PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

BANGALORE

A Project Report

On

"DETECTING HUMAN LIFE DURING FIRE"

Batch Details

Sl. No.	Roll Number	Student Name
1	20211CAI0159	Safia Rafi
2	20211CAI0152	Fardeen Shariff
3	20211CAI0093	Nikith Murali
4	20211CAI0161	Shilpa Nagraj
5	20211CAI0177	Sohan S

School of Computer Science,

Presidency University, Bengaluru.

Under the guidance of,

Mr. Santhosh Kumar K L
Assistant Professor
School of Computer Science and Engineering
Presidency University

CONTENTS

INTRODUCTION

Problem Statement Number: PSCS220

Organization: Adani Group

Category (Hardware / Software / Both): Software

Problem Description:

The main objective of the idea is to implement human detection in extreme cases using deep learning algorithms via a camera on a drone dispatched during a fire emergency. During fire accidents it's difficult for firemen to know the number of people trapped in and their location. Searching in a smoky environment is hard. Most causalities could be avoided if location of trapped individuals could be known. Our approach to solving this problem is to use drone with camera feed which can detect humans in such situations. We use deep learning algorithms to process the video feed to detect features like human faces and poses. Faces help us recognize the individuals trapped and poses can help us know their conditions, if they need immediate medical attention. This also helps prioritize based on their position and condition. Hence this technology can be used to avoid numerous deaths and help the society

Difficulty Level: Medium

LITERATURE REVIEW

1] Real-time Human Detection in Fire Scenarios using Infrared and Thermal Imaging Fusion

This paper proposes a system that combines infrared (IR) and thermal cameras for human detection in fire scenarios. The system uses a lightweight deep neural network based on YOLOv4-Tiny, integrated with a feature fusion technique. The method processes images from both cameras, merges the data, and achieves high accuracy (mAP@0.5 of 95%) on NVIDIA Jetson Nano hardware. The calibration of cameras ensures better detection, especially in low visibility caused by smoke.

2] A Human Detection Approach for Burning Building Sites Using Deep Learning Techniques

This research focuses on detecting humans trapped in burning sites using thermal cameras and deep learning models. The study proposes using CNNs on thermal images to detect victims in low-visibility fire environments. An Autonomous Embedded System Vehicle (AESV) equipped with thermal cameras collects real-time data from burning sites, which is then analyzed to detect humans and help rescue operations.

3] Fire Detection System in Python Using OpenCV

This work presents a fire detection system using image processing techniques. It employs the Haar-Cascade classifier and Raspberry Pi as the central processing unit to detect fire in real-time. The system is designed to capture images, process them using OpenCV libraries, and trigger fire alerts when flames are detected. It uses a camera to capture video feeds and identifies fire using machine learning algorithms.

EXISTING METHOD DRAWBACKS

1] A Human Detection Approach for Burning Building Sites Using Deep Learning Techniques (Jaradat & Valles)

- **Heavy reliance on thermal imaging:** While thermal cameras can detect heat and are effective in smoky environments, they may not always provide enough contrast to differentiate between humans and other heat sources, especially in complex environments with multiple heat-emitting objects.

- **CNN model limitations:** The approach relies on a CNN trained on thermal images, which can lead to false positives or missed detections if the training dataset does not cover enough variations (e.g., body poses, environmental conditions).

- **Dataset limitations:** There are challenges in acquiring realistic datasets for thermal imaging, especially those that match the angles and resolutions required for training. Modifying RGB datasets to mimic thermal images may not capture the full complexity of real-world conditions.

- **Limited smoke assessment:** While thermal cameras can work in smoke-filled environments, the methodology lacks advanced smoke classification or visibility predictions that could further enhance detection accuracy in extremely dense smoke conditions.

2] Fire Detection System Using OpenCV (Ismail et al.)

- **False alarms:** The use of traditional image-processing techniques such as Haar cascades for fire detection can lead to false alarms, especially in cases where light sources mimic the color or intensity of fire. This is a significant drawback in environments with fluctuating lighting conditions.

