Machine Learning Project work: “Real-Time Fire Detection on the Edge”

Mario Vento, Pasquale Foggia
and Diego Gragnaniello

SEPTEMBER 11-15•2023 UDINE ITALY

ICIAP 2023

UDINE

ONFIRE International Contest!!!

- This very same contest has been accepted by the **22nd International Conference on Image Analysis and Processing** (Udine, September 11-15)
- International research teams will participate (submit their technique) and present their method
- The **student team ranking first will be invited to participate in the conference** to present their solution

The task to address

- The aim of this project is to detect FIRE in videos acquired by video surveillance cameras
  - Static camera
  - Classify a sequence of frames (images) as either (P)ositive (fire) or (N)egative (no fire)
  - Fire can occur at a specific frame that must be detected

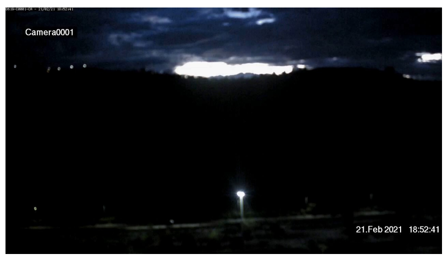
N

P

The task to address

- Videos are acquired by different cameras at different resolutions/frame rate
    - May contain over-imposed text

