FIRE DETECTION
RGB VIDEOS FOR FOREST FIRE PREVENTION

Alessio Olivieri
Emilio Soriano Chávez
Robert Li

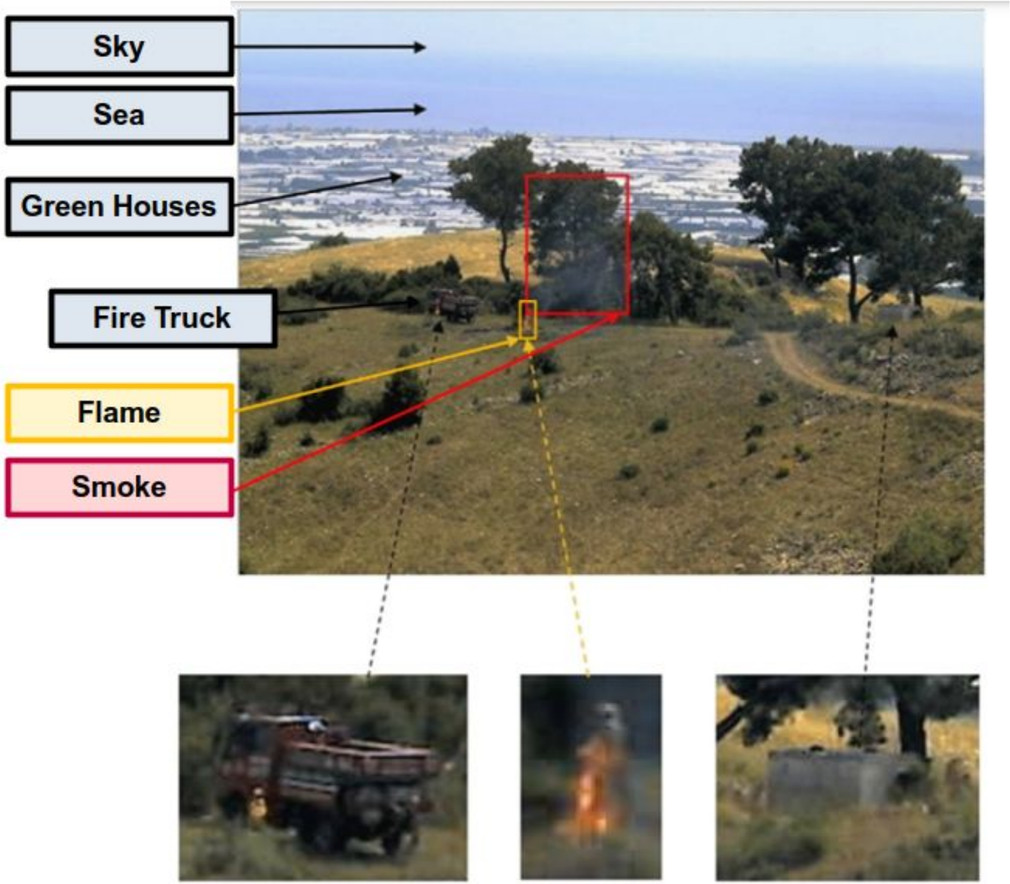
INTRODUCTION

- Forest fires are an **important threat** to natural ecosystems.
- **Early detection** is essential to prevent extensive damage and reduce associated risks.
- *Smoke-based fire detection* is not effective in open environments due to smoke dispersion, wind patterns, and smoke densities.
    - **False alarms** are frequent when smoke from non-fire-related sources trigger the sensors.
- *Temperature-based fire detection* easily identifies regions with high temperatures.
    - But it **cannot differentiate** between regular heat sources and actual fire incidents.

