CECS 553 – MACHINE VISION

Project Name : **FIRE DETECTION USING DEEP LEARNING**

Team Members : **Aishwarya Bhavsar & Andrew Ramirez**

Date : 29th Nov 2022

Outline:

1. Introduction
2. Problem statement
3. Proposed solution(s)
4. Solution(s) details
5. Deliverable materials (Demo, …)

1. INTRODUCTION

- Having a fire detection system **can significantly reduce damages and maximize fire control efforts**.
- It is also one of the most fundamental steps you can take for **fire safety measures.**
- Even if you are sleeping or busy working, early fire detection **will warn you and help you respond** quickly so you'll be out of danger.
- **Saves life.**
- **Reduces loss of property.**
- **Shorten recovery time.**
- **For insurance discounts.**
- **Keep you code-compliant.**

2. PROBLEM STATEMENT

- Fires can start randomly, and they can spread quickly, causing major damage before they are noticed.
- Currently, we have fire alarm systems that detect the fire using **smoke sensors.** But they are inefficient, in terms of time and are sometimes are not properly installed. If the owner is away, they may not be around to hear the alarm.
- We need a detection system that can **identify the fire with accuracy** within seconds and notify the user to take further action.
- It can take **just 30 seconds for a small flame** to turn into a major blaze (Department of Homeland Security).
- The top three **causes of fires** in homes are
  - cooking,
  - heating equipment,
  - electrical malfunction.
- An average of **358,500** homes experience a structural fire each year (NFPA).
- More than **3,000 Americans** die in fires each year (FEMA).
- Every day, at least **one child dies** from a fire inside the home (Stanford Children's Hospital).
- Reference : https://www.thezebra.com/resources/research/house-fire-statistics/

3. PROPOSED SOLUTION

- Creating a customized CNN Architecture: TensorFlow API Keras
- Use data augmentation techniques.
- Create CNN Model.
- Use activation functions for improving the accuracy.

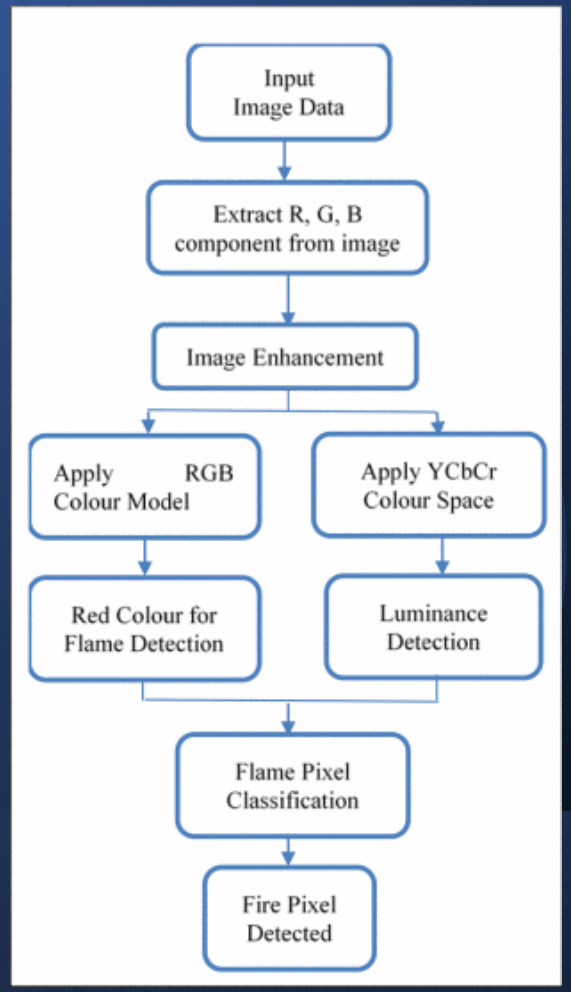
- **REAL TIME TESTING:** [Using OpenCV]
- 1. Take a real time video,
- 2. Cut it into frames.
- 3. Preprocess frames
- 4. Input the frames into the CNN Model.
- 5. Detect whether "Fire" or "No Fire"