- **Limited to visible flames:** The system primarily detects fire based on visible flames, which means it cannot detect smoldering fires or fire hazards that do not yet produce visible flames. This limits its applicability in early-stage fire detection.

- **Environmental limitations:** The system struggles in outdoor or open environments where the spread of fire might not follow predictable patterns due to wind or other factors. Its effectiveness is also reduced in very large spaces like stadiums and aircraft hangars.

- **Limited by hardware:** The reliance on a low-cost Raspberry Pi and basic webcam limits the system's overall performance in terms of processing power, resolution, and sensitivity, which affects real-time detection accuracy and speed.

3] Real-time Human Detection in Fire Scenarios using Infrared and Thermal Imaging Fusion (Truong et al.)

- **Infrared camera limitations in heavy smoke*:** Infrared cameras can become less effective in environments with dense smoke as they lose clarity and become unreliable. This can lead to missed detections or inaccurate localization of people in need of rescue.

- **Thermal camera confusion:** Thermal cameras, while effective in smoke, can be confused by heat sources other than humans (e.g., fires, heated objects). This can result in false positives, where non-human heat sources are mistakenly detected as people.

- **Fusion complexity:** The fusion of thermal and infrared images adds complexity to the system. Proper calibration and alignment are crucial for accuracy, and any misalignment between the cameras can lead to detection errors. Additionally, the fusion process requires high computational resources, which could limit its applicability in real-time scenarios.

- **Limited to specific hardware:** The proposed solution relies on NVIDIA Jetson Nano and specific camera models, making it less flexible for implementation in systems with different hardware configurations.

PROPOSED METHODS

a) Video Upscaling Algorithm for Reducing Smoke and Improving Visibility.

- **Algorithm:** Smoke Reduction and Super-Resolution Video Upscaling.

1] Input: Low-resolution, smoke-obscured video frames captured during fire scenarios.

2]Output: Clearer, upscaled video frames with reduced smoke and improved visibility.

Description: This algorithm aims to enhance visibility by detecting smoke and applying dehazing techniques before upscaling video frames to a higher resolution. The video first undergoes a segmentation phase to isolate smoke, followed by smoke reduction using deep learning-based dehazing. The dehazed frames are then upscaled using GAN-based super-resolution to improve the visual quality, providing clearer, sharper frames to help rescue efforts during fires.

b) Multiple Posture Detection Algorithm for Body Posture Recognition.

- **Algorithm:** CNN-Based Multiple Posture Detection

1] Input: Video frames containing people in various postures (standing, lying, crouching)

2] Output:

  - Posture labels for each frame (e.g., lying down, standing) with confidence scores.
  - Optional: Highlight bounding boxes around detected individuals and their posture.

3] Description: This algorithm focuses on recognizing different human postures during fire situations, which is critical for assessing an individual's condition. It uses deep learning-based CNNs for feature extraction and posture classification. In cases requiring further refinement, a pose estimation model can be employed to detect human body key points and classify postures based on joint positions. By recognizing various postures like lying down (indicating unconsciousness) or crouching (suggesting escape attempts), the algorithm provides vital information for first responders to prioritize rescue efforts.

OBJECTIVES

The main objectives of the project, focusing purely on the software aspect, are as follows**:**

Develop a Human Detection Model Using Deep Learning:

- Build an accurate and robust deep learning model using Convolutional Neural Networks (CNNs) to detect human presence in images or video streams captured during fire emergencies.

Implement Real-time Face Detection:

- Utilize Single Shot Detection (SSD) techniques to perform efficient and real-time face detection in challenging conditions such as smoke, low visibility, and fluctuating lighting.

Detect Multiple Body Postures:

- Train a model that can recognize various body postures (e.g., lying down, standing, crouching) to help locate and identify individuals in different scenarios and analyze their conditions during a fire.

Optimize for Low-latency Inference:

- Ensure the system performs real-time detection with minimal delay to allow rapid response in emergency situations.

Improve Detection Accuracy in Challenging Environments:

- Enhance the robustness of the model to handle real-world challenges like smoke, fire, debris, and low visibility, using techniques such as data augmentation, transfer learning, and specialized preprocessing.

