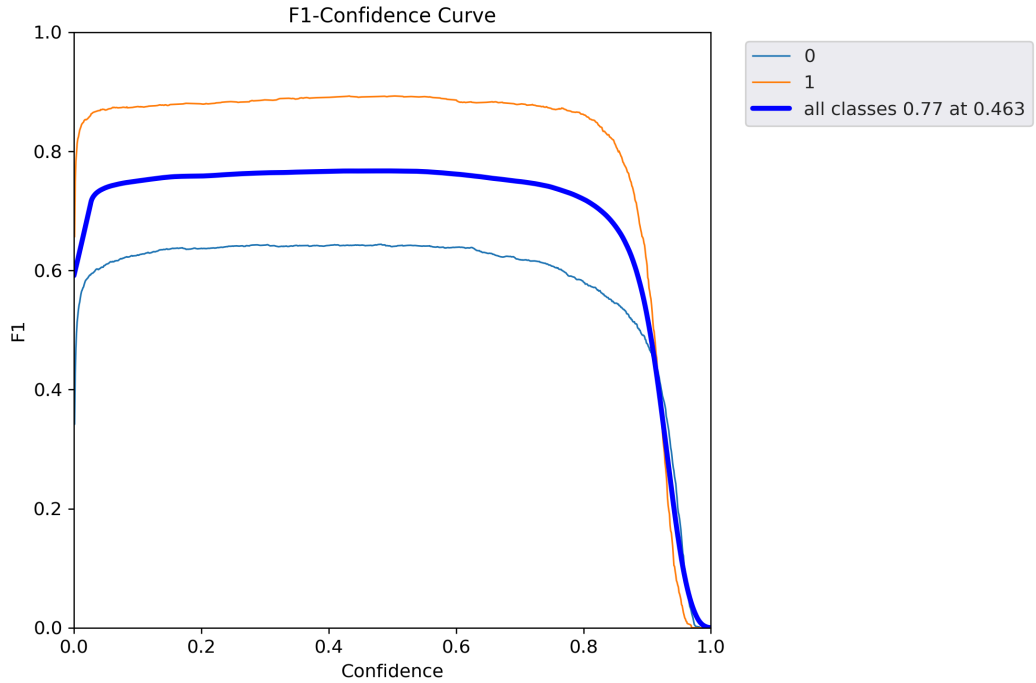


Figure 29: The F1 curve of yolov5l model.

the cost of increased inference time and computational requirements. By training and evaluating yolov5x on our dataset, we can explore its potential for achieving

state-of-the-art performance in fire detection. Figure 30 presents the confusion matrix of the trained model, which provides valuable insights into its performance. The true positive (TP) value for fire is 0.62, indicating that the model has successfully detected a significant portion of actual fire instances. Additionally, the true negative (TN) value for fire is 0.89, indicating high accuracy in correctly identifying non-fire instances. These results suggest that the model exhibits a good ability to detect fire while minimizing false detections.

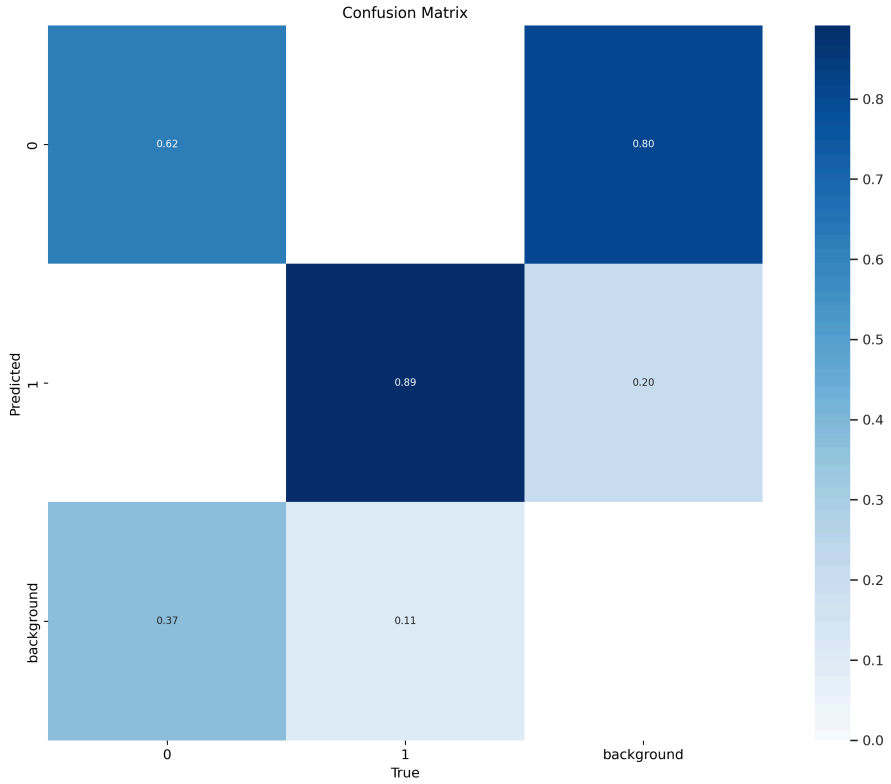


Figure 30: The confusion matrix of yolov5x model.

To further evaluate the model's performance, we generated precision and recall curves based on confidence values, as depicted in Figure 31 and Figure 32, respectively. At a confidence threshold of 0.998, all classes achieve a precision of 1, implying a high level of confidence in the model's predictions. Similarly, at a confidence threshold of 0, the recall reaches 0.82, indicating that the model has successfully captured a significant portion of the fire instances.

