Enhancing Fire Detection in Images using Faster R-CNN with Gaussian Filtering and Contrast Adjustment

Dwiki Lazzaro*, Febryanti Sthevanie, Kurniawan Nur Ramadhani

School of Computing, Informatics, Telkom University, Bandung, Indonesia
Email: 1subadrun@student.telkomuniversity.ac.id, 2sthevanie@telkomuniversity.ac.id,
3kurniawann@telkomuniversity.ac.id
Correspondence Author Email: subadrun@student.telkomuniversity.ac.id

Abstract—A system is designed with an accurate and efficient model to detect fires, aiming to assist in fire prevention. Designing such a system poses a challenging task, as numerous aspects need to be considered, including model accuracy, parameter count, computational complexity, and more. Therefore, the research will incorporate techniques such as Image Smoothing Filtering and Contrast Adjustment to enhance the fire detection process. The primary objective is to develop a robust system that can effectively identify and detect fire occurrences. Accuracy is crucial to ensure reliable results, while efficiency plays a significant role in real-time fire detection. By implementing Image Smoothing Filtering, the system can reduce noise and enhance image quality, improving detection performance. Contrast Adjustment techniques will further contribute to the system's efficiency by emphasizing fire patterns and enhancing their visibility. The system's design encompasses careful consideration of various factors to strike a balance between accuracy, efficiency, and computational complexity. By utilizing Image Smoothing Filtering and Contrast Adjustment, the research aims to develop a comprehensive fire detection system that can aid in preventing fire incidents. This study endeavors to contribute to the advancement of fire detection technologies and pave the way for future innovations in this field.

Keywords: Image Filtering; Object Detection; CNN; Faster R-CNN

1. INTRODUCTION

There are numerous factors that can lead to fires in the surrounding environment, causing extensive damage and significantly impacting both life and activities within that area [1], [2]. Effective and timely detection and prevention of fires can help mitigate these losses. Hence, the need for a fast and accurate fire detection system is crucial in preventing fire disasters in specific regions [3].

Fire detection is a critical field that has yet to be comprehensively explored. It relies on the challenge of differentiating images based on color, shape, size, and other characteristics [1]. While humans can easily identify fire hazards, fires often emerge unexpectedly from various unforeseen sources, making early detection challenging for many individuals [3]. Even seemingly straightforward scenarios, such as fires caused by cigarette butts, can prove difficult to detect rapidly and accurately.

To address this issue, a robust system leveraging fire patterns will be developed for fire detection. Prior research, an innovative fire recognition algorithm was introduced, showcasing a remarkable blend of accuracy and efficiency for intelligent monitoring systems. This algorithm exhibits versatility and achieves superior recognition rates across diverse fire scenarios. By employing a low-complexity feature detection technique, the algorithm effectively filters fire images, leading to enhanced accuracy in fire identification. Furthermore, ongoing investigations aim to extend the algorithm's capabilities to accurately detect and filter smoke in complex environments, enabling more effective warnings during fire incidents. This research contributes to the advancement of fire detection methodologies, emphasizing the importance of accuracy and efficiency in addressing fire-related challenges[4], [5].

Saeed F et al. present an Adaboost MLP model for fire forecasting. Trained using sensor data and validated with temperature samples, the model consists of multiple MLP models boosted by Adaboost. It achieves improved temperature prediction accuracy. Adaboost-LBP and CNN models are also introduced for fire and smoke detection, surpassing traditional CNN models in accuracy. The Adaboost-LBP model captures fire regions of interest (RoIs), while the CNN model better classifies these RoIs [3].

Barmpoutis P et al. propose a novel fire detection approach combining deep learning and spatial texture analysis. It employs a Faster R-CNN network for candidate region detection and uses VLAD encoding to differentiate fire-colored objects from the actual fire. The approach exhibits high true positive rates and reduced false positives when evaluated on fire images and objects. Future work involves expanding the dataset and extending the approach to detect fire in video sequences [6].