Evaluate Model Performance:

- Conduct thorough performance evaluation using metrics like accuracy, precision, recall, and F1-score to ensure the system reliably detects humans in critical situations.

Ensure Scalability and Flexibility:

- Develop the software to be flexible and scalable, allowing integration with different input sources (e.g., video feeds from surveillance cameras) and easy deployment on various platforms, including edge devices.

EXPERIMENTAL DETAILS/METHDOLOGY

Software and Hardware Requirements:

Hardware Requirements:

Processing Unit:

- GPU (Graphics Processing Unit): For training deep learning models and running real-time inference.
- CPU (Central Processing Unit): High-performance processor for general computations.

Memory:

- **RAM**: Minimum 16 GB for training models, recommended 32 GB or more for handling large datasets and training efficiently.
- **Storage:** At least 500 GB SSD for faster read/write operations, with optional external storage for large video datasets.

Software Requirements:

Operating System:

- **Windows 10/11**: For compatibility with most deep learning frameworks and GPU support.

Programming Language:

- **Python 3.x**: Primary language for writing the code and implementing deep learning algorithms.

Deep Learning Frameworks:

- **TensorFlow 2.x / Keras** or **PyTorch**: For building and training CNNs, SSD, and body posture detection models.

- **Libraries/Dependencies:**

  OpenCV: For image processing and handling video feeds from the drone camera.
  NumPy and **Pandas**: For numerical computations and data manipulation.
  scikit-learn: For additional machine learning tasks and model evaluation.
  Matplotlib/Seaborn: For visualizing results and performance metrics during training

  Development Tools:

- **Jupyter Notebook** or **PyCharm**: For writing and testing Python code.

- **Anaconda**: For managing Python packages and virtual environments.
- **Git**: For version control, with **GitHub** or **GitLab** for collaboration and code storage.

Video/Annotation Tools:

- **LabelImg** or **YOLO (You Only Look Once)**: For labeling human faces and body postures in the dataset

METHODOLOGY

1.Libraries Used:

OpenCV (cv2): OpenCV is used for video capturing, image processing, drawing bounding boxes, and saving the processed video. It's a versatile library for real-time computer vision tasks.

NumPy (np): Used for numerical operations, though not utilized explicitly in this code.

Ultralytics YOLO: A Python API for using YOLOv8 models, a state-of-the-art object detection algorithm. YOLOv8 (YOLO class) is used to perform object detection in each frame of the video.

2.Modules and Methodology:

Video Capture and Initialization:

cv2.VideoCapture(video path): Opens the video file for frame-by-frame processing.
- The code checks if the video has been opened successfully (cap.isOpened()).
- **cv2.VideoWriter**(): Initializes the output video writer with codec (fourcc), frame rate (20 FPS), and resolution (from the input video) to save the processed video.

YOLOv8 Model Loading:

model = YOLO('yolov8n.pt'): Loads the YOLOv8 "nano" model for object detection, which is a lightweight version optimized for performance. This model is trained to detect a variety of objects including humans and possibly fire (if the dataset includes it).

Bounding Box and Color Definitions:

The code defines colors for the bounding boxes:
- Green for humans.
- Red for fire.
- Yellow for smoke.