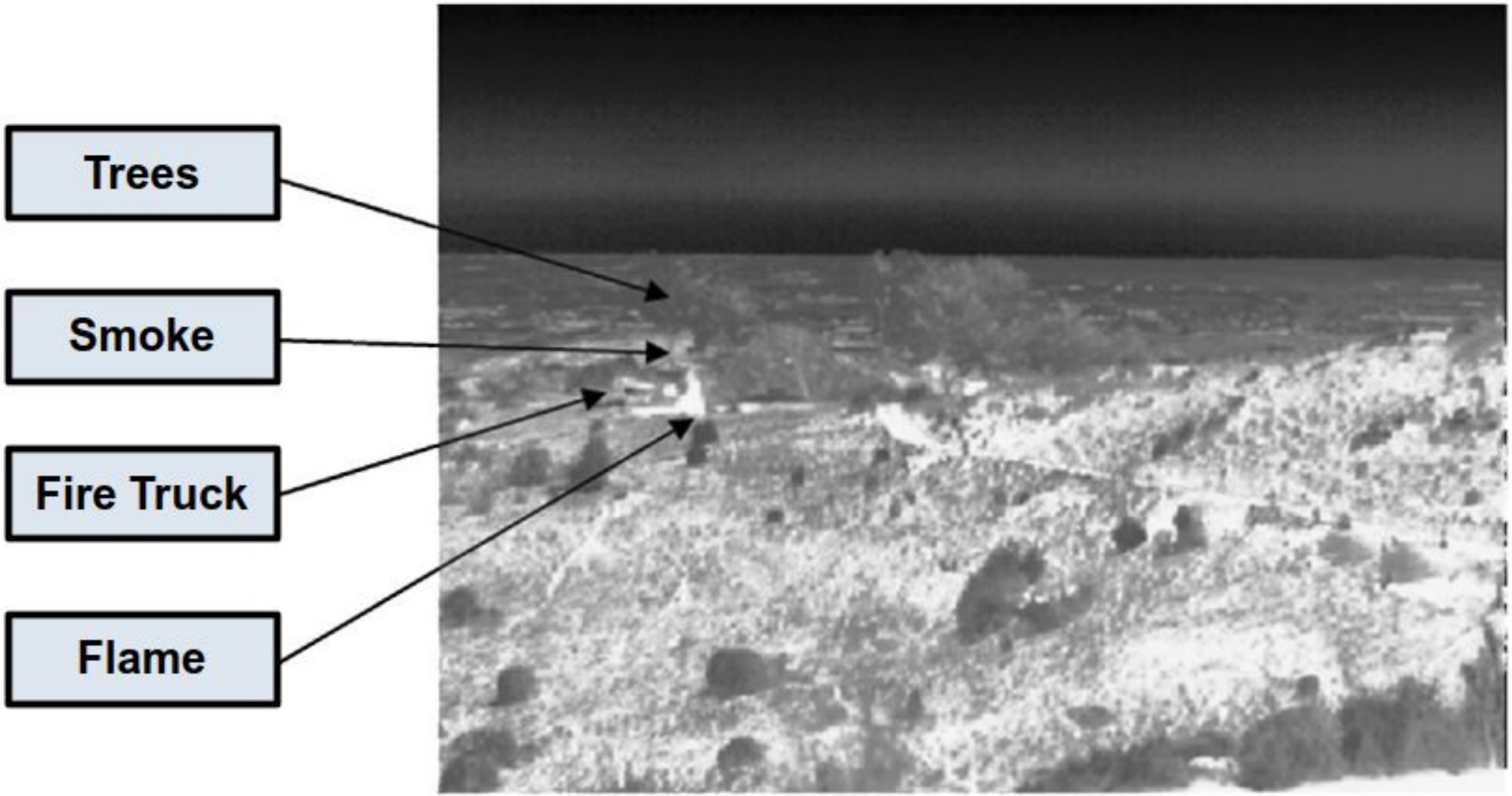
FIRE DETECTION USING CAMERAS

- One approach is to fire detection are **Infrared (IR) Cameras**.
- The **high cost** of infrared cameras compared to conventional digital cameras represents a problem.
- Another issue is the fact that smoke spreads faster than fire.
- It may not be even possible to observe flames for a long time.
- Wildfire smoke detection using an IR camera is very difficult as **smoke is invisible**.
- *There are too many cameras and too few pairs of eyes to keep track of them.*

