CSE 4128 : Image Processing and Computer Vision Laboratory

Fire Detection Using Image Processing

Presented To:

Dr. SK. Md. Masudul Ahsan

Professor
Department of Computer Science & Engineering
Khulna University of Engineering & Technology

Dipannita Biswas

Lecturer
Department of Computer Science & Engineering
Khulna University of Engineering & Technology

Presented By:

Md. Masudur Rahman Rabby

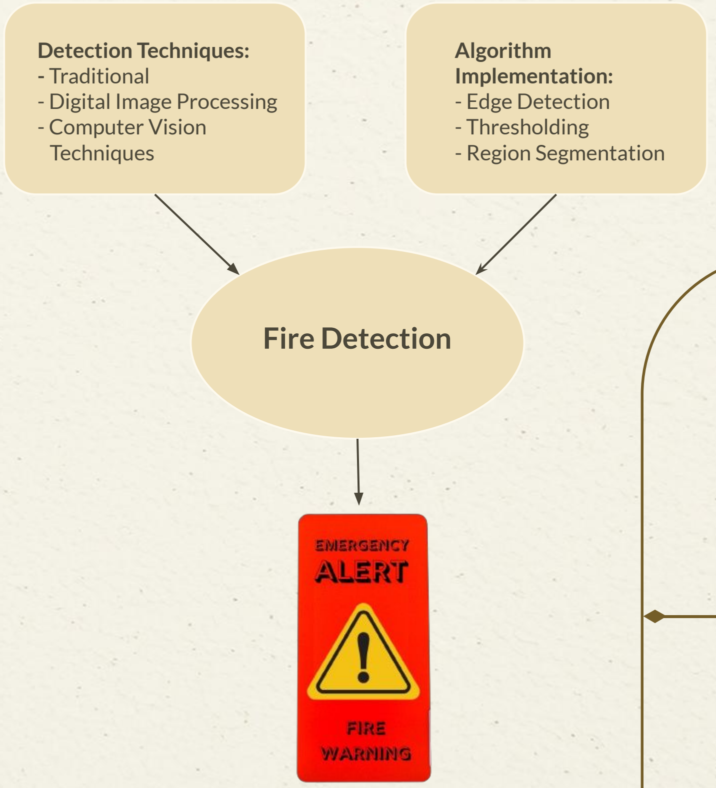
Roll: 1907113
Department of Computer Science & Engineering
Khulna University of Engineering & Technology

Table of contents

01
Introduction

Introduction

- **Fire detection:** The process of identifying the presence of fire through sensors or computer vision techniques.
- **Methods:** Using image processing, computer vision techniques or combined technologies.
- **Detection Techniques:** Includes infrared sensors, thermal imaging cameras, and computer vision algorithms that analyze patterns and changes indicative of fire.
- **Algorithm Implementation**: Implement algorithm like edge detection, thresholding, and region segmentation to isolate and identify potential fire regions in digital images or video frames.



02
Project Focus

Project Focus

- Develop a fire detection algorithm using basic image processing techniques.
- Developing a GUI for user comfort of the application.
- Avoid reliance on machine learning models which require extensive training data and computational resources.

03
Application Features

Application Features

- User-friendly GUI built with Tkinter.
- Allows easy image loading and processing.
- Sequential display of image processing steps for transparency and educational purposes.

04
Key Steps

Key steps

- **Load Image**: Allows selection and loading of an image.
- **Clear Panels**: Resets GUI for new image results.
- **Display Functions**: Shows images and results on GUI frames.
- **Detect Fire**: Key steps include:
    - Convert to HSV for color-based analysis.
    - Apply threshold to identify fire regions.
    - Use morphological operations to refine results.
    - Find and draw bounding boxes around detected fires.