N

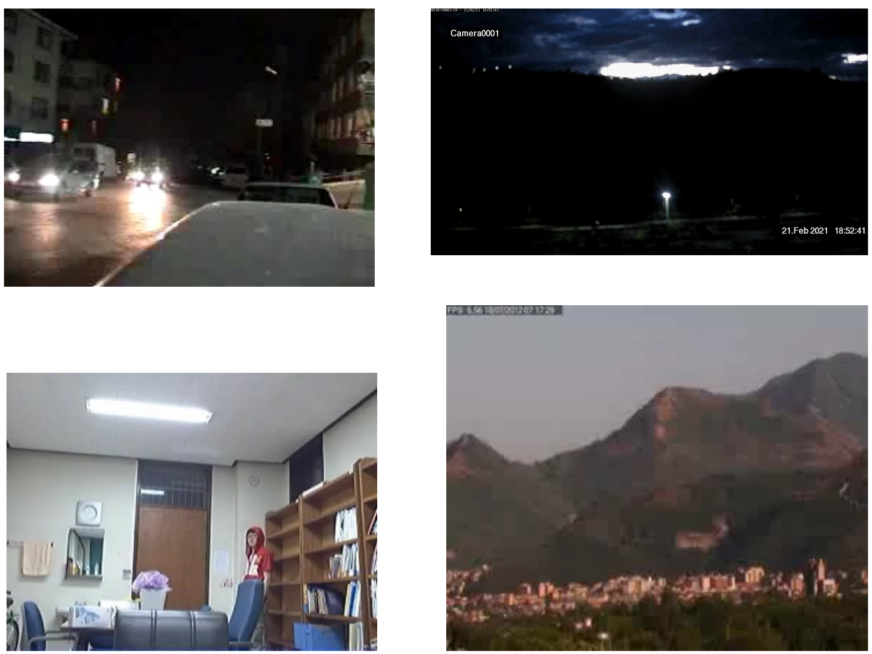


P

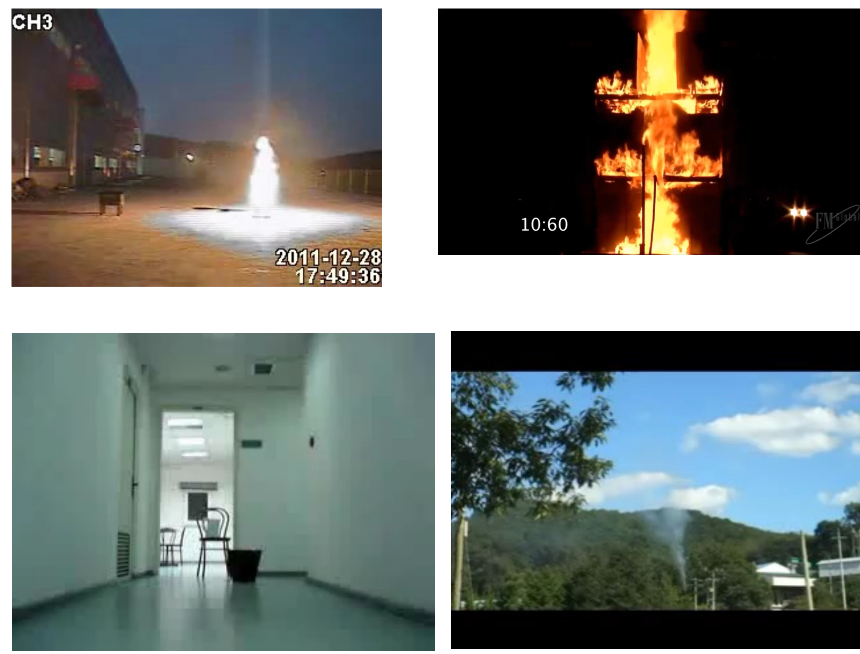
The task to address

- Videos are acquired during daylight/night, indoor/outdoor and by different distances
  - Note: at a distance, the smoke only may be visible

N



P



The task to address

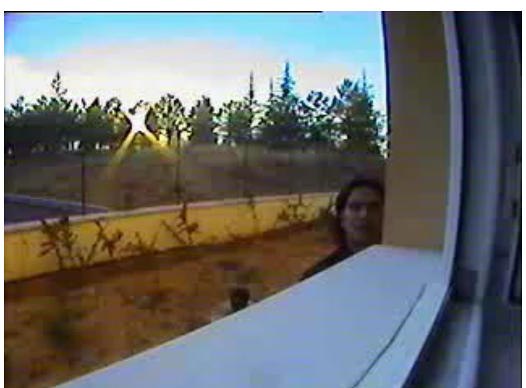
- Fire may not be visible in the first frames
  - You have to provide the first frame in which it is visible

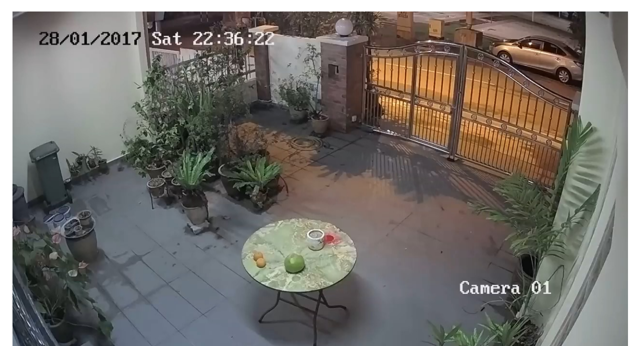
The task to address

- Videos may contain moving objects

The task to address

- Videos may contain Fire-like (yellowish-reddish) or smoke-like (fog, clouds) objects





The training set

- 330 videos:
  - 246 FIRE
  - 84 NO FIRE
- We will provide the link to online datasets from which these are selected
- You can expand it with more data of your choice

IMPORTANT: Model training and validation is not a feed-forward process

- Alternate trainings and validations:
  - After each validation, try to understand which samples are misclassified and why
  - Change your model and/or training set to improve the performance

IMPORTANT: Model training and validation is not a feed-forward process

- Alternate trainings and validations:
  - If the performance seems very good, be sure that the validation set is challenging enough
  - Keep track of your countermeasures/improvements for the final project presentation

What you can use

- You can extend the provided training set
- You can use data augmentation tecniques
- Any algorithm (whatever kind of classifier, including non-neural ones, preprocessing, training strategy, validation)
  - But you must be able to explain what you have used
  - It must be **runnable inside Google Colab**
- You can use local computing power (you are not limited to Colab) for the training

Model evaluation

We define:

- g_i the first frame in which the fire visible
- p_i the first frame in which the fire is detected
- Δt a guard time equals 5 seconds

- True Positive (TP): all the detections in (P)ositive videos for which $p \geq max(0, g - \Delta t)$

- False Positive (FP): all the detections occurring at any time in (N)egative videos or in (P)ositive videos for which $p < max(0, g - \Delta t)$

- False Negative (FP): the set of positive videos for which no fire detection occurs.

