JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DE-HAZING/DE-SMOKING FOR REPRODUCING REAL TIME INDOOR AREAS UNDER FIRE

[1] K. Archana Bhat, [2]Deepthi S, [3]Divyashree S, [4]Benitta Hathsiyal X, [4]Ananya Bhagavan

[1]Assistant Professor, Department of Computer Science and Engineering, Bangalore, India

[2,3,4,5] UG Student, Department of Computer Science and Engineering, City Engineering College, Bangalore, India

Abstract: The study investigates various de-hazing and de-smoking algorithms, analyzing their effectiveness in enhancing the visibility of fire-related features within obscured environments. De-hazing is the analysis and manipulation of a digitized image, especially in order to improve its quality. The main aim is to solve the problem of low visibility by advanced image processing technique purpose to improve the perceptual quality of images. Exterior images captured in bad weather, such as fog and fog, are not clearly articulated, so they don't make sense. It is caused by smoke, fog and dust that is present in the atmosphere. Such images are reduced in quality, thus reducing variability and visibility.

***Index Terms* – De-smoke, De-haze, Fire and Smoke Detection, Image Processing, Visibility**

I. INTRODUCTION

De-smoke and de-haze technologies have emerged as pivotal tools in enhancing the efficiency of fire detection systems. In the realm of fire prevention and safety, these innovative technologies play a crucial role in mitigating the challenges posed by smoke and haze. De- smoke technology is designed to intelligently filter and eliminate smoke particles from visual data, enabling fire detection systems to operate with greater accuracy and speed as in [7][8]. Similarly, In [15] de- haze technology focuses on reducing the impact of atmospheric haze, which can impede visibility and hinder the timely identification of fires [12]. By incorporating de-smoke and de-haze capabilities into fire detection systems, we can significantly improve the reliability of early fire detection, providing a more robust solution for safeguarding lives and property against the devastating effects of fires. These advancements mark a significant stride in the ongoing pursuit of leveraging cutting-edge technologies to enhance fire safety measures and emergency response protocols.

Furthermore, the integration of de-smoke and de-haze technologies aligns with the broader trend of leveraging artificial intelligence and advanced image processing techniques in safety and security applications. The continuous refinement of these technologies holds the promise of further improving the resilience and accuracy of fire detection systems, contributing to a safer and more secure built environment for communities worldwide. Therefore, removing haze from images is an important and widely demanded topic in computer vision and computer graphics areas [9][11]. The main challenge lies in the ambiguity of the problem. Haze attenuates the light reflected from the scenes, and further blends it with some additive light in the atmosphere. The target of haze removal is to recover the reflected light i.e., the scene colors from the blended light. This problem is mathematically ambiguous. There are an infinite number of solutions. given the blended light. Ambiguity is a common challenge for many computer vision problems. Many de-smoke technologies offer customizable sensitivity levels, allowing users to tailor the system's response to specific environments and potential fire risks [13]. This flexibility ensures that the technology can be fine-tuned to balance the need for early detection with the avoidance of false positives, adapting to the unique characteristics of different settings. This discovery is against conventional theories but we can prove its validity theoretically and experimentally. The second technique is a novel edge-aware filter. It is non-iterative and can be computed in real- time, but still exhibits very high quality. This filter is found to be superior to previous techniques in various edge-aware applications including haze removal. Thus, the state-of-the-art is advanced in a broader area as proposed in [10].

II. OBJECTIVES

The objectives for dehazing and desmoking of real-time video detection aim to enhance visibility, achieve real-time processing, and preserve image details while removing haze and smoke. The goal is to develop algorithms that adapt to varying levels of haze or smoke density, ensuring robustness across different conditions. Efficiency is crucial, requiring low computational cost for real-time performance on standard hardware as in [14]. Additionally, these algorithms should improve object detection and recognition, enhancing scene understanding by removing visual obstructions caused by haze and smoke. User-friendly interfaces for parameter control and seamless integration with other systems are essential. Thorough validation and testing, along with scalability to handle high-resolution videos and large-scale deployments, are also key objectives. Achieving these goals can significantly improve the quality and usability of video surveillance, monitoring, and analysis systems in challenging environmental conditions.

III. RELATED WORK

An end-to-end image-dehazing network is designed based on the CNN architecture [1], which has shown superior performance among neural networks. However, the CNN structure is modified to decrease the computational complexity for real-time image- dehazing, which is the main goal of this study. Motivated to the new development on the deep learning models,

A Fire and smoke detection based on a video camera using YOLOv2 model is proposed [2]. Fire and smoke detection have a higher speed in imaging processing. In such a case, YOLOv2 is the best technique to encounter the detection of these objects. This detection is essential for firefighters because it will give an early alerting sign for fire and smoke accidents to take remedial actions accordingly.