05
Methodology

Methodology

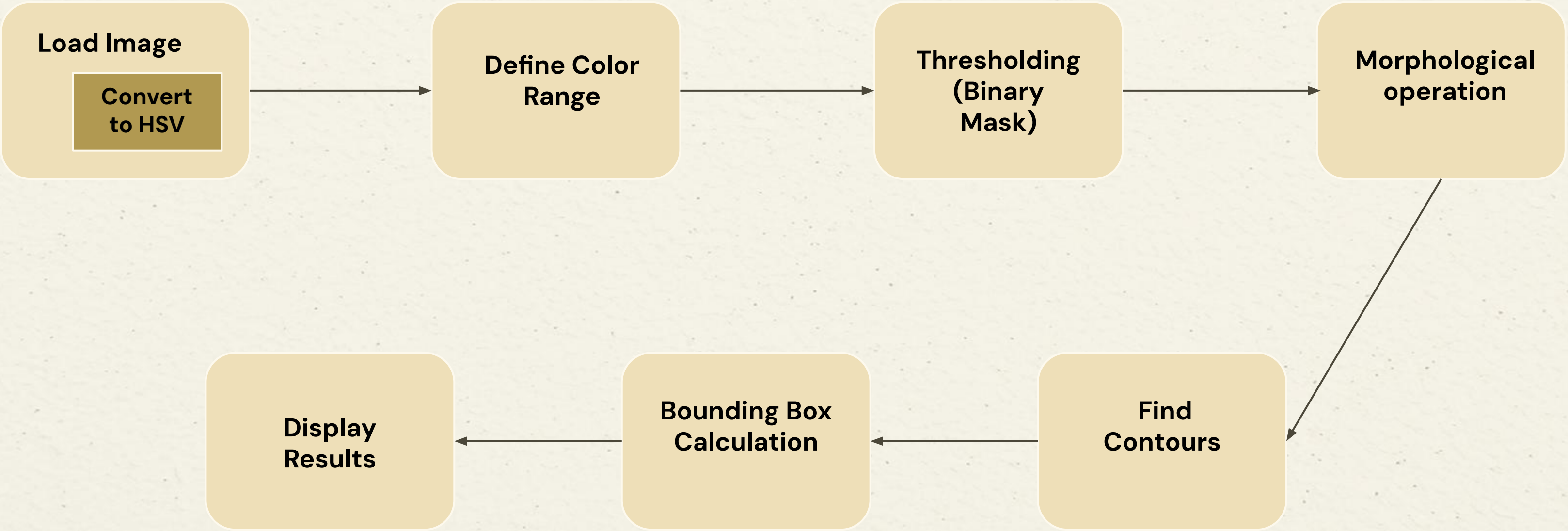


Fig 1: Block diagram of Fire Detection Algorithm

Methodology (cont.)

- **Convert to HSV**:
  - The algorithm first convert the loaded images from RGB (Red-Green-Blue) to HSV (Hue-Saturation-Value) color space.
  - Then HSV separates color information from intensity, making it suitable for color-based segmentation.

Fig 2: Original input image(RGB)

- **Color Range Definition**:
  - It define the HSV color range that corresponds to fire colors.
  - The lower and upper bounds of HSV values (e.g., [18, 50, 50] to [35, 255, 255]) that typically represent the colors of fire.

Fig 3: HSV converted image

Methodology (cont.)

- **Thresholding and Binary Mask Creation**:
  - After that a threshold is applied to the HSV image based on the defined color range.
  - Next creating a binary mask where pixels within the thresholded range are set to white (255), indicating potential fire regions, and others are set to black (0).

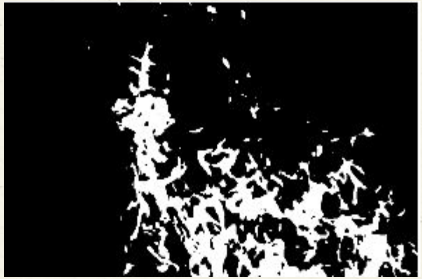


Fig 4: Binary fire mask

Methodology (cont.)

