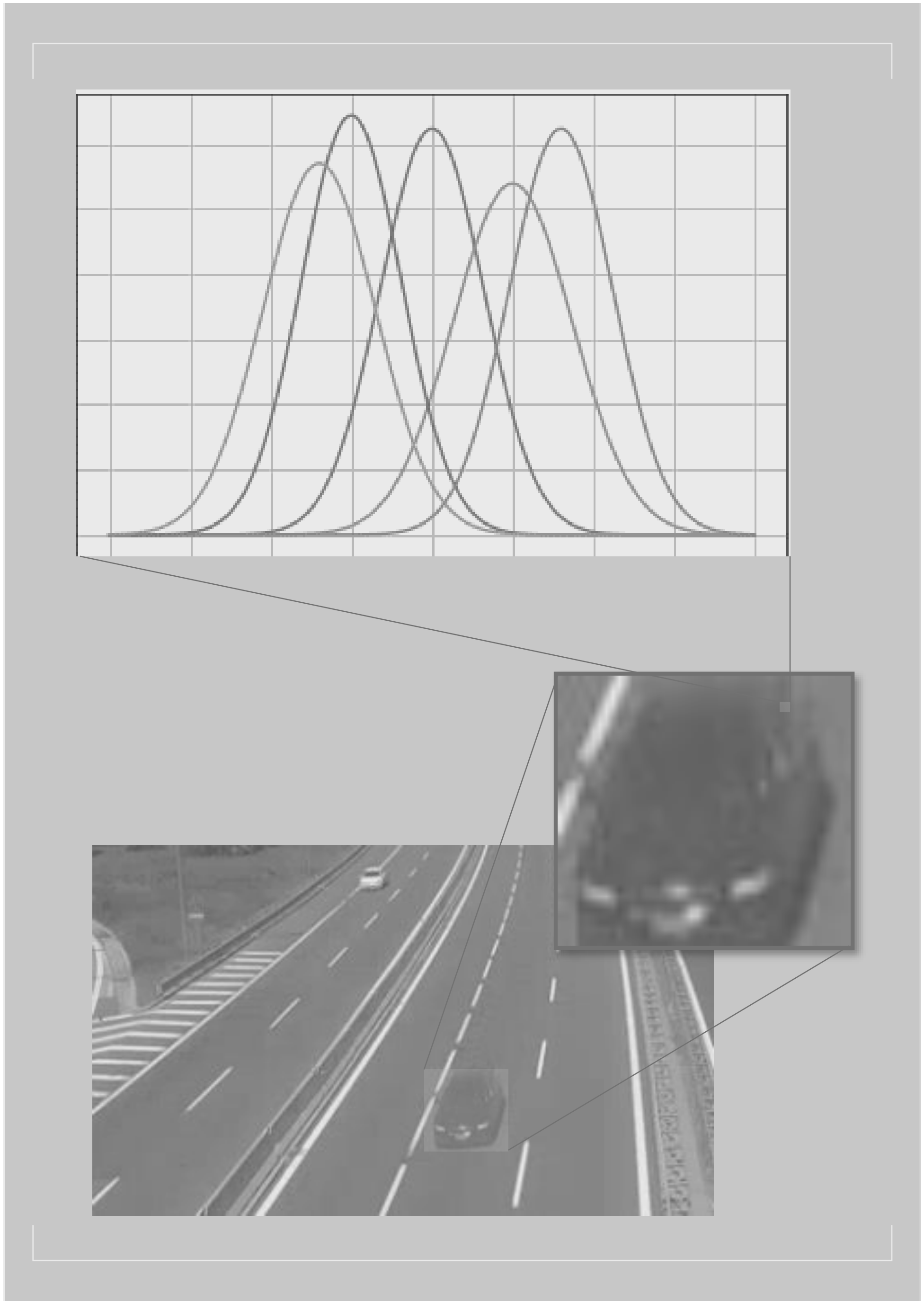
Background subtraction

Adaptive Gaussian Mixture Model

Andrea Mazzeo
Daniele Moltisanti



GMM overview

- For each pixel generates a mixture of gaussian distributions

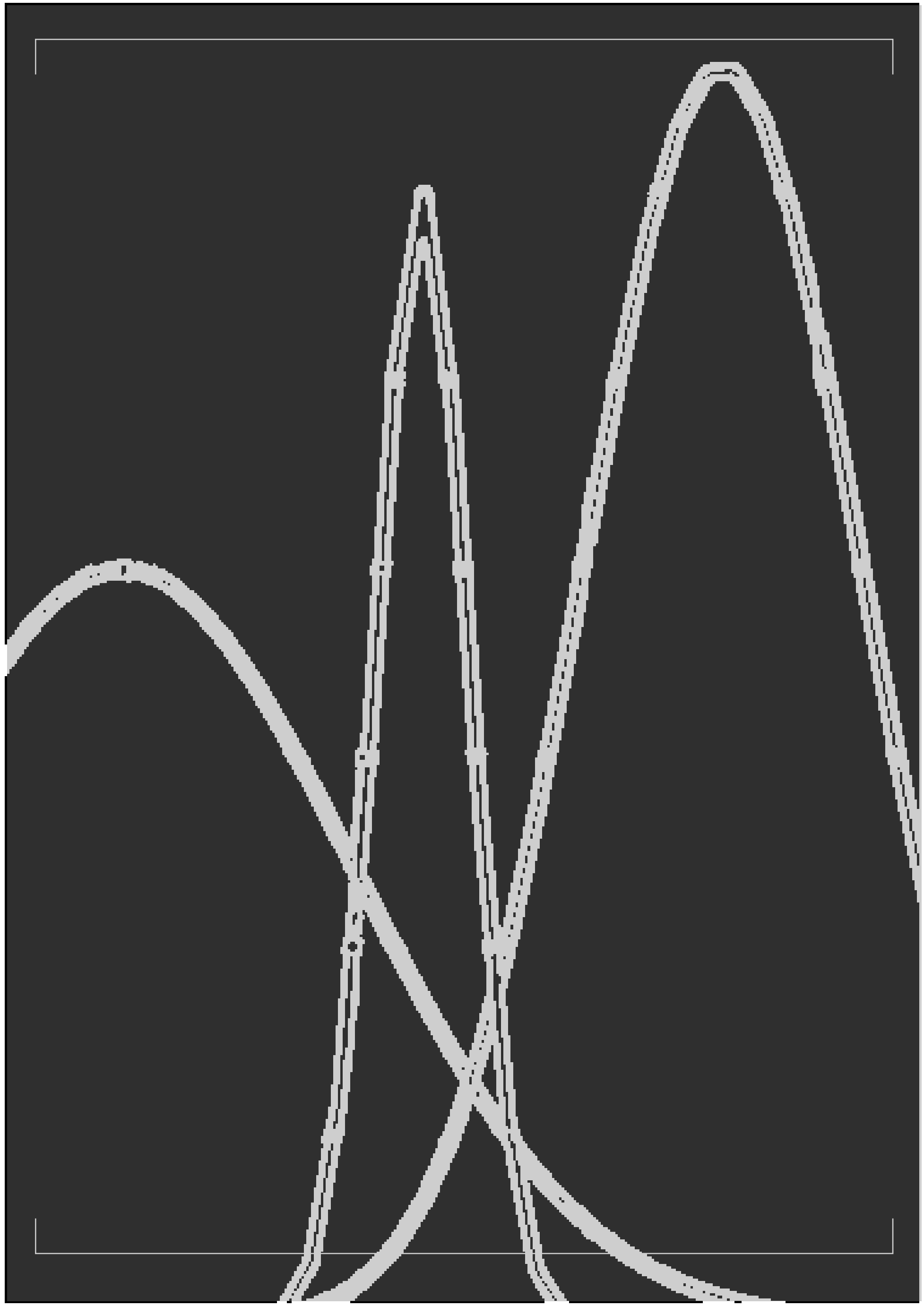
$$p(X_t) = \sum_{i=1}^{k} w_{i,t} * N\left(X_i, \mu_{i,t}, \Sigma_{i,t}\right)$$

- Two steps:
  1. Update the model
  2. Check if pixel belongs to background

Mahalanobis distance

$$\left(\sqrt{(x-\mu)^T * \Sigma(x-\mu)}\right) < T\sigma^{-1}$$

- Two possibile cases:
  - Match with one gaussian
  - No match with any gaussian



Towards our implementation

- Slow updating in transition scenario

- Two parameters usefull:
  - **Threshold**

    Allows to manage the foreground detection.
  - **Learning rate**

    Allows to manage the learning capability.