Combining precision and recall, we obtained the precision-recall curve, as shown in Figure 34. The curve reveals that the model achieved an mAP@.5 (mean average precision at IoU threshold of 0.5) value of 0.781. This means that approximately 78.1% of the predictions on the test dataset were correct, further highlighting the model's effectiveness in fire detection.

Finally, the F1 curve, displayed in Figure 34, demonstrates the model's performance across different confidence thresholds. With a confidence threshold of 0.528, the

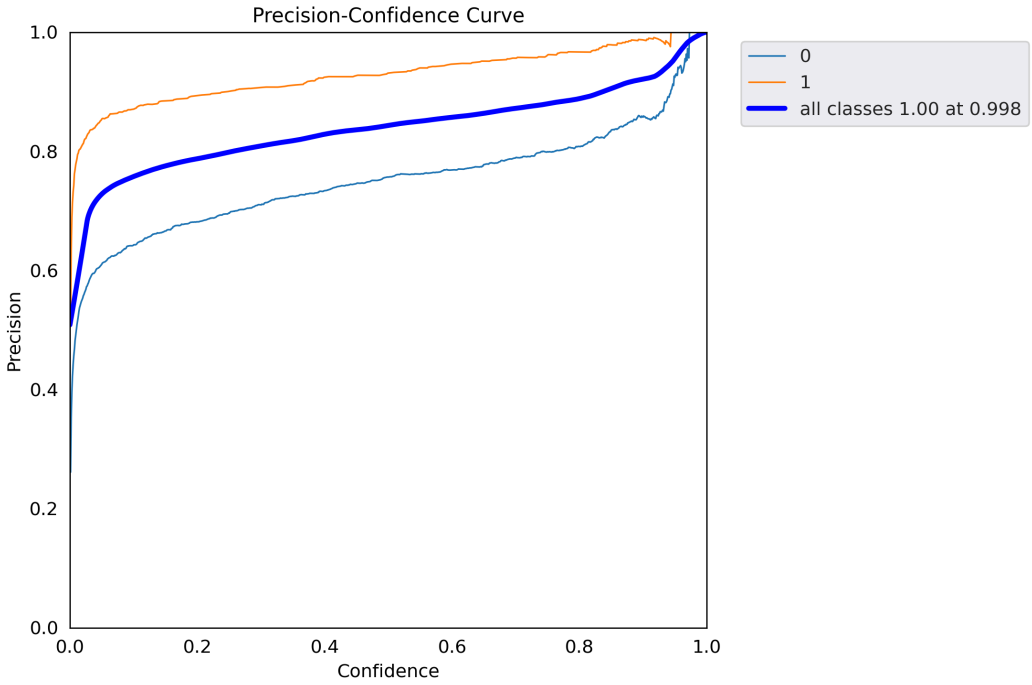


Figure 31: The precision curve of yolov5x model.

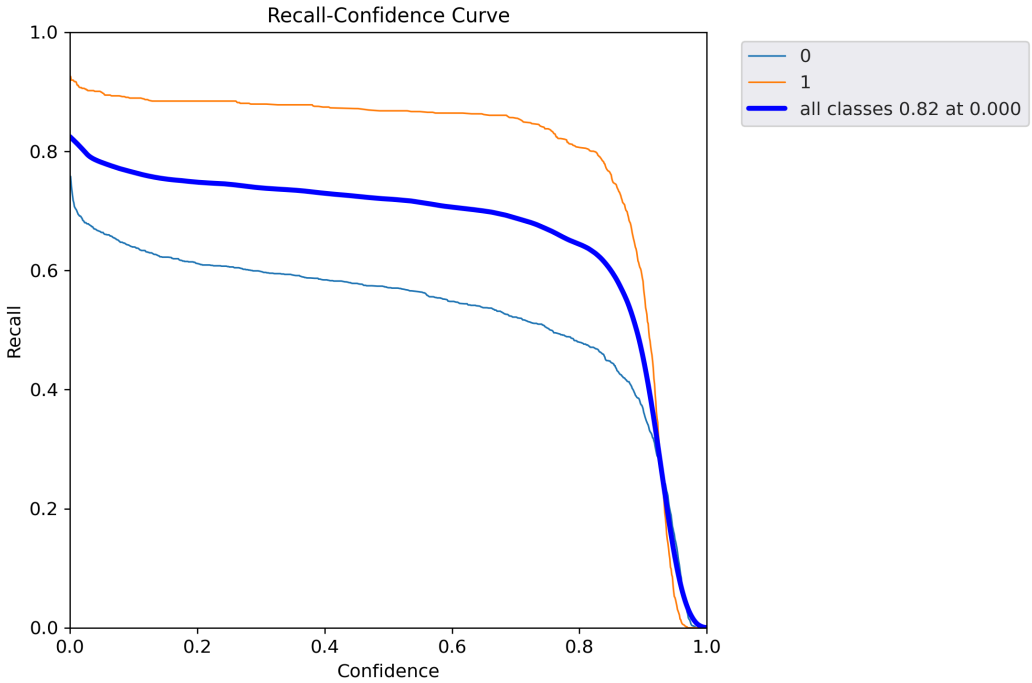


Figure 32: The recall curve of yolov5x model.