- **Morphological Operations for Noise Reduction**:
  - Then used morphological operations such as dilation and erosion to refine the binary mask.
  - Dilation expands the white regions, helping to connect nearby pixels that may belong to the fire region.
  - Erosion then shrinks the regions, smoothing out irregularities and removing small noise components.

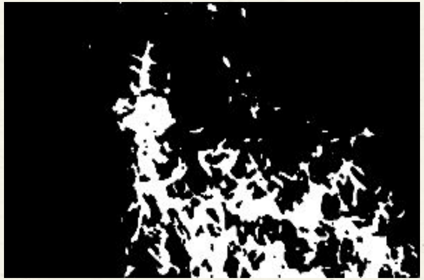


Fig 5: Fire mask after morphological closing

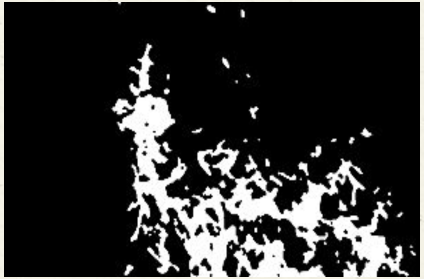


Fig 6: Fire mask after morphological opening

Methodology (cont.)

- **Contour Detection**:
  - Now from the mask identifying contours within the processed binary mask.
  - Contours are curves joining continuous points along the boundary of white regions, representing potential fire areas.

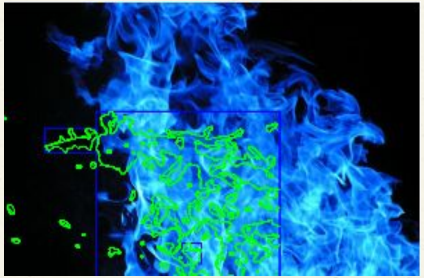


Fig 7: Contour detected image

- **Bounding Box Calculation**:
  - Finally, computing bounding rectangles around the detected contours.
  - Bounding rectangles provide a visual representation of the spatial extent of identified fire regions in the original image.

Fig 8: Fire area detected boxed image

06
Result and Output

Result and output

- **Input and output image**:
  - Here after finding the contours, on the basis of contour detection the algorithm decides whether the fire is present or not.

Fig 9: Input image

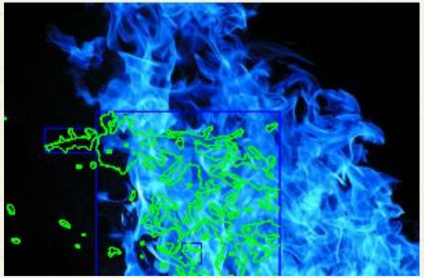


Fig 10: Contour detected image

Fig 11: Output image(fire detected)

Result and output (cont.)

- **Input and output image**:
  - Here after finding the contours, on the basis of contour detection the algorithm decides whether the fire is present or not.

Fig 12: Input image

Fig 13: Contour detected image (not detected)

Fig 14: Output image(no fire detected)

07
Limitations

Limitations

- ❑ Sensitivity to lighting conditions.
- ❑ Occlusions and obstructions affecting detection.
- ❑ Limited robustness to image noise.
- ❑ Dependence on camera quality and resolution.
- ❑ Designed for static images, not real-time video.
- ❑ Environmental variability affecting accuracy.
- ❑ Non-generalized HSV threshold values.
- ❑ Limited detection range for unusual fires.

08 Conclusion

Conclusion

The project demonstrates effective fire detection using image processing techniques.

The HSV thresholds and algorithm parameters may need adjustments for different environments and fire types.

The project is a little sensitive to lighting, environmental variables, and computational efficiency are notable limitations.

Integrating the real-time processing and improving robustness to environmental variations can enhance the system.

Thanks!

Do you have any questions?

masudurrabby8@gmail.com
+8801840794147