Model evaluation: the contributions to the final score

- The final score is obtained by combining two different contributions:

1. Detection performance metrics:
   - Precision
   - Recall
   - The delay between the fire ignition and detection

2. Complexity:
   - The processing capabilities in terms of frame rate
   - The GPU memory required

Model evaluation: the contributions to the final score

- The final score is obtained by combining:
  - Precision
  - Recall

$$\mathrm{P} = \frac{|TP|}{|TP| + |FP|}$$

$$R = \frac{|TP|}{|TP| + |FN|}$$

Model evaluation: the contributions to the final score

- The final score is obtained by combining:
  - Precision
    $$P = \frac{|TP|}{|TP| + |FP|}$$
  - Recall
    $$R = \frac{|TP|}{|TP| + |FN|}$$
  - Detection delay

Max delay: 60 seconds

$$D = \frac{\sum_{i=1}^{|TP|} d_i}{|TP|}, \quad d_i = |p_i - g_i|, \qquad D_n = \frac{max(0; 60-D)}{60}$$

Time instant when fire is predicted

Time instant when fire is visible

Model evaluation: the contributions to the final score

- The final score is obtained by combining:
  - Precision
  - Recall
  - Detection delay
  - Processing frame rate
  - Memory occupancy

$$\mathrm{P} = \frac{|TP|}{|TP| + |FP|}$$

$$R = \frac{|TP|}{|TP| + |FN|}$$

$$D = \frac{\sum_{i=1}^{|TP|} d_i}{|TP|}, \quad d_i = |p_i - g_i|, \qquad D_n = \frac{max(0; 60-D)}{60}$$

$$PFR = \frac{1}{\frac{\sum_{i=1}^{|N|} t_i}{|N|}}, \qquad PFR_{delta} = \max\left(0; \frac{PFR_{target}}{PFR} - 1\right)$$

$$MEM_{delta} = \max\left(0; \frac{MEM}{MEM_{target}} - 1\right)$$

Penalize:

memory occupancy greater than MEM_{target}

proc. frame rate lower than PFR_{target}

Model evaluation: the contributions to the final score

- The final score is obtained by combining:
  - Precision
  - Recall
  - Detection delay
  - Processing frame rate
  - Memory occupancy

$$\boxed{\mathrm{P} = \frac{|TP|}{|TP| + |FP|} \qquad R = \frac{|TP|}{|TP| + |FN|}}$$

$$D = \frac{\sum_{i=1}^{|TP|} d_i}{|TP|}, \quad d_i = |p_i - g_i|, \qquad \boxed{D_n = \frac{max(0; 60-D)}{60}}$$

$$PFR = \frac{1}{\frac{\sum_{i=1}^{|N|} t_i}{|N|}}, \qquad \boxed{PFR_{delta} = \max\left(0; \frac{PFR_{target}}{PFR} - 1\right)}$$

$$\boxed{MEM_{delta} = \max\left(0; \frac{MEM}{MEM_{target}} - 1\right)}$$

- The final score

$$FDS = \frac{P * R * D_n}{(1 + PFR_{delta}) * (1 + MEM_{delta})}$$

Share the load

- Each member of the team will be requested to submit an estimate of the individual effort contributed by all members
  - To prevent "free riders"
  - Submissions will be "blind" (each member will not see the submissions of other members)

IMPORTANT: Don't forget to

- Write the **names of all the team members** in the Google Drive folder (in a text file)
- Ensure that the **link** you submit is **readable to anyone** (no authorization must be requested)
- Make sure that the **test script** is **compliant with the specification** (if you have doubts about the specification, **ask**)