4. SOLUTION IN DETAILS

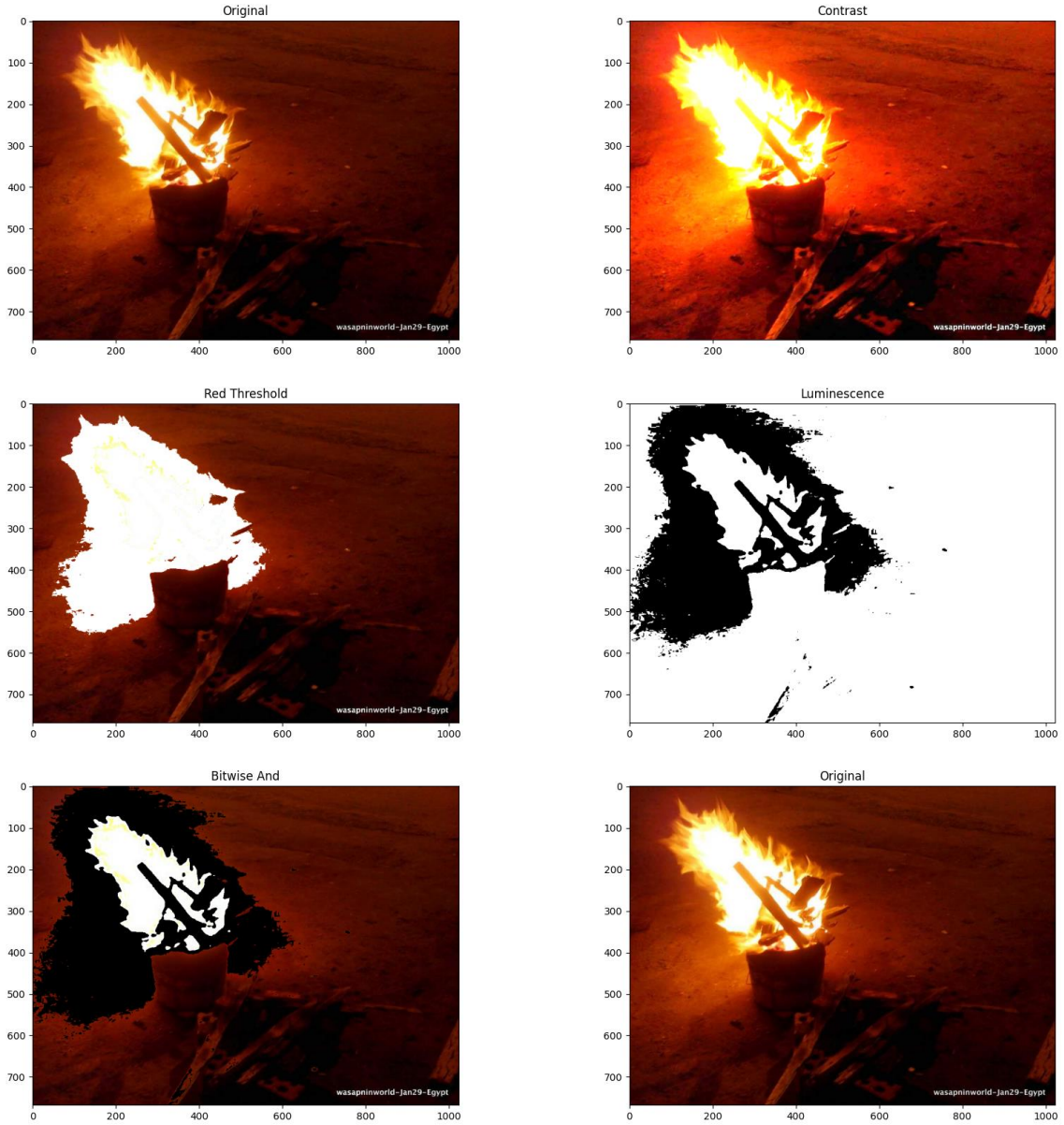
- **Step 1 :** Obtain Dataset
- **Step 2 :** Import Libraries (Data Science, TensorFlow, System Libraries)
- **Step 3 :** Create Helper Functions
- **Step 4 :** Load & Transform The Data
- **Step 5 :** Placing Data into Data Frames
- **Step 6 :** Visualizing Images From Dataset
- **Step 7 :** Data Pre-processing
- **Step 8 :** Training The Model
- **Step 9 :** Model Evaluation
- **Step 10 :** Visualizing Loss Curves
- **Step 11 :** Make Predictions on the Test Data
- **Step 12 :** Plotting Classification Reports & Confusion Matrix