Satellite remote sensing technology is currently the primary means of wildfire monitoring. Its fundamental principle lies in identifying and monitoring fire spots by utilizing the electromagnetic radiation characteristics released during biomass combustion [3]. This technology boasts wide-ranging fire detection, high resolution, and timely response to dynamic changes, particularly playing an increasingly crucial role in the detection of wildfires in landscapes and the prediction of potential hazards.

Single image haze removal based on the improved atmospheric scattering model [4] is fast single image dehazing algorithm. In this algorithm, the transmission via a linear model is estimated that describes the relationship between the transmission and the haze aware density feature. Based on the segmentation strategy and the proposed guided energy model (GEM), one can effectively obtain the scene incident light using the same way as we estimating the atmospheric light.

In [5] proposed method compares several CNN models with different parameter settings for smoke detection in both normal and foggy environments is done. After the comparison of different CNNs, we found that VGG-16 is better than other models i.e., AlexNet and Google Net.

To solve the problems encountered by the traditional image dehazing algorithms in the fire scenario, a novel network built with CNN is proposed in [6]. Unlike the general visual enhancement model, the method is committed to adapting to the image degradation caused by different colors of the haze. The method adopts CNN similar to extract the low-dimensional features of the inputs and then outputs the scene transmittance map t(x).

IV. METHODOLOGY

The methodology involves preprocessing the image data, image enhancement, applying dehaze and de-smoking algorithm and real time implementation.

4.1 Preprocessing

In the initial stages of the preprocessing pipeline, real-time images or video feeds of the indoor area are captured using suitable cameras or sensors. These devices must possess attributes such as high sensitivity to low light conditions and a wide dynamic range to effectively capture the dynamic nature of indoor fire scenarios. Additionally, fast frame rates are essential for ensuring smooth and continuous video feeds, facilitating real-time analysis.

4.2 Image Enhancement

Image enhancement techniques are employed to improve the quality and clarity of the captured images, essential for effective analysis and decision-making in indoor fire scenarios. Contrast enhancement methods are utilized to enhance the distinction between objects and the background, aiding in the identification of critical details within the scene. Adjusting the gamma value modifies the brightness levels, improving the overall appearance of the image and ensuring optimal visibility of important features.

4.3 De-hazing/De-smoking Algorithms

These algorithms form the core of the de-hazing/de-smoking process, crucial for enhancing visibility in indoor fire scenarios. Leveraging the dark channel prior method, haze or smoke is estimated and effectively removed from the image, restoring clarity. Atmospheric light estimation accurately models the haze or smoke present, guiding subsequent processing steps.

4.4 Real-Time Implementation

Efficient real-time implementation strategies are pivotal for timely response in indoor fire scenarios. Parallel processing techniques are harnessed to bolster computational efficiency, ensuring swift image processing and analysis. Leveraging hardware acceleration, such as GPUs or FPGAs, accelerates the execution of intensive tasks, enabling rapid decision-making.

4.5 System Architecture

As shown in Fig. 4.1, The system architecture for indoor fire scenario management integrates real-time image acquisition, preprocessing, object detection using the YOLOv5 model, de-hazing/de-smoking algorithms, and parallel processing for rapid decision-making. Images captured by sensors undergo preprocessing to enhance quality before being fed into the YOLOv5 model for object detection. Detected objects are then processed by de-hazing/de-smoking algorithms to improve visibility in smoke or haze.

4.5.1 YOLO Model

This component represents the YOLO (You Only Look Once) model, which is a deep learning-based object detection system YOLO is utilized in this project for detecting various objects, including fire and smoke, within the video frames.

4.5.2 Video/Image Input

This represents the original, non-hazy input video stream. It serves as the primary source of video data for the system.

4.5.3 Hazy Video/Image Input

This component symbolizes a version of the input video stream that has been affected by haze or smoke. In real-world scenarios, videos captured in environments with fire or smoke may suffer from reduced visibility due to haze, making it challenging to detect objects accurately.

4.5.4 Dehazed Video/Image Output

This depicts the output of the dehazing process applied to the hazy video stream. The dehazing algorithm aims to enhance the visibility of objects within the video frames by reducing the effects of haze or smoke.