Threshold example

- **THRESHOLD VALUE = 20**
- Never used, due to its instability
- Result few accurate

Threshold example

- **THRESHOLD VALUE = 70**
- Foreground detected highly accurated
- Subject to instability
- Default value of OpenCV function
- Used in daily scenario

Threshold example

- **THRESHOLD VALUE = 140**
- More stability
- Allows to reduce lights effects
- Less realistic foreground detection
- Used in night scenario

Learning rate example

- **LEARNING RATE VALUE = 0,001**
- Learning too slow
- Used as lower bound
- Used in stable scenario:
  - Day
  - Night

Learning rate example

- **LEARNING RATE VALUE = 0,05**
- Learning fast
- Used as higher bound
- Used in transition scenario:
  - Sunset
  - Sunrise

Learning rate example

- **LEARNING RATE VALUE = 0,1**
- Learning too fast
- Never used

The main problem

Manage GMM parameters in automatic way.
Regularized as a function of time.

Our idea

- Split the time in three scenario:
  - Day time
  - Night time
  - Transition: Sunrise, Sunset.
- According to each scenario, GMM parameters have to be set.
- Day:
  - Normal threshold – Low learning rate
- Night:
  - High threshold – Low learning rate
- Transition:
  - Normal threshold – High learning rate

Our implementation

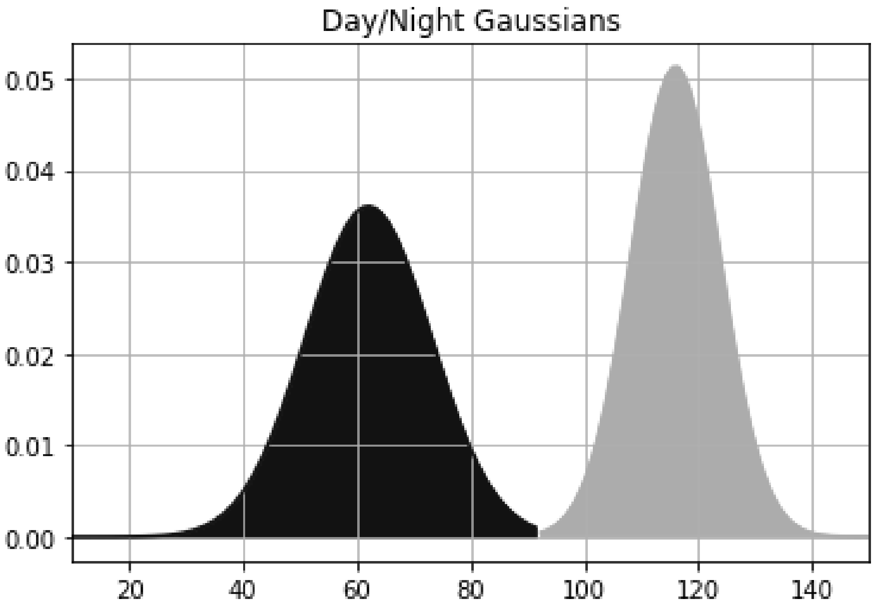
Training phase

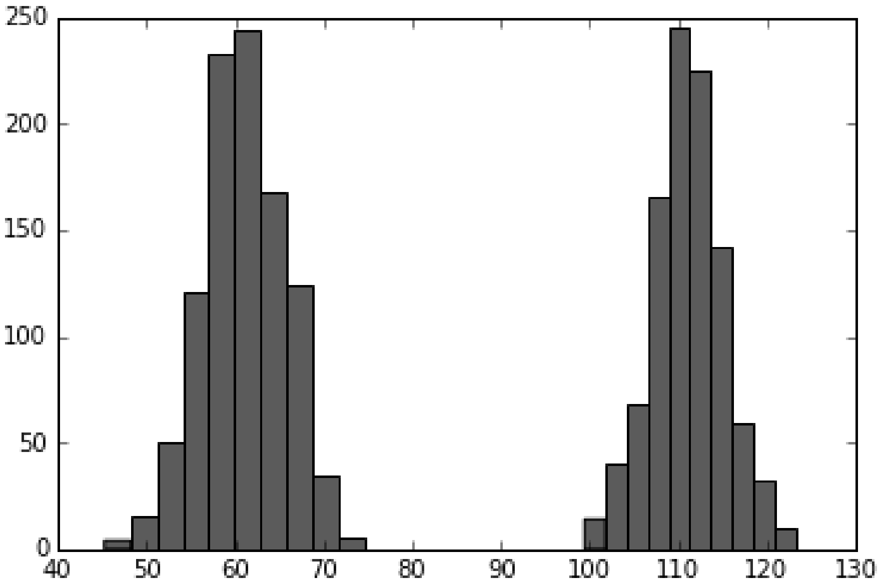
Background subtraction

Results analysis

Training phase

- The goal is to acquire a gaussian model able to descern night and day.
- The model is build starting from the average of pixel value.
- The training phase is universal. It provides a general model valid for each webcam.





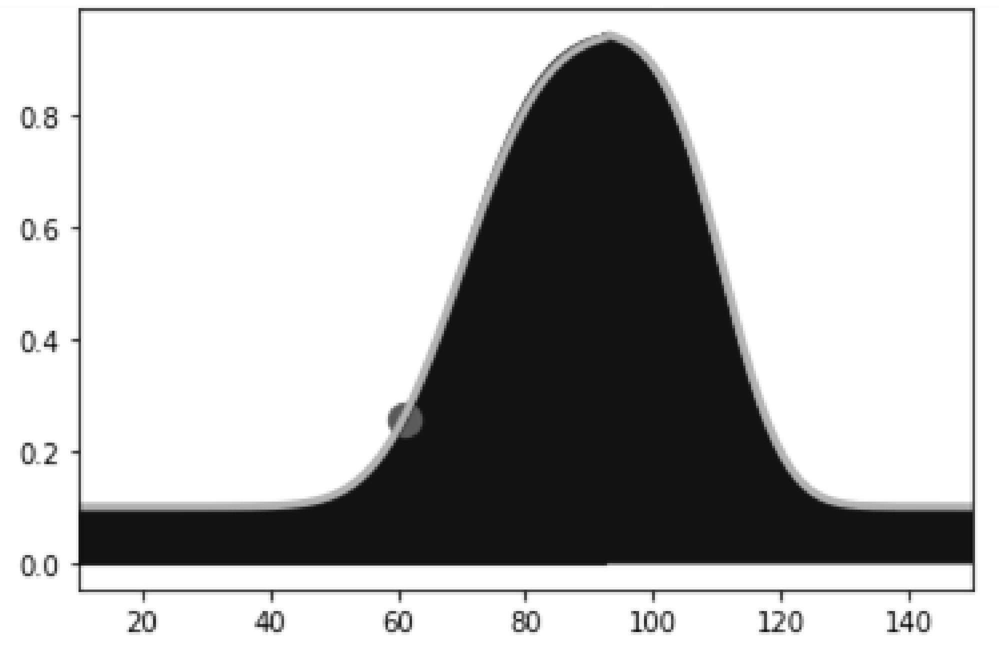
Training phase

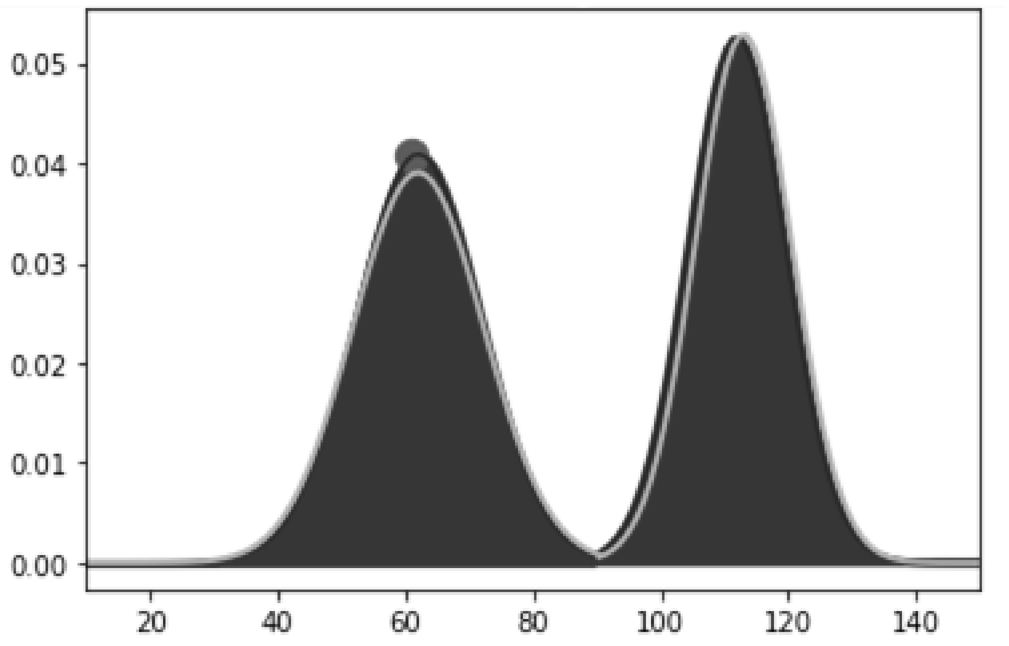
How the average is computed?