4. SOLUTION IN DETAILS - PARAMETERS

```
x = Dense(256, activation='relu')(pretrained_model.output)
x = Dropout(0.2)(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.2)(x)


outputs = Dense(2, activation='softmax')(x)

model = Model(inputs=inputs, outputs=outputs)

model.compile(
    optimizer=Adam(0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```



Methodology for dataset transformation

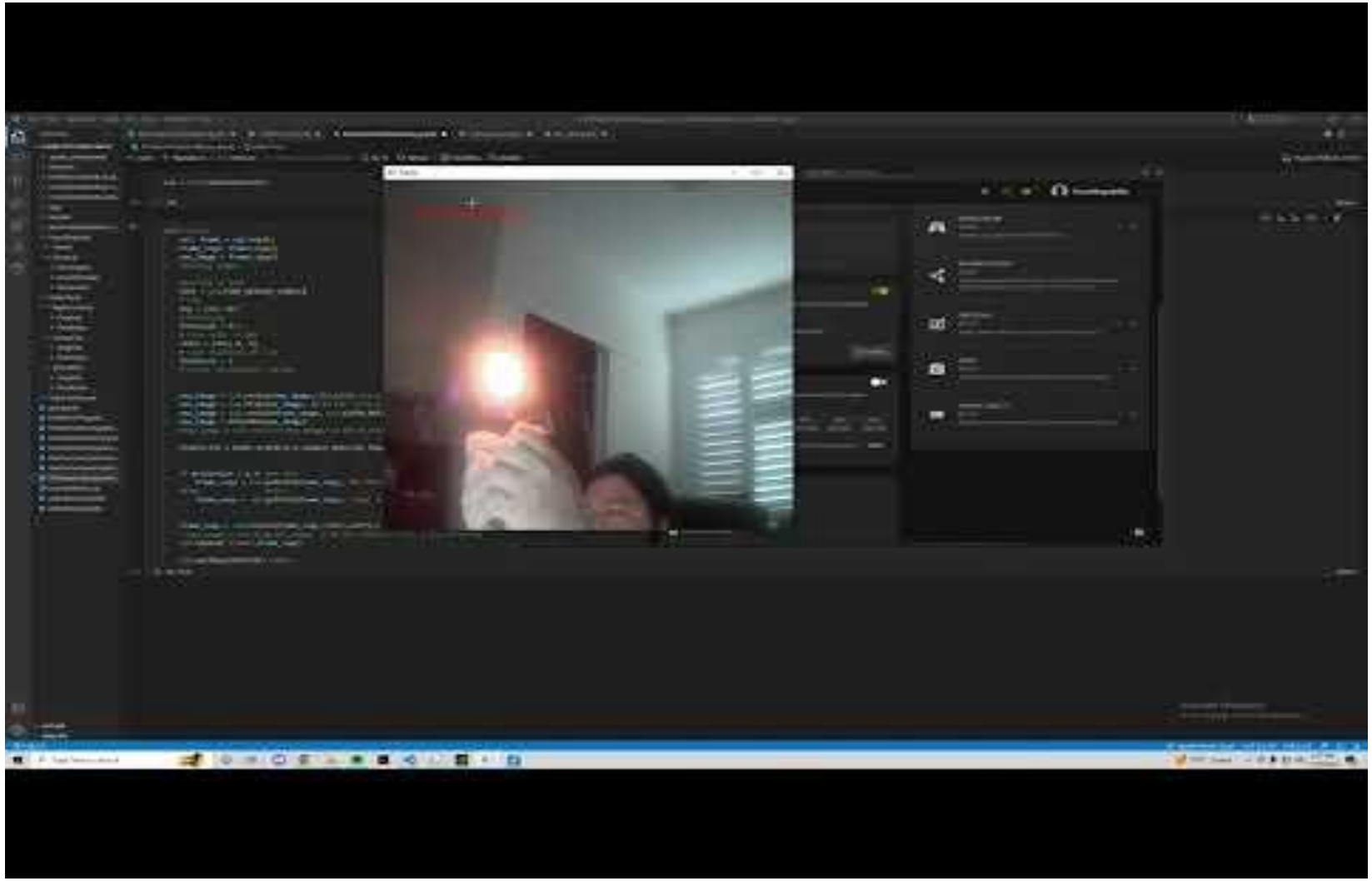


A methodology for fire detection using colour pixel classification by Nurul Shakira Bakri; Ramli Adnan; Abd Manan Samad; Fazlina Ahmat Ruslan

4. SOLUTION IN DETAILS – Real Time Processing

- **Step 1 :** OpenCV to get the image
- **Step 2 :** Create input & target arrays.
- **Step 3 :** Split the real time video in frames.
- **Step 4 :** Feed the frames in our CNN model created.
- **Step 5:** Train the Neural Network.
- **Step 6 :** Predict the confidence.
- **Step 7 :** Conclude whether "**Fire Detected**" or "**Non - Fire**".

5. DEMO



6. FUTURE SCOPE

1. Future studies may focus on deploying the model into **raspberry pi** and using necessary support packages to detect the real time fire by making challenging and **specific scene understanding datasets for fire** detection methods and detailed experiments.

2. We can also expect better deep learning **architectures** to emerge in the future, offering better feature extraction.

3. **CNN can be clubbed with R-CNN LSTM.**

7. REFERENCE

•**DEEP LEARNING ALGORITHM FOR FIRE DETECTION**

• https://ieeexplore.ieee.org/document/9263456

•**Early fire detection using deep learning & OpenCV** https://towardsdatascience.com/early-fire-detection-system-using-deep-learning-and-opencv-6cb60260d54a

•https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8307064

• https://ieeexplore.ieee.org/document/9619342

• https://ieeexplore-ieee-org.csulb.idm.oclc.org/document/8368692/references#references

THANK YOU
Questions?