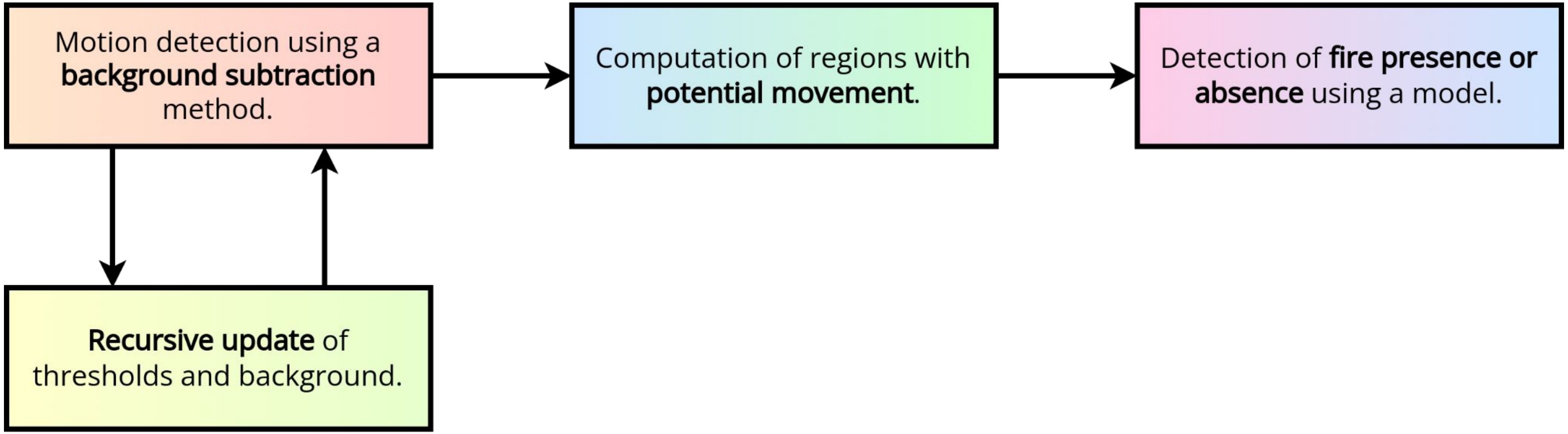
ISSUES WITH IR CAMERAS I



ISSUES WITH IR CAMERAS II



MOTION DETECTION ALGORITHM



BACKGROUND SUBTRACTION

- First, we compare consecutive video frames and **identify pixels or regions** that exhibit **significant changes**.
- We initialize a Threshold Image (T), with the same size as the input frame.
- We initialize a Background Image (B), and assign the first frame of the video.
- A **moving pixel** is determined using the following equation:

$$|I(x,n) - B(x,n)| > T(x,n)$$

BACKGROUND SUBTRACTION

$$|I(x,n) - B(x,n)| > T(x,n)$$

RECURSIVE UPDATE

- The **Background Image (B)** is updated as follows:

$$B(x,n+1) = \begin{cases} a \cdot B(x,n) + (1-a) \cdot I(x,n) & \text{if} |I(x,n) - I(x,n-1)| > T(x,n) \\ B(x,n) & \text{otherwise} \end{cases}$$

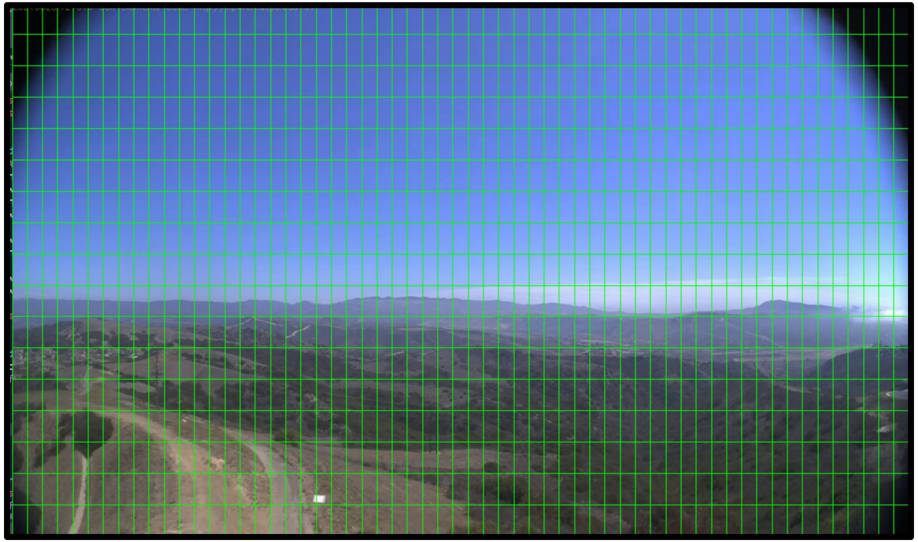
- The **Threshold Image (T)** is updated as follows:

$$T(x,n+1) = \begin{cases} a \cdot T(x,n) + (1-a) \cdot (c \cdot |I(x,n) - B(x,n)|) & \text{if} |I(x,n) - I(x,n-1)| > T(x,n) \\ T(x,n) & \text{otherwise} \end{cases}$$