OUTCOMES

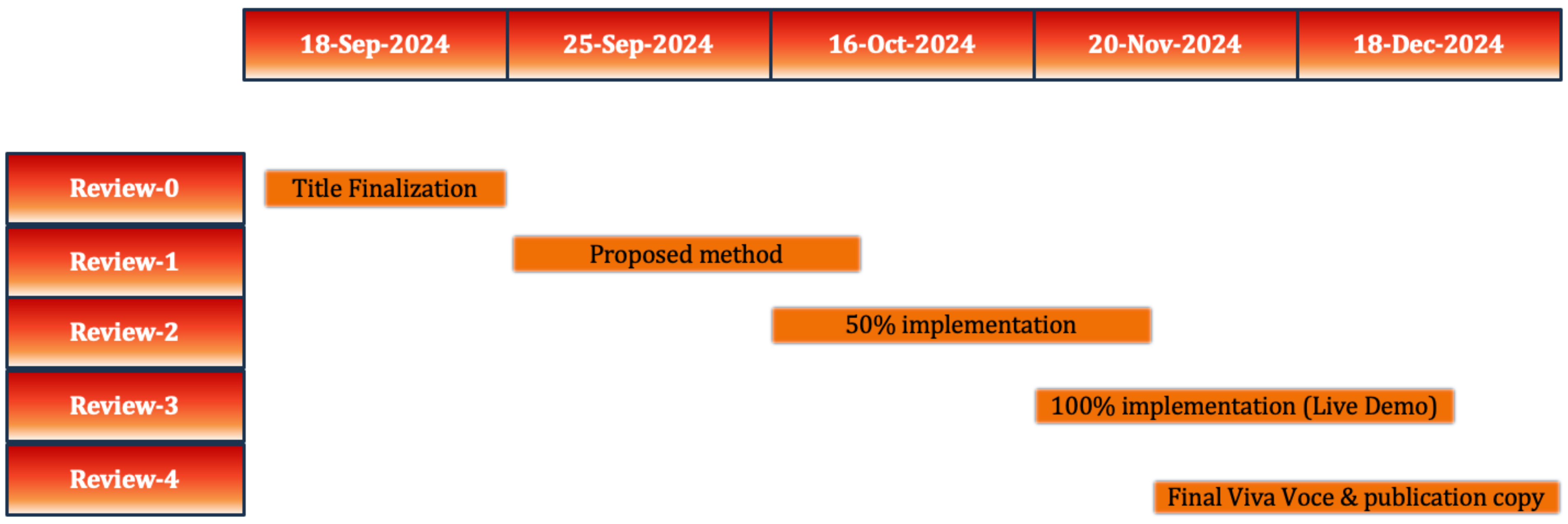
- A human detection system that can efficiently filter smoke and enhance video quality to enable accurate detection of humans.

- Multiple posture recognition system to accurately detect humans in different postures and improve accuracy. This also helps in identifying the condition of the human and helps in determining who needs immediate medical attention.

- Fast response times with real-time processing, ensuring the system is practical in emergency situations.

- The system provides timely alerts about human presence and their conditions, helping rescue teams prioritize and act faster.

TIMELINE OF THE PROJECT/ PROJECT EXECUTION PLAN



CONCLUSION

- The project successfully employs video upscaling and smoke reduction techniques, significantly improving visibility in smoke-filled environments. This enhances the ability to detect individuals during fire emergencies.

- Using deep learning algorithms, the system accurately detects human presence and recognizes various postures (e.g., standing, crouching, lying down), which is crucial for assessing individuals' conditions during a fire.

- The system performs effectively under challenging conditions such as heavy smoke, low light, and debris, ensuring reliable detection of human life in various fire environments.

- The project demonstrates great potential for improving fire rescue efforts by helping firefighters identify and prioritize rescue operations, thereby saving more lives in fire emergencies.

REFERENCES

1] Do, Truong-Dong & Truong Nghe, Nhan & Le, My-Ha. (2023). Real-time Human Detection in Fire Scenarios using Infrared and Thermal Imaging Fusion. 10.48550/arXiv.2307.04223.

2] Jaradat, Farah & Valles, Damian. (2019). A Human Detection Approach for Burning Building Sites Using Deep Learning Techniques. 10.1109/CSCI.2018.00276.

3] B. E. Marty Ahrens, "National Fire Protection Association. (2020). Fire Loss in the United States During 2019.," Sep. 01, 2020. Accessed: May 09, 2023. [Online]. Available: https://www.nfpa.org/News-and Research/Publications-and-media/NFPA-Journal/2020/September October-2020/Features/Fire-Loss

4] T. D. Do, K. Kim, H. Park, and H. J. Yang, "Image and encoded text fusion for deep multi-modal clustering," ACM International Conference Proceeding Series, pp. 308–312, Sep. 2020, doi: 10.1145/3426020.3426110.