Xiaowei W et al. present a novel segmentation method for apple recognition is introduced, combining a modified Gaussian kernel function with a convolutional neural network (CNN). The improved Gaussian kernel function enables rapid segmentation of regions from large and complex areas. Experimental results demonstrate that the proposed method outperforms other existing methods, achieving accurate identification of small and medium-sized apple images with excellent real-time performance. Furthermore, future work will focus on enhancing the method by incorporating deep learning techniques and applying it to practical engineering applications. This research paves the way for further advancements in Apple recognition and contributes to the development of more robust and efficient segmentation algorithms [7].
Abdusalomov et al. present a new approach to identifying and isolating shadow pixels from moving objects. The method includes techniques such as enhancing image contrast, subtracting background information, removing shadows using a geometry-based method, reducing noise, and filling gaps in the moving object mask. This approach successfully tackles issues like ghosting artifacts and the similarity between the background and the current image. The researchers propose future improvements by integrating a convolutional neural network (CNN) to handle challenging shadows in outdoor environments. The ultimate goal is to implement this strategy in smart cities to accurately recognize the shapes of moving objects and track them effectively [8].

This research will utilize a dataset consisting of fire images and employ Convolutional Neural Networks (CNNs) with the Faster R-CNN architecture. Prior to the detection process, the images will undergo thorough processing, including techniques such as Image Smoothing Filtering and Contrast Adjustment. These preprocessing steps aim to enhance the system's overall efficiency, reduce noise, and improve the accuracy of fire detection [6], [9].

The incorporation of image processing techniques is essential to optimize time and cost efficiency during the system's implementation. Additionally, it aims to yield high accuracy in fire detection results [6]. This research endeavors to contribute to the rapid development of fire detection studies and foster the creation of new systems or methods, particularly in the domain of fire detection.

By successfully developing a reliable and efficient fire detection system, the potential impact of fire disasters can be minimized, protecting lives, property, and the environment. The outcomes of this research have the potential to inspire further investigations and advancements in fire detection technologies, ultimately enhancing overall safety and security against fire hazards.

2. RESEARCH METHODOLOGY

2.1 System Design

In the given diagram at Figure 1, we can visualize the systematic flow of our designed system. The first step involves annotating all the images in our possession to create a dataset. This dataset serves as the foundation for our subsequent processes. Moving on to the preprocessing phase, we encounter two distinct processes: Gaussian and contrast adjustment. These processes operate independently until completion. Each process contributes to refining and enhancing the quality of our images. Once both processes are completed, we move on to the next stage. At this point, we partition the data into separate training and testing sets. The training data, which constitutes a portion of our dataset, is then used to train the Faster R-CNN model. This training enables the model to learn and generate prediction values based on the provided data. After training the model, we proceed to compare the predicted results with the test data. This comparison allows us to evaluate the model's performance and accuracy. We employ the F1-Score methodology for this evaluation, which provides a comprehensive measure of precision and recall. By following this systematic approach of dataset annotation, preprocessing, model training, and evaluation using the F1-Score, we can effectively design and optimize our system for accurate predictions.

2.2 Image Detection

Image detection is the process of classifying digital pictures into related groups using their input. In this complicated process, which is affected by many variables, accuracy and speed are essential measures. It is difficult to achieve high accuracy and quick processing speed in image recognition. It necessitates choosing appropriate algorithms, improving picture quality through preprocessing methods, and making use of human experience. The goal of this study is to create an image detecting system that is very accurate and fast. The objective is to improve accuracy and processing speed by experimenting with different algorithms and taking work requirements into account. The findings will help advance image identification technologies and spur advancements in pattern recognition and computer vision. The goal of this study is to expand picture detecting skills while overcoming obstacles [10].