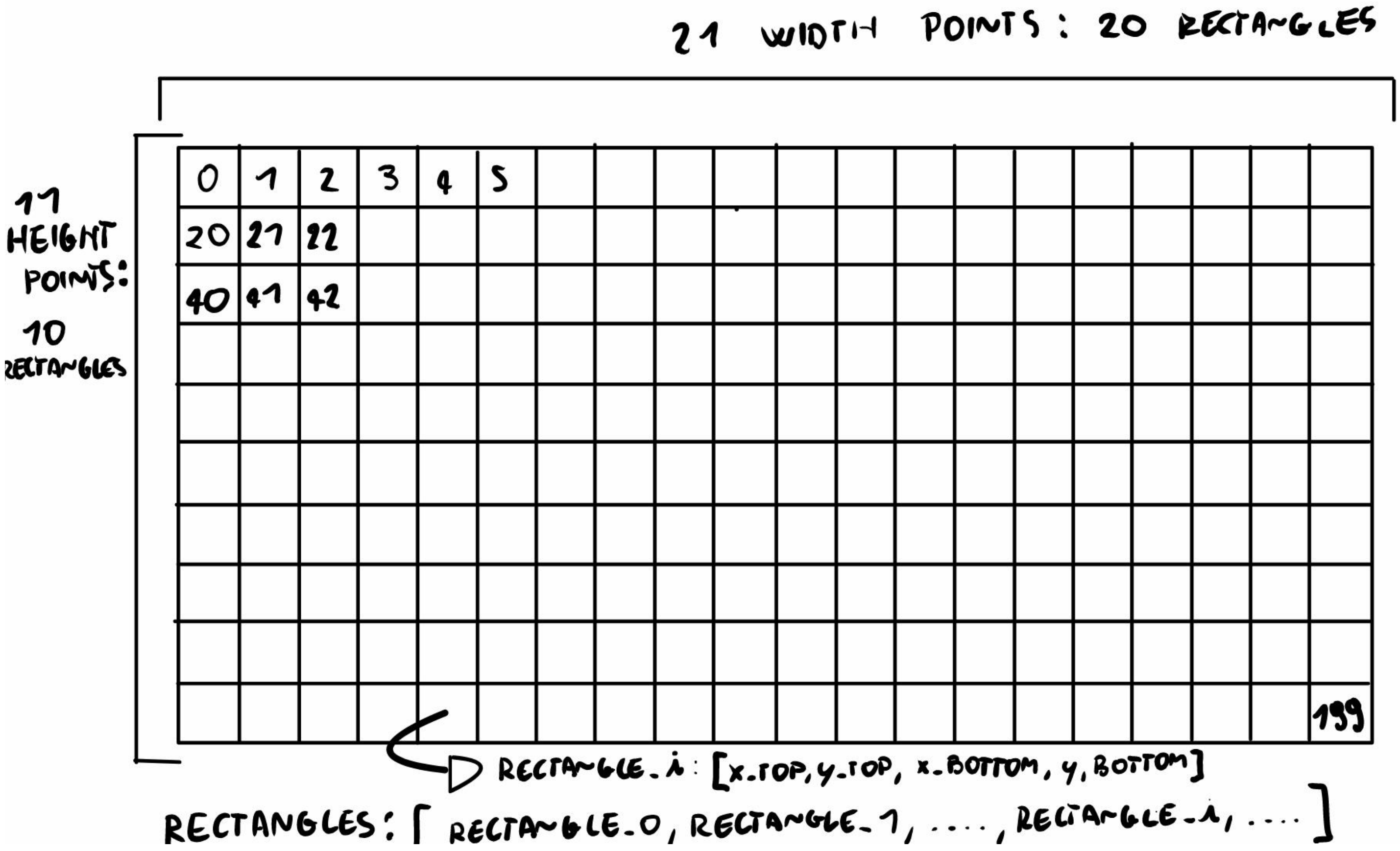
- This is done for every analyzed frame of the video.

COMPUTATION OF REGIONS I

- After determining which pixels are moving, we must determine if the **number of moving pixels is enough** to conclude movement.
- To do so, the video frame is divided into segments (squares) to create a **meshgrid**.