model attained an F1 score of 0.78, indicating a balance between precision and recall. These results collectively indicate the model's strong performance in fire detection

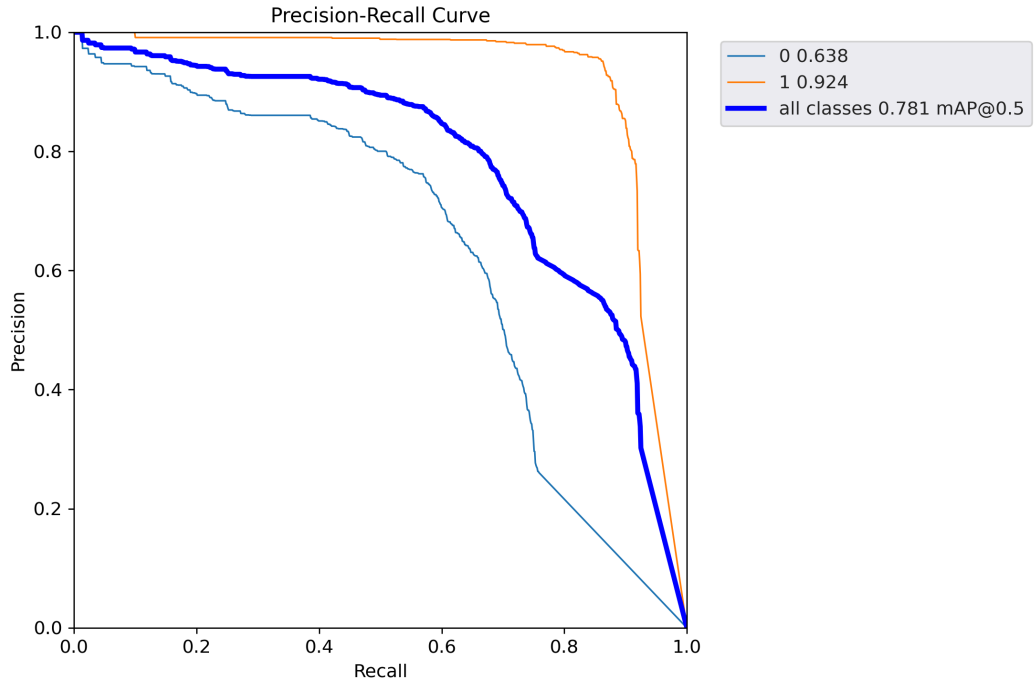


Figure 33: The precision and recall curve of yolov5x model.

tasks.

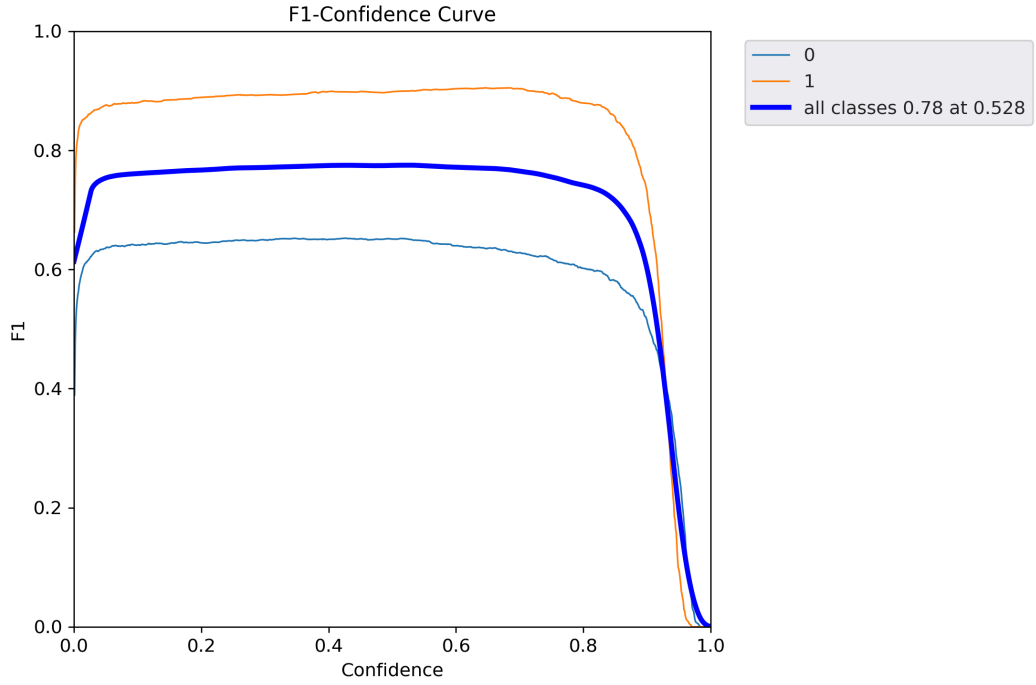


Figure 34: The F1 curve of yolov5x model.

Overall, the evaluation results suggest that the trained model exhibits promising performance in detecting fire instances. With high TP and TN values, high precision and recall at suitable confidence thresholds, and a satisfactory mAP@.5 score, the model demonstrates its effectiveness in accurately identifying fire and minimizing false detections.

3.3 Deployment

The trained model is seamlessly deployed in the cloud, enabling effortless integration with IoT devices. The carefully designed workflow encompasses multiple components, as illustrated in Figure 35, ensuring a seamless flow of data and communication between the cloud-based model and the IoT devices.