4.5.5 Fire and Smoke Detection Algorithm

This component represents the algorithm responsible for detecting fire and smoke within the video streams. It utilizes the YOLO model for object detection and operates on both the original video stream and the dehazed video stream. By analyzing the video frames, this algorithm identifies regions containing fire and smoke, providing valuable information for fire detection and monitoring purposes.

4.5.6 Arduino uno

Implementing window opening functions and rotating water sprinklers on Arduino Uno involves interfacing sensors like temperature, humidity, and light sensors to detect environmental conditions, while actuators such as servo or stepper motors control window operations.

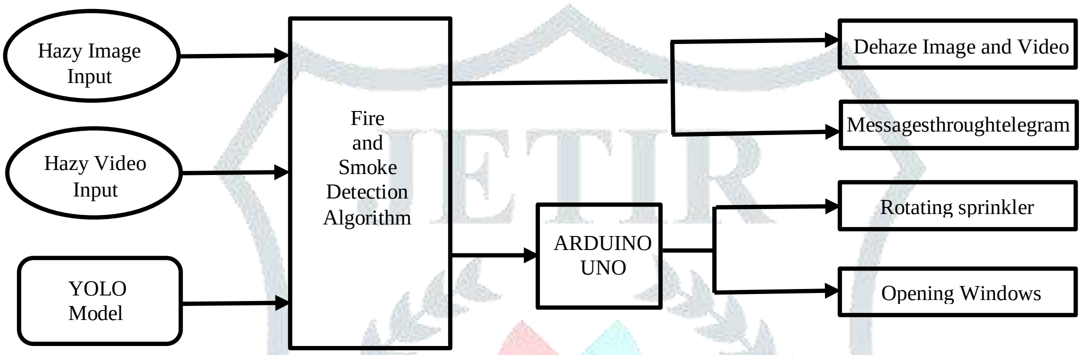

Figure 4.1. System architecture using YOLO model

V. ANALYSIS AND PREDICTIONS

In the integrated framework for indoor fire management with YOLOv5, the analysis and prediction pipeline begins with the acquisition of historical records and real-time sensor data, providing foundational information. Subsequently, data preprocessing and object detection using YOLOv5 enable the identification and localization of fire-related objects in the indoor environment. The application of de-hazing/de-smoking algorithms to the detected regions enhances visibility, crucial for accurate assessment and decision-making. Real-time implementation techniques, including parallel processing and hardware acceleration, ensure swift and efficient processing of data streams.

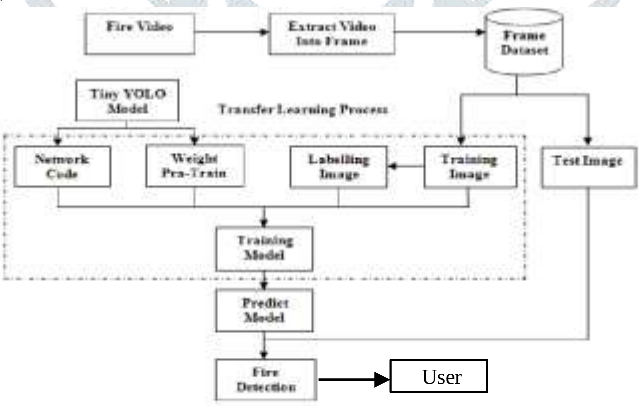

Figure 5.1. General framework for fire detection using YOLO model

Finally, decision support systems analyze the outputs from YOLOv5 and the integrated results from de-hazing/de-smoking, generating comprehensive insights and actionable recommendations for responders. This holistic approach empowers responders with timely and informed decision-making capabilities, optimizing their ability to mitigate risks and manage indoor fire incidents effectively proposed in [14]. Real-time implementation techniques ensure efficient processing. Decision support systems analyze YOLOv5 outputs and integrated results to generate actionable recommendations, facilitating informed decision-making and proactive measures during indoor fire incidents.

VI. IMPLEMENTATION

The implementation of an integrated framework for indoor fire management with YOLOv5 represents a significant advancement in enhancing the response to indoor fire incidents. By combining state-of-the-art object detection capabilities with advanced image processing techniques, this framework aims to provide responders with timely and accurate information to effectively mitigate risks and manage emergencies.

6.1 Dataset

We meticulously curated a diverse array of images depicting various fire scenarios, which served as our comprehensive dataset for training the fire detection system. Each image encapsulates a distinct facet of fire incidents, contributing to the robustness and accuracy of our model. Through this dataset, we aimed to capture the complexity and nuances of real-world fire scenarios, ensuring that our system is adept at detecting fires across different environments and contexts. By leveraging these meticulously selected images, our training process was tailored to enhance the system's performance and reliability in identifying and responding to fire emergencies effectively.