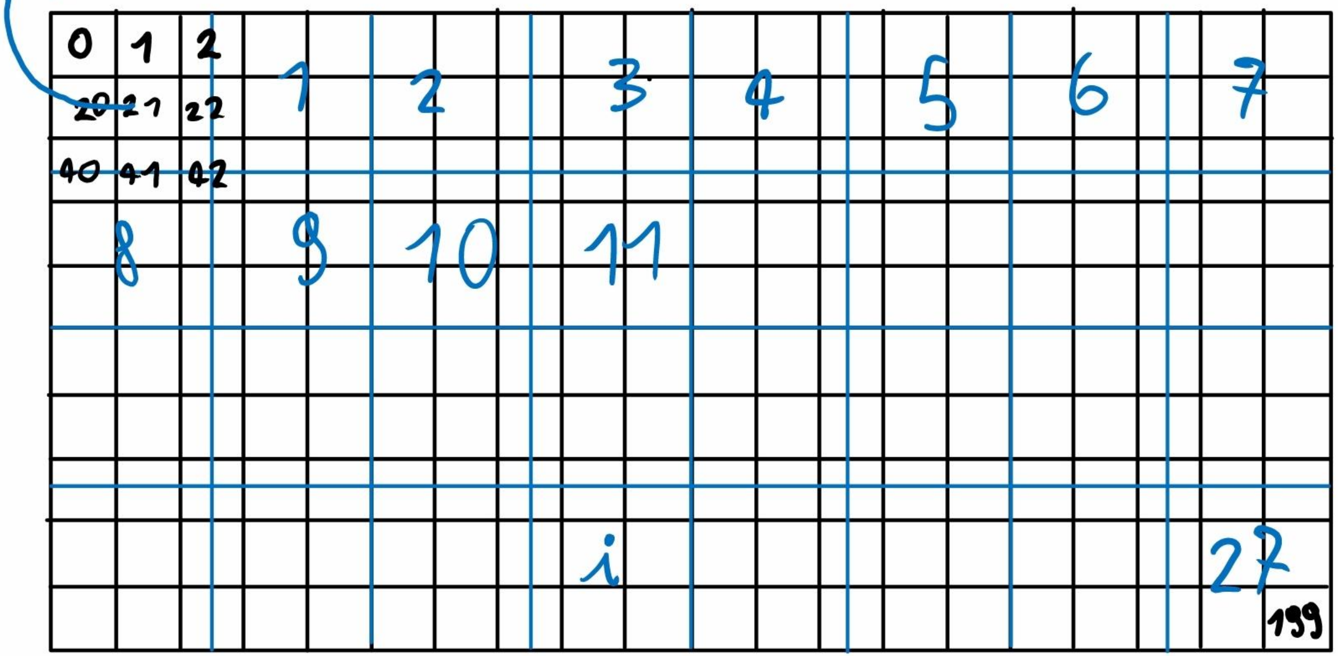


REGIONS DATA STRUCTURE

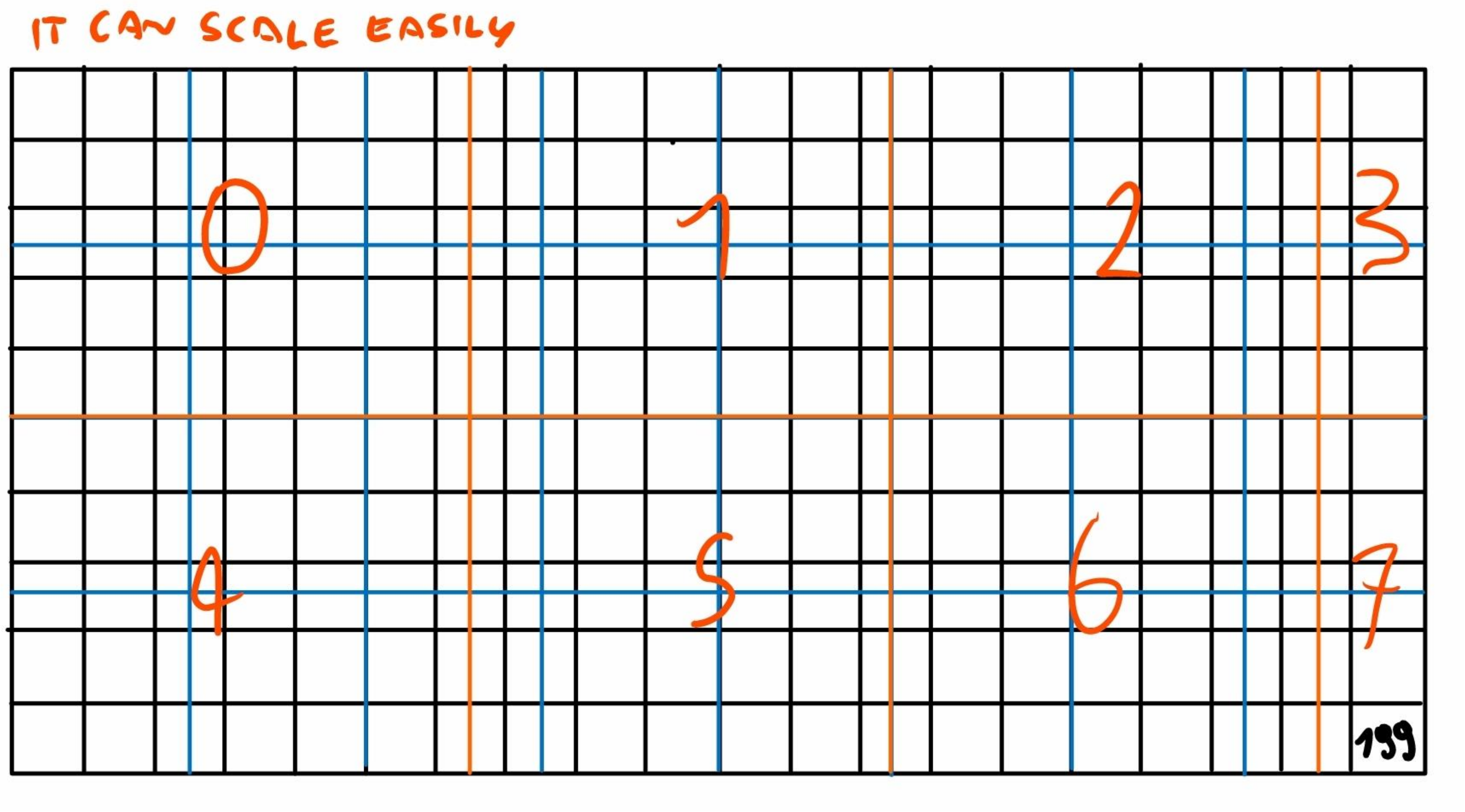


REGIONS DATA STRUCTURE

BIG_RECTANGLE_0_CONTAINS: [0, 1, 2, 20, 21, 22, 40, 41, 42]
CONTAINED: [BIG_RECTANGLE 0_CONTAINS, ..., BIG_RECTANGLE_27_CONTAINS]
BIG_RECTANGLE_0: [x_TOP, y_TOP, x_BOTTOM, y_BOTTOM]
BIG_RECTANGLES: [BIG_RECTANGLE_0, ..., _27]



REGIONS DATA STRUCTURE



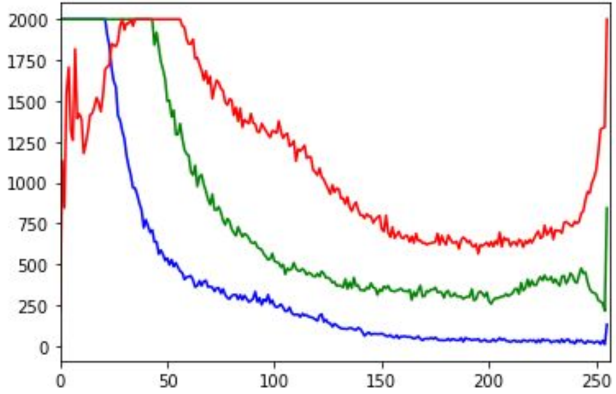
COMPUTATION OF REGIONS II

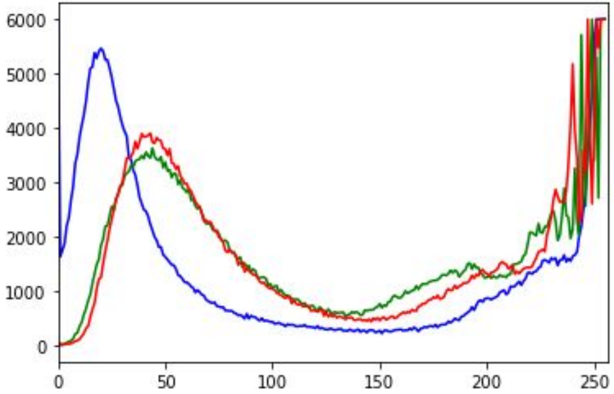
- If the more than 25% of pixels of smaller region are moving, we say that the region is moving
- For each *large* region, we calculate the number of *small* regions in which movement was noticed
- If the area of moving regions exceeds 25% we consider the *large* area instead.
- This is done for each degree of moving regions
- The areas where movement is determined are then sent to the model to determine the probability that the fire is present.