Dwika Lazzaro, Copyright © 2023, MIB, Page 1573
Submitted: 30/06/2023; Accepted: 26/07/2023; Published: 31/07/2023
2.2 Image Filtering

The real operation of the devices necessarily creates internal and external disturbances throughout the process of picture capture, transmission, reception, and processing, leading to various forms of noise. When a fire erupts, the photographs that are acquired are occasionally impacted by noise that is created by external elements like weather and sunlight. As a result, prior to performing fire detection, preprocessing techniques like picture smoothing and filtering are required. Mean filtering, median filtering, and Gaussian filtering are often employed techniques for this purpose [11]. The goal of this work is to improve fire detection reliability and accuracy by applying picture smoothing and filtering techniques to the fire images. Prior to undertaking fire detection, these strategies are essential for lowering noise and enhancing image quality.

2.3 Gaussian Filtering

Gaussian Filtering is a Low Pass Filter used to reduce noise (high-frequency components) and blur areas in an image. The filter is implemented as a symmetric kernel of odd size that is applied to each desired pixel to achieve the desired effect [12]. The 2D Gaussian Filter can be described by the following equation:

\[ G(x, y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}} \]  

In this equation, \( G(x, y) \) represents the Gaussian kernel at position \((x, y)\), \( \sigma \) is the standard deviation controlling the amount of blurring, and \( e \) denotes the exponential function. The Gaussian filter works by convolving the image with this kernel, resulting in a weighted average of neighboring pixels. The Gaussian filter is effective in reducing noise while preserving the overall structure and important features of the image. It smooths out the image by attenuating high-frequency details, thereby achieving a more visually pleasing and noise-free result [12]. In our study, the Gaussian filter is applied as a preprocessing step to enhance the quality of fire images before conducting fire detection algorithms. By reducing noise and blurring unwanted artifacts, the Gaussian filter helps to improve the accuracy and reliability of the subsequent fire detection process [13].

![Figure 2. Comparison between basic image (a) with gaussian filtered image (b)](image)

The performance of the Gaussian filter is evaluated using appropriate metrics and compared with other filtering techniques to determine its effectiveness in enhancing fire detection results. Through experimental analysis, we aim to demonstrate the benefits of incorporating the Gaussian filter into the fire detection pipeline and highlight its role in achieving more accurate and robust fire detection outcomes [14].

2.4 Contrast Adjustment

Contrast adjustment in image processing involves enhancing or modifying the contrast of an image to improve its visual quality and make details more distinguishable. Contrast refers to the difference in pixel intensity values between the brightest and darkest areas of an image. A high-contrast image has well-defined details, while a low-contrast image appears dull with less distinguishable details [15]. In digital images, pixel intensities are represented by numerical values ranging from 0 (black) to 255 (white) in an 8-bit grayscale image. The contrast of an image determines how wide or narrow this range of intensities is [9]. A high-contrast image has a large difference between the brightest and darkest areas, resulting in clear and well-defined details. In contrast, a low-contrast image has a limited difference in intensities, making it appear dull or washed out with less distinguishable details.

![Figure 3. Comparison between basic image (a) with contrast adjustment image (b)](image)
The goal of contrast adjustment is to enhance the visibility and improve the overall quality of an image by expanding or compressing the range of intensities. By adjusting the contrast, the visual separation between objects, textures, and structures within an image can be increased, leading to improved image interpretation and analysis [16].

2.5 Faster R-CNN

Faster R-CNN is an advanced object detection algorithm that provides significant improvements over previous methods in accurately and efficiently detecting objects in images [14], [17]. The key concept behind Faster R-CNN is the integration of a Region Proposal Network (RPN) into the network architecture. This network plays a crucial role in generating region proposals, which are candidate bounding boxes for objects in an image. These proposals are then utilized for the object detection process. The architecture of Faster R-CNN comprises two primary components: the RPN and the subsequent Fast R-CNN network. The RPN and Fast R-CNN share convolutional features, leading to efficient computation. The RPN generates region proposals based on predefined anchor boxes and predicts the probability of object presence and the adjustments needed for accurate bounding box coordinates. The Fast R-CNN network takes the generated region proposals and performs object classification and bounding box regression. It utilizes a region of interest pooling layer to align the features of each proposal for further processing. These features are then fed into fully connected layers for object classification and fine-tuning of the bounding box coordinates.