6.2 Components for model implementation

This section gives the details of hardware components used to build the fire detection model. Fig 6.1 shows the hardware components used to build the fire detection model using Arduino uno, DC Motor, Water pump, Sprinkler and LCD Display.

Fig 6.1 Hardware Components of Fire detection model

6.2.1 Arduino uno

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

6.2.2 LCD Display

A 16*2 LCD demonstrates contains two lines likewise; there are 16 characters for each line. Each character is appeared by 5x7 pixel lattice. This LCD includes two registers, specifically, Order and Data. The charges select extra charge bearings that are given to the LCD. A charge is a rule given to LCD to do a predefined errand like presenting it, clears its screen, sets the cursor position. The data enroll saves the data to be appeared on the LCD.

6.2.3 Jumper Wires

A jump wire is an electrical wire or group of them in a cable with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

6.2.4 DC Motor

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic; to periodically change the direction of current flow in part of the motor. DC motor has two wires, we can say them positive terminal and negative terminal, when these wires are connected with power supply the shaft rotates. DC motor has two wires, we can say them positive terminal and negative terminal, when these wires are connected with power supply the shaft rotates. We can reverse the direction of the rotation. L293d chip is very safe to use for DC motor control. This L293D is 16bit chip. Chip is design to control four DC motor, there are two inputs and two outputs for each motor.

6.3 Software Implementation

Implementing the integrated framework for indoor fire management with YOLOv5 on Arduino Uno involves adapting the system to operate within the platform's limited computational resources. Basic environmental data acquisition using compatible sensors interfaced with Arduino Uno is followed by simplified preprocessing steps. Object detection, utilizing lightweight image processing techniques or pre-trained models tailored for microcontrollers, is then implemented.

Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The AT mega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An AT mega16 U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an infield is required. The Arduino Software (IDE) include a serial monitor which allows simple textual data to be sent to and from the board. Rather than requiring a physical press of the reset button before an upload, the Arduino Uno board is designed in a way that allows it to be reset by software running on a connected computer. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar.

6.3.1 Environment Setup

Install necessary software libraries and frameworks, including Python, OpenCV, and PyTorch (for YOLOv5), on the computer system. Configure the development environment to ensure compatibility and smooth operation of the software components.

6.3.2 Data Acquisition

Interface sensors with the platform to collect environmental data such as temperature, smoke density, and gas levels. Utilize sensor libraries and communication protocols compatible with the platform. Utilize sensor libraries and communication protocols (e.g., I2C, SPI) to collect environmental data. Implement error handling and data validation to ensure data integrity.

6.3.3 Object Detection

Use YOLOv5 for object detection in real-time images or video feeds. Utilize pre-trained models or train custom models on a dataset containing indoor fire-related objects. Implement image processing techniques for object detection, such as image resizing and normalization.

6.3.4 De-hazing/De-smoking Algorithms

Develop and integrate de-hazing and de-smoking algorithms in Python to enhance visibility in detected regions affected by smoke or haze. Implement techniques such as dark channel prior, transmission map calculation, and image restoration. Optimize algorithms for efficiency and real-time performance, considering the computational resources of the platform.

6.3.5 Real-Time Implementation

Optimize code for speed and memory usage to ensure real-time processing. Utilize multi-threading or parallel processing techniques to leverage the platform's multi-core capabilities. Profile code and identify bottlenecks for further optimization.

6.3.6 Yolo – Object Detection Algorithm

Deep Learning consists of a very enormous number of neural networks that use the multiple cores of a process of a computer and video processing cards to manage the network's neuron which is categorized as a single node. Deep learning is used in numerous applications because of its popularity especially in the field of medicine and agriculture. Here YOLO deep learning technique is used to identify persons wearing and not wearing face masks. Joseph Redmon et al. introduced You look only once also known as YOLO in 2015. YOLO is a convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural net5 work to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. Some improvements were done over years and YOLOv2 and YOLOv3, v5 versions were introduced respectively in 2016, 2018.The model uses YOLOv5 and it provides good results regarding object classification and detection. In the previous version of Yolov3 Darknet-19 is used.