FEATURE EXTRACTION I

- The model was trained using **labeled image datasets** containing *fire* and *non-fire* fores images.
- For each image, a **histogram** was computed for each of the RGB and HSV channels.
- We work under the assumption that the histograms of a *fire* and a *non-fire* image (or frame) **differ significantly** so as to be used as representations of the images.
- This allows the model to differentiate between them, without using the entire image.

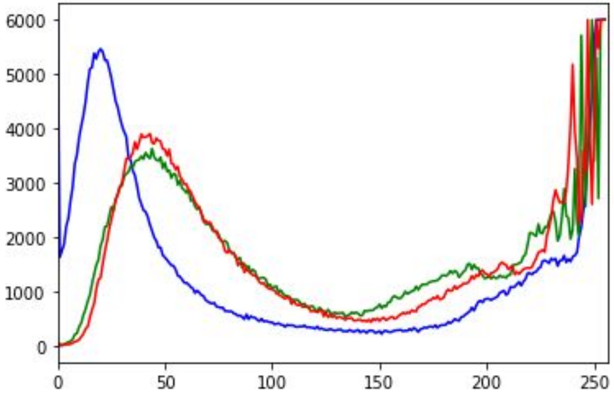
FEATURE EXTRACTION II

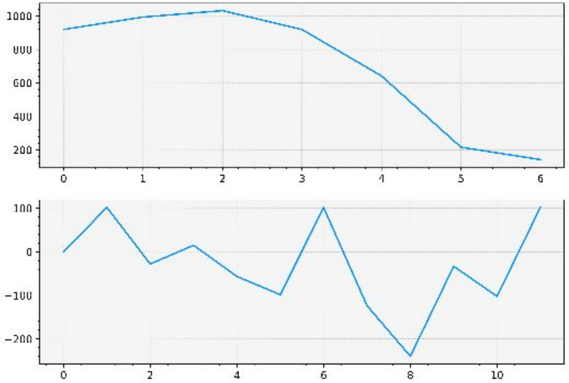




FEATURE EXTRACTION III

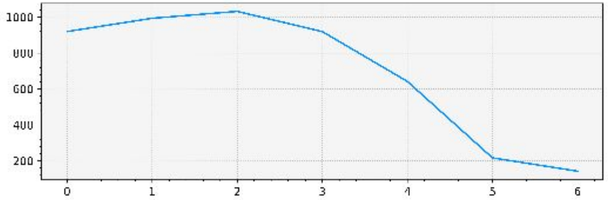
- The histograms were further compressed into relevant features using a **Discrete Wavelet Transform** (DWT).
- DWT decomposes a signal into several frequency **sub-bands**.





FEATURE EXTRACTION IV

- We extract **features** from the frequency sub-bands of each histogram by simply calculating a list of **statistics**.