- Taking 500 points from 5 squares
- Random points with uniform distribution
- Able to cover all image

Background subtraction

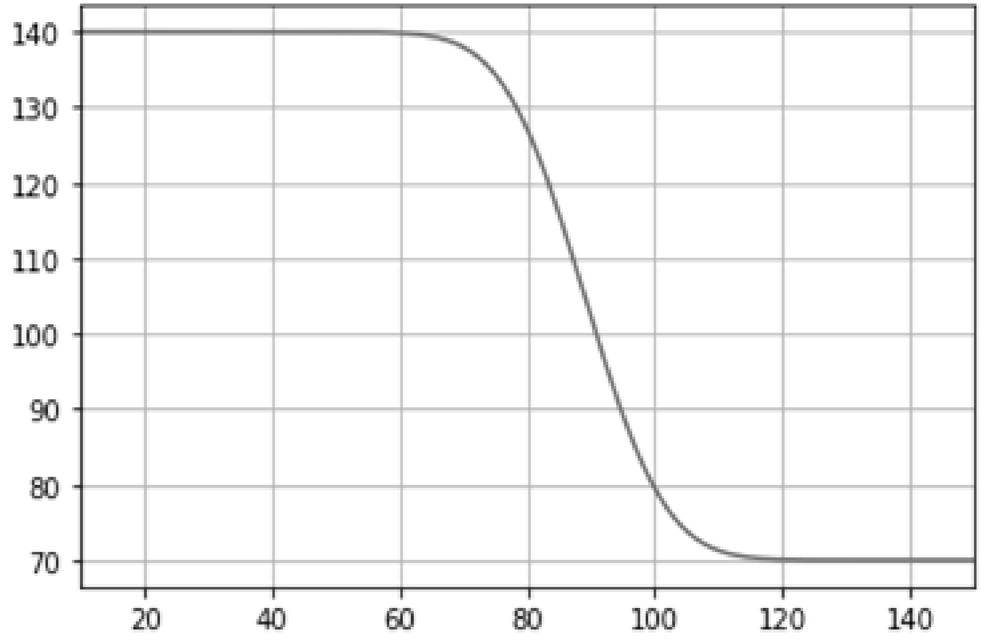
1. Each five frames takes one frame.
2. Evaluate the average.
3. Understand the scenario (night, day, transition).
4. Regulate the GMM parameters.
5. Update the gaussians.