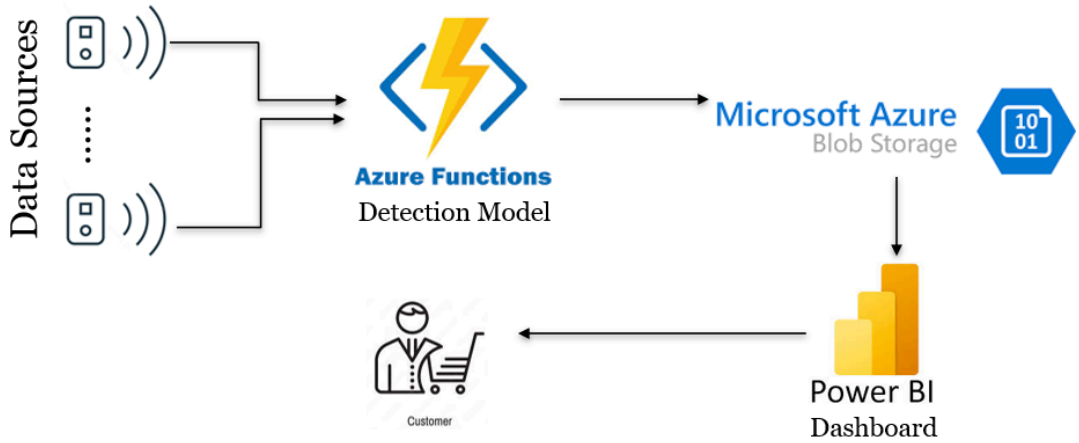


Figure 35: The workflow of the deployed model.

3.3.1 Data Sources

The data sources comprise a diverse range of IoT devices and other systems equipped with image-capturing capabilities, all aimed at leveraging an inference model to gather crucial fire-related information. The primary objective of these sources is to capture real-time images of fire incidents and promptly analyze them using the deployed inference model. Through seamless integration, these sources can effortlessly upload captured images to the model, initiating the analysis process. Once the analysis is complete, the sources retrieve the URL of the analyzed images, allowing for further examination and dissemination of the fire analysis results. One illustrative example of the returned analyzed images can be observed in Figure 36. This figure showcases a representative output obtained after running the inference model on a given input image. Upon examination of Figure 36, it is evident that the model successfully identifies and labels multiple objects present within the image. Specifically, the analysis reveals the presence of two individuals, a chair, and a prominent candle flame.

The detection of these objects signifies the model's ability to discern and categorize different elements within the image, specifically focusing on objects that are relevant

to fire analysis. By accurately identifying these objects, the model demonstrates its efficacy in recognizing potential fire hazards or situations where fire-related information is of utmost importance.

The inclusion of this specific example in Figure 36 aids in providing a concrete representation of the model's detection capabilities and offers readers a visual reference to better comprehend the effectiveness of the fire analysis model in action.

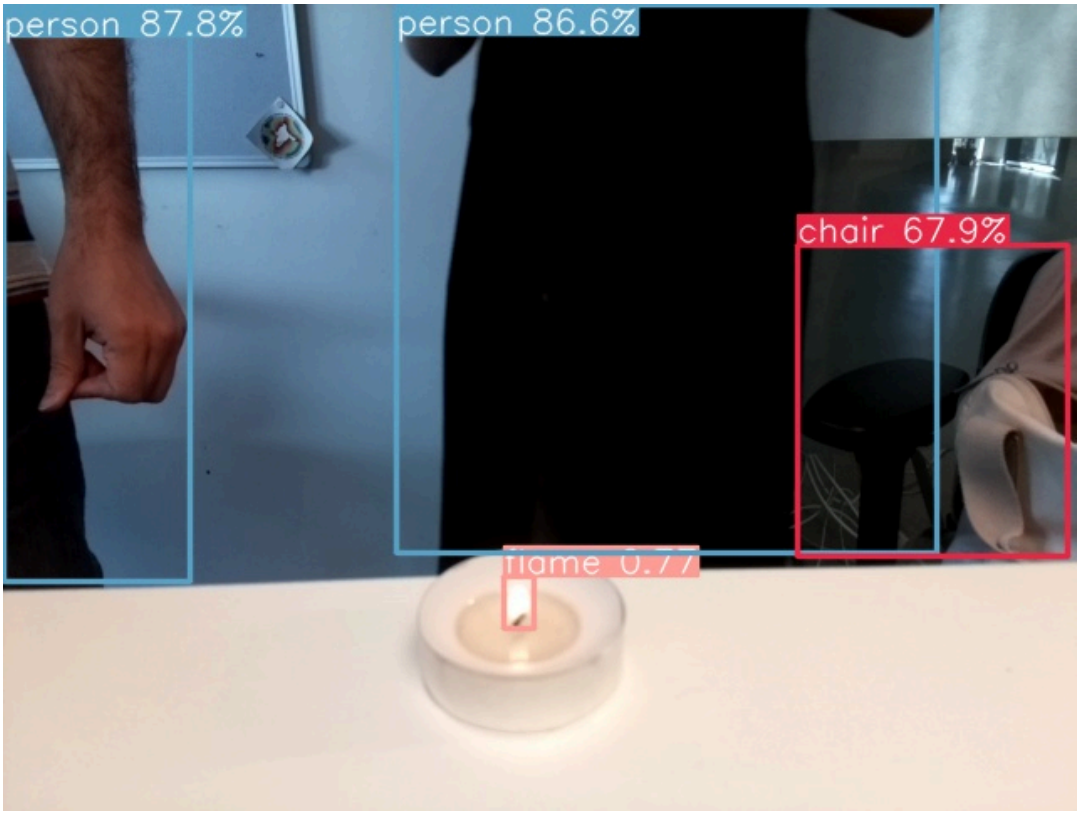


Figure 36: One example of analyzed image.