Figure 4. The Faster R-CNN methodology

By leveraging shared features and the integration of the RPN, Faster R-CNN achieves both high accuracy and improved speed compared to previous approaches. It effectively eliminates redundant computations and enables end-to-end training, leading to superior object detection performance [18].

2.6 SSD MobileNet V1

SSD MobileNet V1 is a specific object detection model that combines two important components: the Single Shot Multibox Detector (SSD) and the MobileNet architecture. This model is specifically designed for real-time object detection on devices with limited computational resources, such as mobile phones and embedded systems.

The SSD part of the model refers to the object detection algorithm itself. It is an efficient framework that can detect objects in images using a single pass through a convolutional neural network (CNN). Unlike other methods that involve multiple stages or complex post-processing steps, SSD directly predicts object categories and their bounding box coordinates in a single step [19].

MobileNet, on the other hand, is a lightweight CNN architecture created specifically for devices with limited resources. It achieves a balance between model size and accuracy by employing depth wise separable convolutions, which reduce computational complexity while maintaining good performance.

Figure 5. The SSD MobileNet methodology
By combining SSD with the MobileNet architecture, SSD MobileNet V1 offers real-time object detection capabilities on devices with restricted computational power. This makes it well-suited for applications that require fast and efficient object detection, such as real-time video analysis, robotics, and mobile applications [14].

2.7 Dataset

By In this research project, the classification task will involve utilizing a dataset composed of jpg images. The dataset contains a total of 748 images, with 598 images assigned for training purposes and 150 images designated for testing. The dataset exclusively focuses on a single class, which is “Fire.” All the images used in this research have been acquired from the following source: https://www.kaggle.com/datasets/phylake1337/fire-dataset. Included below are a few sample images from the dataset that will be utilized in this study.

![Sample Images](a) (b) (c) (d) (e) (f)

**Figure 6.** From image (a) to (c) is sample data for training and image (d) to (f) is sample data use for testing

2.8 Evaluation Metrics

When it comes to fire detection, the process yields four possible outcomes that depict the results of the detection task. These outcomes are crucial in assessing the accuracy and effectiveness of the detection system. Let’s explore each outcome in more detail:

The first outcome is a True Positive (TP). This occurs when an image containing fire is correctly identified as such. In other words, the detection system accurately detects the presence of fire in the image. The true positive result signifies the system’s ability to correctly recognize fire instances, which is a desirable outcome for effective fire detection [20].

On the other hand, we have the False Negative (FN). This outcome occurs when an image containing fire is mistakenly classified as non-fire. This means that the detection system fails to identify the fire present in the image. A false negative result indicates a missed detection, which can be problematic as it may lead to delayed or inadequate response measures in fire-related incidents [20].

Moving on, we have the True Negative (TN). This outcome arises when an image without any fire is correctly identified as non-fire. The detection system correctly recognizes that there is no fire present in the image. The true negative result reflects the system’s capability to accurately distinguish non-fire instances, which helps minimize false alarms and unnecessary interventions [20].

Lastly, we have the False Positive (FP). This outcome occurs when an image without any fire is incorrectly classified as containing fire. This means that the detection system falsely indicates the presence of fire in the image. False positives can lead to unnecessary panic, resource wastage, and disruption of regular operations if frequent false alarms are triggered [20].
Table 1. Confusion Matrix

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True Positive (TP)</td>
<td>True Negative (TN)</td>
</tr>
<tr>
<td>False</td>
<td>False Positive (FP)</td>
<td>False Negative (FN)</td>
</tr>
</tbody>
</table>

By considering these four possible outcomes, fire detection systems can be evaluated and fine-tuned to improve their accuracy, minimize missed detections, reduce false alarms, and enhance overall reliability.