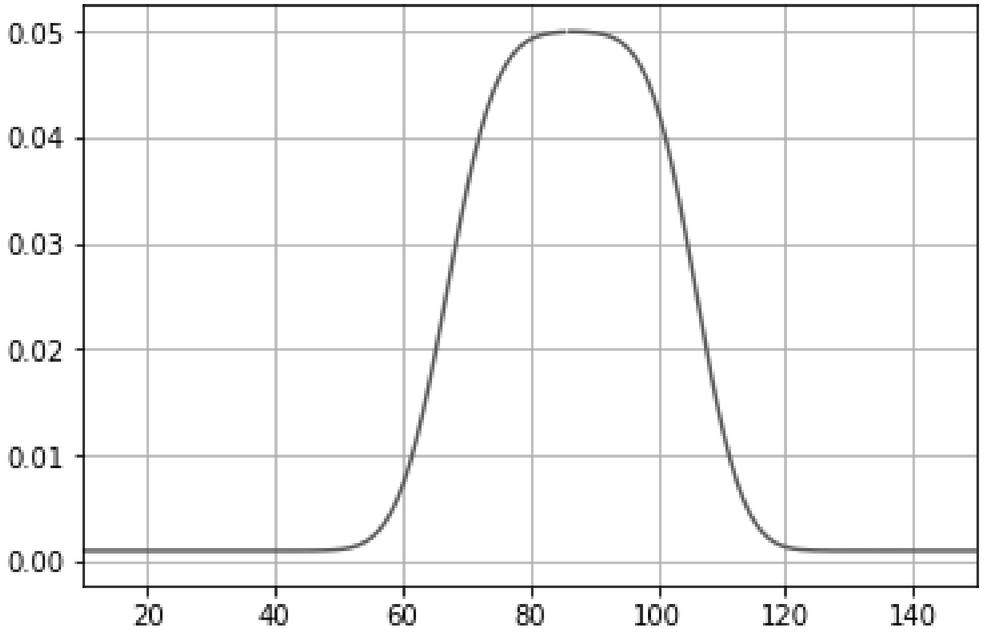


Background Subtraction

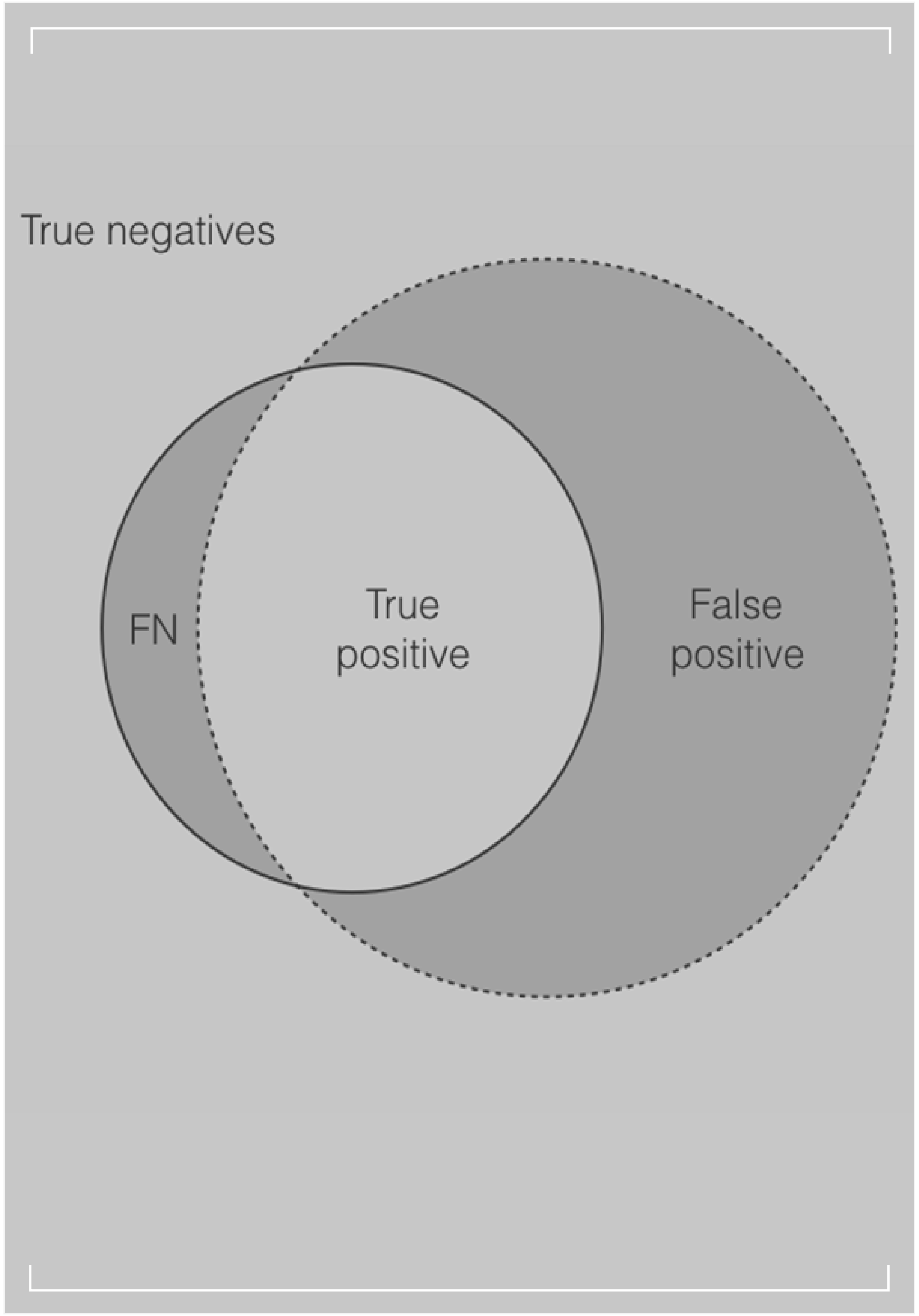
Update GMM parameters



THRESHOLD



LEARNING RATE



Results analysis

Compare ground truth and model identified:

- If pixel is classified as foreground in both → ***True Positive***
- If pixel is classified as background in both → ***True Negative***
- If pixel is classified as foreground in ground-truth and background in model identified → ***False Negative***