3.3.2 Azure Functions

The trained model is deployed using Azure Functions, an on-demand cloud service that offers comprehensive infrastructure and continuously updated resources for efficient application execution. The decision to select Azure Functions is driven by several compelling reasons. Firstly, it enables us to focus exclusively on the critical code components, specifically the inference of the trained model, while leaving the remaining operational aspects to Functions. By abstracting away the underlying infrastructure and handling operational tasks, Azure Functions allows us to dedicate our efforts to the core functionality of the model, accelerating development and deployment processes.

With its serverless computing capabilities, Azure Functions empowers us to create versatile web APIs, process IoT data streams, and seamlessly ingest data into databases, among other powerful functionalities. This serverless architecture proves highly efficient for real-time processing, particularly when the requirements are limited

to basic operations. The inherent scalability of Azure Functions ensures that resources are automatically allocated and adjusted based on the workload demands, optimizing performance and minimizing costs as the application scales.

Furthermore, Azure Functions seamlessly integrates with other Azure services, creating a cohesive ecosystem that enables us to leverage a wide array of tools and functionalities for enhanced application development and deployment. The integration capabilities streamline data flows, facilitate inter-service communication, and enable easy access to specialized services, empowering us to leverage the full potential of Azure's rich ecosystem. This comprehensive integration further accelerates development cycles and enables the creation of robust and feature-rich applications.

In addition to its integration prowess, Azure Functions provides robust monitoring and logging capabilities. These features allow us to gain valuable insights into system performance, track the execution of functions, and effectively troubleshoot any issues that may arise. By utilizing the monitoring and logging functionalities of Azure Functions, we can proactively identify bottlenecks, optimize performance, and ensure the reliability and stability of the deployed model.

Given the critical importance of obtaining prompt fire analysis results in real-world scenarios, the utilization of Azure Functions becomes crucial for achieving expeditious outcomes. Its ability to handle the operational aspects, efficient real-time processing, seamless integration with other Azure services, and comprehensive monitoring capabilities contribute to the overall efficiency, reliability, and performance of the fire analysis system.

3.3.3 Azure Blob Storage

In Azure Functions, the trained model receives the posted images and performs the inference. Subsequently, the detected image and the originally posted images are securely and efficiently stored in Azure Blob Storage, which is Microsoft's robust and scalable object storage solution purpose-built for effectively managing vast amounts of unstructured data. Leveraging Azure Blob Storage offers several distinct advantages. Firstly, it provides high scalability, allowing seamless storage expansion as data volumes grow over time, ensuring that the system can handle increasing storage requirements without compromising performance or stability.

Additionally, Azure Blob Storage ensures data durability and availability through its built-in redundancy and replication mechanisms. By automatically creating redundant copies of stored data across multiple storage nodes and data centers, Azure Blob Storage enhances data durability, minimizing the risk of data loss and maximizing availability. This fault-tolerant design guarantees that the system remains resilient even in the face of hardware failures or unexpected disruptions.

Furthermore, Azure Blob Storage offers a cost-effective pricing model, making it an ideal choice for storing massive datasets. Its flexible and tiered pricing structure allows for optimized cost management by aligning storage costs with actual usage patterns. By leveraging this pricing model, organizations can effectively manage storage costs while benefiting from the ability to store and access large volumes of data.

Moreover, Azure Blob Storage seamlessly integrates with other Azure services, facilitating streamlined data processing and analysis pipelines. Its integration capabilities enable effortless data transfer and interoperability with various Azure tools and services, empowering users to leverage a comprehensive ecosystem for advanced data processing, analytics, and insights generation.

By utilizing Azure Blob Storage as the storage solution for both detected and original images, the system leverages its inherent scalability, durability, and cost-efficiency. The seamless integration with other Azure services further enhances the system's overall efficiency, reliability, and analytical capabilities. This comprehensive storage solution contributes to the seamless operation and optimal performance of the system, ensuring reliable and efficient management of the images throughout the entire workflow

3.3.4 Active Learning

The collected detected and original images are utilized for active learning, which falls under the category of human-in-the-loop machine learning strategies employed to enhance the performance of the trained model. Active learning leverages the iterative process of iteratively selecting, labeling and incorporating new data into the model training process. As IoT devices and other systems consistently upload images to the model, an increasing number of images containing the objects of interest and representing the actual deployment environment are obtained. These images serve as valuable training data to improve the model's accuracy and robustness.

After the collected images are labeled and incorporated into the training set, the model undergoes re-training to adapt to the evolving data distribution. This iterative process helps the model generalize better to real-world scenarios and capture the intricacies and variations of the objects of interest within its deployment environment. By continuously updating the training data and re-training the model, its performance steadily improves, enabling more accurate and reliable predictions.

Furthermore, users have the opportunity to actively participate in the refinement of the detection model. Through feedback mechanisms integrated into the application, users can provide annotations, corrections, or ratings on the model's predictions. This user feedback serves as a valuable source of information to verify and fine-tune the model's outputs. By incorporating user feedback, the detection model can adapt to user-specific preferences and optimize its performance for the target application domain.