In object detection, several common evaluation metrics are used: precision, recall, accuracy, and F1 score. Precision measures the accuracy of object identification, while recall captures the ability to detect relevant objects. Accuracy represents the overall correctness of the system, and the F1 score combines precision and recall for a balanced assessment. These metrics help assess the strengths and weaknesses of object detection algorithms, providing insights into system performance and effectiveness [20].

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{2}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{3}
\]

\[
\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}
\]

\[
F1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}
\]

3. RESULT AND DISCUSSION

3.1 System Requirements

We implemented our proposed method using the Python programming language, leveraging its robust ecosystem and extensive libraries for machine learning and computer vision. The training and testing processes of our model were performed in Google Colab, a cloud-based environment that offers powerful GPU resources, specifically an Nvidia V100 16GB GPU, to accelerate the computations involved in training deep learning models.

3.2 Preprocess

To ensure the validity and efficacy of our proposed method, we conducted a comprehensive evaluation and comparison with other existing models. Before diving into the training phase, we carefully preprocessed our dataset, which consisted of annotated images in XML format.

![Figure 7](image.png)

(a) (b)

Figure 7. Image from dataset (a), the XML annotation (b)

We unified these images and annotations, converting them into a file format suitable for training called a record file. This step facilitated the subsequent training process and ensured that our model could effectively learn from the annotated data.

Before creating a record file, we need to combine two dataset components into a single CSV file. This file will contain the image names and corresponding label information from the annotations, all organized in a convenient table format.

Table 2. CSV Format

<table>
<thead>
<tr>
<th>Filename</th>
<th>Width</th>
<th>Height</th>
<th>Class</th>
<th>XMin</th>
<th>YMin</th>
<th>XMax</th>
<th>YMax</th>
</tr>
</thead>
<tbody>
<tr>
<td>00ccfb25-fire_220.jpg</td>
<td>792</td>
<td>700</td>
<td>Fire</td>
<td>218</td>
<td>0</td>
<td>581</td>
<td>534</td>
</tr>
</tbody>
</table>

Table 2 presents the structured format of the merged CSV data, showcasing the combination of image names with the corresponding annotation coordinates we possess.
3.3 Pure Image Training Process

In the initial phase, we employed the Faster R-CNN algorithm as the core architecture for our model. We utilized a pure dataset without any additional preprocessing techniques such as Gaussian filtering or contrast adjustment. This approach allowed us to establish a baseline performance for our model, serving as a reference point for evaluating the impact of subsequent modifications.

For the training process, we configured a batch size of 4, ensuring a balance between computational efficiency and model convergence. To sufficiently train our model, we performed 10,000 steps, iterating through the dataset in batches and adjusting the model's parameters to minimize the loss function. Following the training phase, we conducted an evaluation process to assess the model's performance on unseen data. This evaluation consisted of 5,000 steps, with each step processing a batch of data. By evaluating the model on a separate dataset, we aimed to gauge its generalization capabilities and obtain accurate performance metrics.

During the training warm-up phase, we set the learning rate to 0.4. This relatively high learning rate initially allowed the model to quickly explore the parameter space and accelerate convergence. As the training progressed, we gradually adjusted the learning rate to 0.0133, a smaller value that facilitated fine-grained optimization and helped the model converge to a more optimal solution.

Upon completing the training and evaluation phases, we obtained promising results. The evaluation metrics revealed an F1-Score of 83.1%, indicating a high level of accuracy and effectiveness in detecting and classifying objects of interest in the images. This F1-Score represents the harmonic mean of precision and recall, providing a balanced measure of our model's performance.

3.4 Gaussian Filtered Image Faster R-CNN

Motivated by the desire to further enhance the performance of our model, we introduced an additional preprocessing step: Gaussian filtering. By applying a 3x3 kernel-based Gaussian filter to the images, we aimed to reduce noise and enhance relevant features related to fire detection. The resulting filtered images served as the input data for the subsequent training and evaluation processes with the updated Faster R-CNN model.

Figure 8. From (a) to (c) are pure image dataset for training Faster R-CNN.