VII. RESULTS AND DISCUSSION

The implementation of dehazing techniques for images of smoke and fire, coupled with YOLO (You Only Look Once) object detection, along with an Arduino-based window opening mechanism, presents a promising solution for improving visibility and safety in environments affected by smoke and fire. By leveraging advanced image processing algorithms and real-time object detection capabilities, this system can help mitigate the risks associated with fire incidents and aid in timely evacuation efforts.

The integration of Arduino-based window opening and rotating sprinklers mechanisms adds a practical dimension to the system, allowing for automated ventilation in response to detected smoke or fire. By dynamically adjusting ventilation based on real-time environmental conditions, this feature enhances air quality, aids in smoke dispersion, and supports evacuation efforts by providing clearer escape routes. The system's architecture orchestrates a seamless integration of dehazing techniques, YOLO object detection, and Arduino-based window opening and water sprinklers mechanisms to fortify safety measures in environments susceptible to smoke and fire hazards. Fig 7.1 shows the hazed images of fire affected areas caused by smoke and particulate matter resulting from fires in indoor environments. These images capture the obscured visibility and reduced contrast inherent in areas affected by fire incidents. The presence of haze complicates visual perception and poses challenges for computer vision algorithms tasked with analyzing such scenes. Understanding and mitigating the effects of haze is crucial for effective monitoring, detection, and response in fire emergency scenarios.

Fig 7.1 Hazed Images of Fire Detected Areas

Visualizing areas detected with fire using haze imaging technology provides critical insights for early detection and effective firefighting strategies, helping mitigate risks and safeguard communities against wildfire hazards. Even though they're not clear, they help us see where the fires are spreading.

Revealing the De-Hazed Images of fire detected areas unveils a clearer perspective, free from the obscuring effects of smoke and haze. This transformation enhances the visibility of critical details, empowering responders with sharper insights into the extent and intensity of the fire. By removing the visual barriers caused by atmospheric interference, these images become invaluable tools in the arsenal against wildfires, enabling swift and precise decision-making to protect lives, property, and natural resources. The below Fig 7.2 shows us the de-hazed images of fire affected areas making the images more visible to see.

Fig 7.2 De-Hazed Images of Fire detected Areas

ACKNOWLEDGEMENT

The authors deeply acknowledge the support from the department of CSE, City Engineering College. We also want to thank all our team members who gave their tremendous support and successfully completed the project all possible features and ideas.

REFERENCES

[1] KyeongDeokMoon ,on “An end-to-end deep learning approach for real-time image dehazing”(2023), Springer
[2] Sergio Saponara on “Real-time fire/smoke detection based on CNN”(2020), Springer
[3] Chengtuo Jin on “Fire Detection Methods Based on Deep Learning: Datasets, Methods, and future directions”(2023) MDPI
[4] Ju Mingye on “Single image haze removal based on the improved atmospheric scattering model”(2020) ELSEVIER
[5] Shahid Mumtaz on “Energy-Efficient Deep CNN for Smoke Detection in Foggy IoT Environment”(2019) IEEE
[6] Chuansheng Wang , on “Color-Dense Illumination Adjustment Network for Removing Haze And Smoke from Fire Scenario Images” (2022), MDPI
[7] Shuping Li “Image Dehazing Algorithm Based on Deep Learning Coupled Local and Global Features”(2022), MDPI
[8] Dengyin Zhang on “Real-time Image Haze Removal Method for Fire Scene Images”(2018), ICMMCT
[9] Zul Imran Azhari, Samsul Setumin on “Digital image enhancement by brightness and contrast manipulation using verilog hardware description language”(2023), ISSN
[10] Shuying Huang, Donglei Wu1 and Yong Yang on “Image Dehazing Based on Robust Sparse Representation”(2018) , IEEE
[11] P. Tamil Mathi on “A SURVEY ON FOREST FIRE DETECTION”(2015), ISSN
[12] Jun Xu, Zi-Xuan Chen, Zi-Xuan Chen and Zhe-Ming Lu on “An Efficient Dehazing Algorithm Based on the Fusion of Transformer and Convolutional Neural Network” (2022), MDPI
[13] Donghui Zhao , Bo Mo , Xiang Zhu , Jie Zhao on “Dynamic Multi-Attention Dehazing Network with Adaptive Feature Fusion”(2022), MDPI
[14] Xin he , wanfeng ji, and jinpeng xie on “Unsupervised Haze Removal for Aerial Imagery Based on Asymmetric contrastive Cycle GAN”(2022), IEEE
[15] G. Harish Babu on “ABF de-hazing algorithm based on deep learning CNN for single I-Haze Detection” (2023), ELSEVIER