The combination of active learning with user feedback creates a powerful synergy in refining and enhancing the trained model's capabilities. The iterative collection of new data, labeling, re-training, and user feedback foster an ongoing feedback loop that continuously improves the model's accuracy, adaptability, and overall performance in real-world deployment scenarios.

3.3.5 Power BI Dashboard

The customer can access comprehensive views of the fire analysis results through the Power BI dashboard, as depicted in Figure 37. Power BI is a powerful business intelligence tool that allows for the creation of interactive and visually appealing dashboards. The dashboard, designed specifically for fire analysis, showcases a range of visual components that provide valuable insights. These components include the original image, the corresponding infrared image, an overlay of the original and infrared images, a video captured by the IoT device, the detected image generated by the deployed model, and a customer-specific 3D model.

The Power BI dashboard presents a user-friendly interface, enabling customers to intuitively navigate and explore the fire analysis results. With its interactive nature, customers can interact with the visualizations, zoom in on specific details, and drill down into specific aspects of the fire analysis. This interactivity fosters a deeper understanding of the data and facilitates data-driven decision-making.

One of the key advantages of utilizing the Power BI dashboard is its ability to consolidate diverse data sources and present them in a cohesive and visually appealing manner. By integrating data from various sources, such as the original image, infrared image, video feeds, and detected image, the dashboard provides a comprehensive overview of the fire analysis process. This consolidation allows customers to gain a holistic understanding of the analyzed data and draw meaningful insights.

Additionally, the Power BI dashboard offers real-time data updates and dynamic visualizations. This feature ensures that customers have access to the most up-to-date fire analysis results, allowing for timely decision-making and response. The dynamic nature of the visualizations provides a dynamic and engaging experience, enhancing the overall impact and usability of the dashboard.

Moreover, the Power BI dashboard provides the flexibility to tailor the visualizations and layout to meet specific customer requirements. This customization capability enables customers to focus on the most relevant aspects of the fire analysis and align the dashboard with their unique needs.

In summary, the Power BI dashboard serves as a powerful tool for presenting and analyzing fire analysis results. Its interactive and visually appealing nature, consolidation of diverse data sources, real-time updates, and customization capabilities provide significant advantages in facilitating data exploration, decision-making, and understanding within the fire analysis domain.

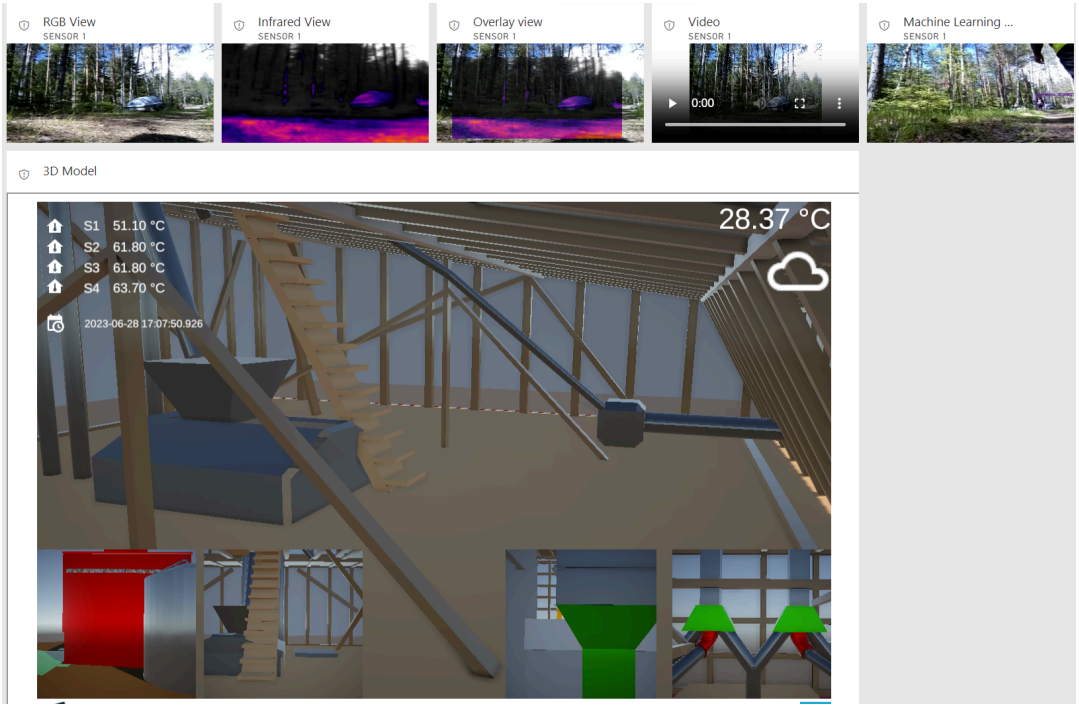


Figure 37: The power bi dashboard. The dashboard presents a comprehensive view of the fire analysis results, displaying various visual components including the original image, the corresponding infrared image, an overlay of the original image and infrared image, a video captured by the IoT device, the detected image generated by the deployed model and a 3D model of the environment.

4 Conclusion

4.1 Summary

In this work, we have made significant contributions to the field of fire detection. Our introduction of a large-scale fire dataset, which surpasses existing datasets in terms of both scale and diversity, marks a major advancement in fire detection research. With over three times the number of images compared to the previously largest dataset, our dataset provides an extensive and diverse collection of fire-related data for training and evaluating fire detection models. The inclusion of fire videos in our dataset adds an additional dimension of realism and complexity, making it a valuable resource for benchmarking and advancing the state-of-the-art in fire detection.