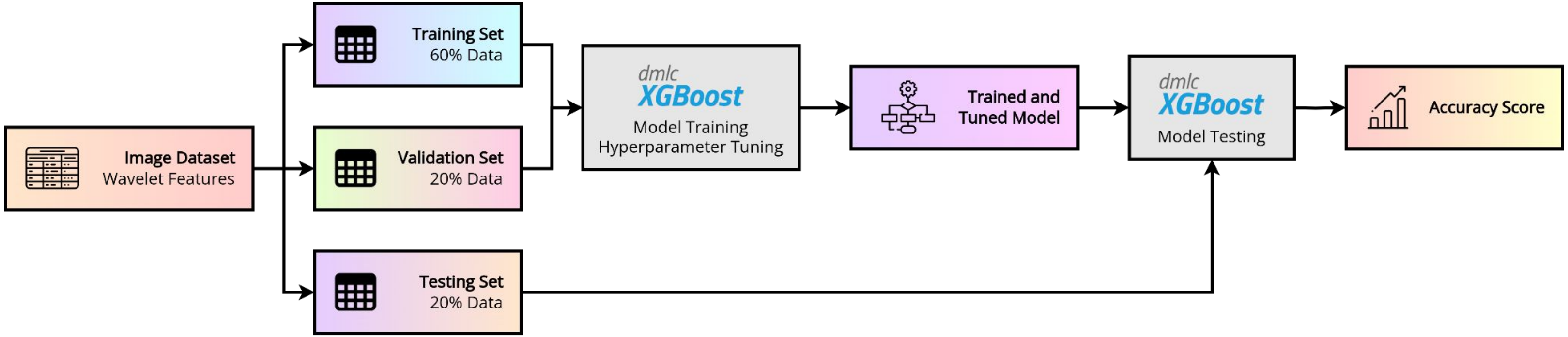



```
array([-3.71069772e-02, -9.65377053e-02, -6.93538747e-02,
       -1.77194182e-01,  4.74088527e-02 , 4.07279452e-02,
       -1.12348408e-01,  9.32261114e-02 , 1.91308254e-01,
       -2.14786885e-01, -1.20654183e-02, ...)]
```

```python
def calculate_statistics (array):
    n5  = np.nanpercentile(array, 5)
    n25 = np.nanpercentile(array, 25)
    n75 = np.nanpercentile(array, 75)
    n95 = np.nanpercentile(array, 95)
    med = np.nanpercentile(array, 50)
    men = np.nanmean(array)
    std = np.nanstd(array)
    var = np.nanvar(array)
    rms = np.nanmean(np.sqrt(array ** 2))
    return [n5, n25, n75, n95, med, men, std, var, rms]
```

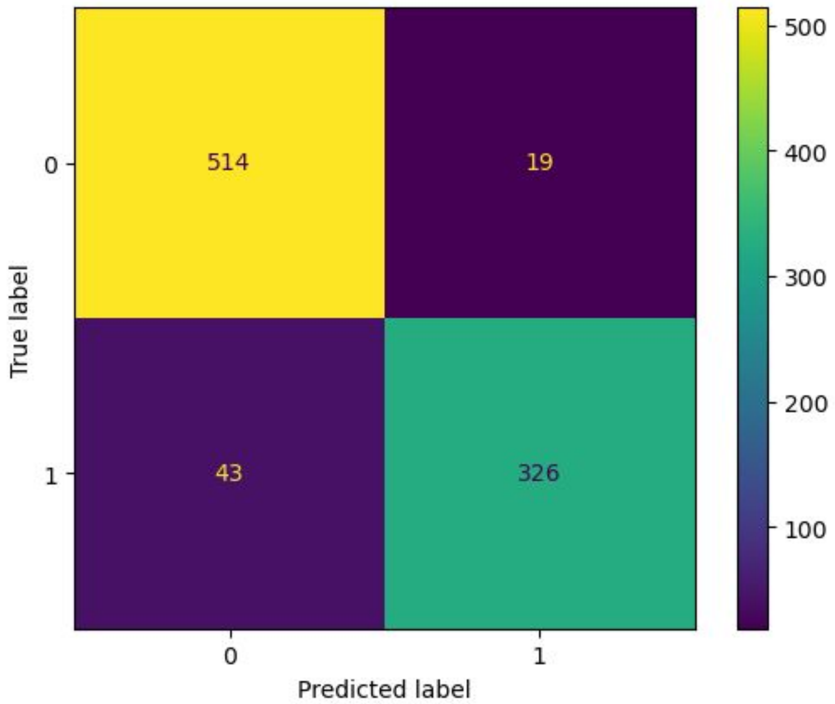
MODEL TRAINING

- Using the generated features, we trained an **XGBoost model**.
- XGBoost stands for *eXtreme Gradient Boosting*, and is a decision tree, ensemble-based model that employs boosting to combine multiple weak decision trees to create a strong model.



RESULTS

- Our model achieved **93.13% accuracy** over the testing set.



REAL-TIME PREDICTION I

REAL-TIME PREDICTION II

CONCLUSIONS

- The use of histograms as representatives for fire and non-fire images is an effective strategy that allows models to perform accurate binary classification.
- This is complemented by a motion detection algorithm for real-time predictions on video cameras, which can be useful for automated forest fire detection.

QUESTIONS?