Figure 9. Evaluation results as illustrated in (a), (b), and (c) from pure image datasets and its bounding box prediction.

Figure 10. From (a) to (c) are filtered image datasets by gaussian effect with 3x3 kernel.
Remarkably, incorporating Gaussian filtering into our pipeline yielded significant improvements. The F1-Score obtained after training and evaluating the updated model on the filtered images reached an impressive value of 93.7%. This substantial increase in performance demonstrated the efficacy of Gaussian filtering as a preprocessing technique, highlighting its ability to enhance the model's ability to detect and classify fire-related objects accurately.

![Figure 11](image1.png)

**Figure 11.** On illustration at (a), (b), (c) are evaluation results from gaussian filtered image datasets and its bounding box prediction.

### 3.5 Image Contrast Adjustment Faster R-CNN

Buoyed by the success of Gaussian filtering, we decided to explore the potential benefits of contrast adjustment, another widely used image enhancement technique. By modifying the contrast values of the images, we aimed to amplify the salient features associated with fire, further empowering the model's discriminatory abilities. However, contrary to our expectations, the results obtained from this experiment were not as promising.

![Figure 12](image2.png)

**Figure 12.** From (a) to (c) are filtered image datasets by contrast adjustment.

The F1-Score achieved after training and testing with the Faster R-CNN model using contrast-adjusted images was 85.7%. Although this score still demonstrated decent performance, it fell short of the improvements attained through.

![Figure 13](image3.png)

**Figure 13.** On illustration at (a), (b), (c) are evaluation results from contrast adjustment image datasets and its bounding box prediction.

In the last phase, we utilize SSD MobileNet as a comparative benchmark to evaluate the performance of our proposed method. Our expectation is that Faster R-CNN, aided by Gaussian filtering and contrast...
adjustment, will demonstrate superior fire detection capabilities. We can observe and validate these anticipated outcomes in Table 3 and Figure 14.

Table 3. Evaluation Results

<table>
<thead>
<tr>
<th>Method</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Image with Faster R-CNN</td>
<td>83.18%</td>
</tr>
<tr>
<td>Gaussian Filtered Image with Faster R-CNN (Proposed)</td>
<td>93.70%</td>
</tr>
<tr>
<td>Contrast Adjustment Image with Faster R-CNN (Proposed)</td>
<td>85.78%</td>
</tr>
<tr>
<td>Pure Image with SSD MobileNet</td>
<td>80.13%</td>
</tr>
<tr>
<td>Gaussian Filtered Image with SSD MobileNet</td>
<td>83.76%</td>
</tr>
<tr>
<td>Contrast Adjustment Image with SSD MobileNet</td>
<td>82.15%</td>
</tr>
</tbody>
</table>

Figure 14. Comparison chart between the proposed method and the comparison method

4. CONCLUSION

In this research we aimed to develop a robust fire detection system using image processing and deep learning. The goal was to improve accuracy and efficiency, reducing the impact of fire disasters on lives and property. The study utilized a dataset of fire images and employed CNNs with the Faster R-CNN architecture for object detection. Preprocessing techniques, including Gaussian filtering and contrast adjustment, were applied to enhance image quality and reduce noise. Gaussian filtering effectively improved fire detection by reducing noise and artifacts. Contrast adjustment, however, did not yield significant improvements and requires further exploration. Evaluation metrics such as precision, recall, accuracy, and F1 score showed high effectiveness in fire detection with a score of 93.7%. The developed system has the potential to inspire advancements in fire detection technology, contributing to safety and security. Limitations include dataset specificity and the need for further research and testing in different contexts. Overall, this research offers a promising approach to fire detection through image processing and deep learning, improving safety and mitigating losses from fire disasters.

ACKNOWLEDGMENT

We extend our heartfelt appreciation to our research partner for their invaluable collaboration and support throughout this project. We are grateful for Telkom University's resources and facilities that aided our research. Our thanks also go to the participants for their valuable contributions. Lastly, we express our deep gratitude to our families and friends for their unwavering support.

REFERENCES