Furthermore, we have leveraged the state-of-the-art Yolov5 model as the foundation of our fire detection system. Through rigorous training on our collected dataset, the proposed model has demonstrated remarkable effectiveness in detecting fires across various real-world scenarios. The evaluation results illustrate the model's robust performance, with an average F1 score of 0.77 and an mAP@0.5 score of approximately 0.77. These metrics highlight the model's ability to accurately detect fires and its reliability even at lower confidence levels. The exceptional performance of our model positions it as a powerful tool in fire detection, capable of providing timely and accurate alerts for potential fire incidents.

Additionally, our research has expanded beyond model development by focusing on the deployment of the trained model to the cloud. This strategic deployment approach offers several notable advantages, including enhanced scalability and accessibility of our fire detection system. By utilizing cloud infrastructure, our system can handle large volumes of data efficiently and accommodate a growing number of users and devices. Furthermore, the cloud-based deployment ensures widespread access to the fire detection system, enabling seamless integration with Internet of Things (IoT) devices and facilitating real-time fire monitoring capabilities. The cloud deployment aspect of our work not only enhances the practicality and effectiveness of our system but also opens up possibilities for future advancements and integration with other smart technologies.

In conclusion, our research has made significant contributions to the field of fire detection through the introduction of a large-scale and diverse fire dataset, the development of an effective Yolov5-based fire detection model, and the deployment of the model to the cloud. These advancements collectively establish a comprehensive and reliable fire detection system that holds immense potential in enhancing fire safety measures and mitigating the devastating impact of fire accidents. Moving forward, continued research and development efforts in this domain can further refine and improve the performance and applicability of fire detection systems, ultimately contributing to the protection of lives and property in our society.

4.2 Future work

1) Completing the annotation of the segmentation dataset, with the assistance of the Segment Anything Model. This model, specifically designed for annotating segmentation datasets, provides advanced tools and techniques to streamline the annotation process. By leveraging the capabilities of the Segment Anything Model, we aim to ensure accurate and comprehensive annotations, covering various segmentation classes and encompassing a diverse range of scenarios. Following the completion of dataset annotation, the next step will involve training segmentation models using state-of-the-art algorithms and architectures. This training process will employ advanced deep learning techniques, such as convolutional neural networks and attention mechanisms, to learn intricate patterns and relationships within the data. After training, a thorough comparison of the performance and accuracy of different segmentation models will be conducted, incorporating evaluation metrics such as intersection over union (IoU) and pixel-level accuracy. This comparative analysis will provide insights into the strengths and weaknesses of various segmentation models, ultimately guiding the selection of the most effective approach for enhancing accuracy and performance in fire analysis.

2) Although this master thesis utilized the yolov5 model, it is worth noting that the yolov8 model has recently been published, building upon the remarkable success of its predecessors. The yolov8 model introduces new features and improvements that significantly enhance performance and flexibility in instance segmentation tasks. By adopting the yolov8 model, future work aims to leverage its advancements to further boost the accuracy and efficiency of fire detection. The incorporation of yolov8 will involve adapting the existing training pipeline and fine-tuning the model on the available fire dataset. The subsequent evaluation will encompass rigorous comparisons between yolov8 and yolov5, taking into account metrics such as mean average precision (mAP), recall, and precision. Through this comparative analysis, it will be possible to identify the model that best addresses the challenges specific to fire detection, ultimately guiding the selection of the most suitable model for accurate fire analysis.

3) While the current detection model exhibits strong performance in identifying static objects with fixed shapes, it encounters inherent challenges when it comes to detecting fire. Fire possesses variable shapes and exhibits dynamic behavior, evolving over time. These dynamic characteristics cannot be effectively described or captured through a single static image. Consequently, a shift towards video understanding becomes imperative for accurate fire detection. Video understanding leverages the temporal nature of video data to capture and analyze the dynamic behavior of fire, providing a more robust and comprehensive approach. For future work, the focus will be on training a fire detection model specifically designed for fire video understanding. This will involve curating and augmenting a comprehensive dataset of fire videos, encompassing a diverse range of fire scenarios and environmental conditions. The training process will encompass advanced techniques such as temporal convolutional networks and recurrent neural networks, allowing the model to capture temporal dependencies and detect fire patterns across video frames. The evaluation of the fire

video understanding model will involve rigorous testing and validation, measuring metrics such as frame-level accuracy, video-level precision, and recall. Through this approach, future work aims to harness the power of video understanding to enhance fire detection accuracy, paving the way for more effective fire analysis in real-world scenarios.

References

[1] Sina. The Investigation Report of Jilin Liaoyuan Central Hospital Especially Serious Fire Accident. Available online: https://news.sina.com.cn/c/2006-12-21/102811850924.shtml (accessed on 28 April 2022).

[2] Shenzhen Emergency Management Bureau. Investigation Report of "Feb. 23" Fire Accident in Hangcheng Street, Bao'an District, Shenzhen City. Available online: http://yjgl.sz.gov.cn/yjgl/zhjg/sgdc/content/post_7756942.html (accessed on 28 April 2022).

[3] FEMA, Statistics (2022) *U.S. Fire Administration*. Available at: https://www.usfa.fema.gov/statistics/ (Accessed: November 2, 2022).

[4] Hu, Z., Zhang, J., Shen, R. and Li, S., 2022. Design and implementation of smart home control system. In ITM *Web of Conferences* (Vol. 47, p. 01023). EDP Sciences.

[5] Baalisampang, T., Saliba, E., Salehi, F., Garaniya, V. and Chen, L., 2021. Optimisation of smoke extraction system in fire scenarios using CFD modelling. *Process Safety and Environmental Protection, 149*, pp.508-517.

[6] Morris, J. (1995). Computer vision in robotics. Computer Vision in Robotics, 1-20.

[7] Phillips III, W., Shah, M., Lobo, N.V,:"Flame recognition in video", *Pattern Recognition Lett. 23* (1–3), 319–327, 2002

[8] 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, pp. 1707–1710, IEEE, 2004.

[9] T. C¸ elik, H. Ozkaramanlı, and H. Demirel, "Fire and smoke detection without sensors: Image processing based approach," in *2007 15th European Signal Processing Conference,* pp. 1794–1798, IEEE, 2007.

[10] Töreyin, B.U., Dedeoğlu, Y., Çetin, A.E.,"Flame detection in video using hidden Markov models", *Proc. IEEE Internat. Conf. on Image Processing*, pp. 1230-1233, 2005.

[11] S. Rinsurongkawong, M. Ekpanyapong, and M. N. Dailey, "Fire detection for early fire alarm based on optical flow video processing," in *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 1–4, IEEE, 2012.

[12] M. Mueller, P. Karasev, I. Kolesov, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," *IEEE Transactions on image processing*, vol. 22, no. 7, pp. 2786–2797, 2013.

[13] I. Mobin, M. Abid-Ar-Rafi, M. N. Islam, and M. R. Hasan, "An intelligent fire detection and mitigation system safe from fire (sff)," *Int. J. Comput. Appl,* vol. 133, no. 6, pp. 1–7, 2016.

[14] Q. Zhang, J. Xu, L. Xu, and H. Guo, "Deep convolutional neural networks for forest fire detection," in 2016 *International Forum on Management, Education and Information Technology Application*, Atlantis Press, 2016.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[16] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in *International Conference on Engineering Applications of Neural Networks*, pp. 183–193, Springer, 2017.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[19] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," *Neurocomputing*, vol. 288, pp. 30–42, 2018.

[20] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep cnn-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–16, 2018.

[21] 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. 18174–18183, 2018.

[22] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik,"Efficient fire detection for uncertain surveillance environment," *IEEE Transactions on Industrial Informatics*, 2019.

[23] Olafenwa Moses. (n.d.). FireNET. GitHub. https://github.com/OlafenwaMoses/FireNET

[24] Ritupande. (n.d.). Fire Detection from CCTV. Kaggle. https://www.kaggle.com/datasets/ritupande/fire-detection-from-cctv

[25] CAIR lab. (n.d.). Fire-Detection-Image-Dataset. GitHub. https://github.com/cair/Fire-Detection-Image-Dataset

[26] phylake1337. (n.d.). Fire Dataset. Kaggle. https://www.kaggle.com/phylake1337/fire-dataset

[27] Roboflow. (n.d.). Wildfire Smoke Object Detection. Roboflow. https://public.roboflow.com/object-detection/wildfire-smoke

[28] DataCluster Labs. (n.d.). Fire and Smoke Dataset. Kaggle. https://www.kaggle.com/datasets/dataclusterlabs/fire-and-smoke-dataset

[29] DataCluster Labs. (n.d.). Domestic-Fire-and-Smoke-Dataset. GitHub. https://github.com/datacluster-labs/Domestic-Fire-and-Smoke-Dataset

[30] Dataset Credits: Baris Dincer (2021), Wildfire Detection Image Data For Machine Learning Process, Kaggle, V1, https://www.kaggle.com/brsdincer/wildfire-detection-image-data, Database Contents License (DbCL) v1.0

[31] Pedro Vinícius Almeida Borges de Venâncio, Adriano Chaves Lisboa, Adriano Vilela Barbosa: An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. In: Neural Computing and Applications, 2022.

[32] M. Frucci, C. Sansone, D. Sanniti di Baja, A. Saggese, and G. Ventre, "Fire Detection Dataset," MIVIA Lab, University of Salerno, Italy, Available: https://mivia.unisa.it/datasets/video-analysis-datasets/fire-detection-dataset/

[33] Roboflow. "Roboflow: Computer Vision Made Easy." Accessed June 27, 2023, available at: https://roboflow.com/.

[34] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... & Girshick, R. (2023). Segment anything. *arXiv preprint arXiv*:2304.02643.

[35] Ultralytics. (2021). *YOLOv5: Detect Objects from Images and Video*. [Source code]. GitHub. Retrieved from https://github.com/ultralytics/yolov5

[36] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[37] Liu, H., Sun, F., Gu, J.,& Deng, L. (2022). SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. *Sensors, 22*(15), 5817. https://doi.org/10.3390/s22155817

[38] Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 3, pp. 850–855.

[39] Wang, Chien-Yao, et al. "CSPNet: A new backbone that can enhance learning capability of CNN." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020.

[40] Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[41] Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[42] He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015): 1904-1916.

[43] Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

[44] Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.

[45] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[46] Daniel, Eldan R. "Wildfire Smoke Detection with Computer Vision." arXiv preprint arXiv:2301.05070 (2023).