- If pixel is classified as background in ground-truth and foreground in model identified → ***False Positive***

Results analysis

Evaluation metrics

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy

$$\frac{TP}{TP + FP}$$

Precision

$$\frac{TP}{TP + FN}$$

Recall

$$2\,\frac{Recall \times Precision}{Recall + Precision}$$

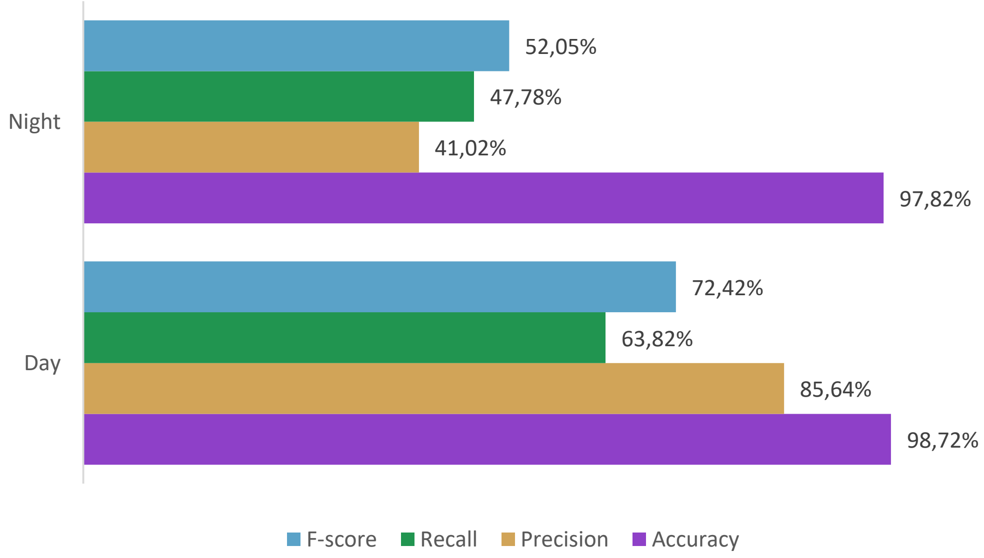
F1 score

GROUND-TRUTH

MODEL IDENTIFIED

Results analysis

Average results:

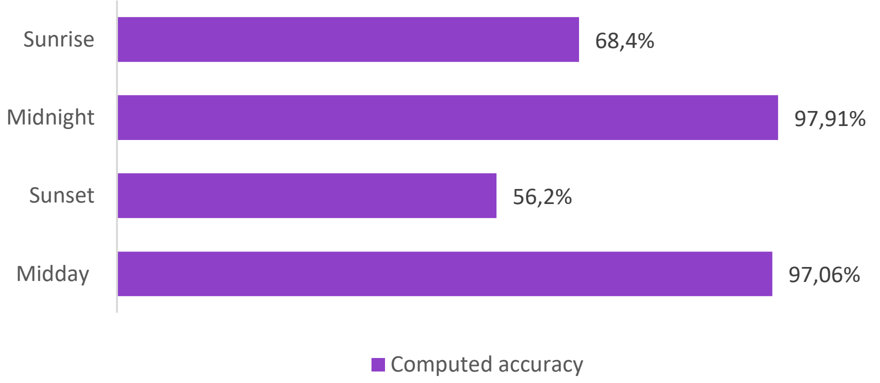


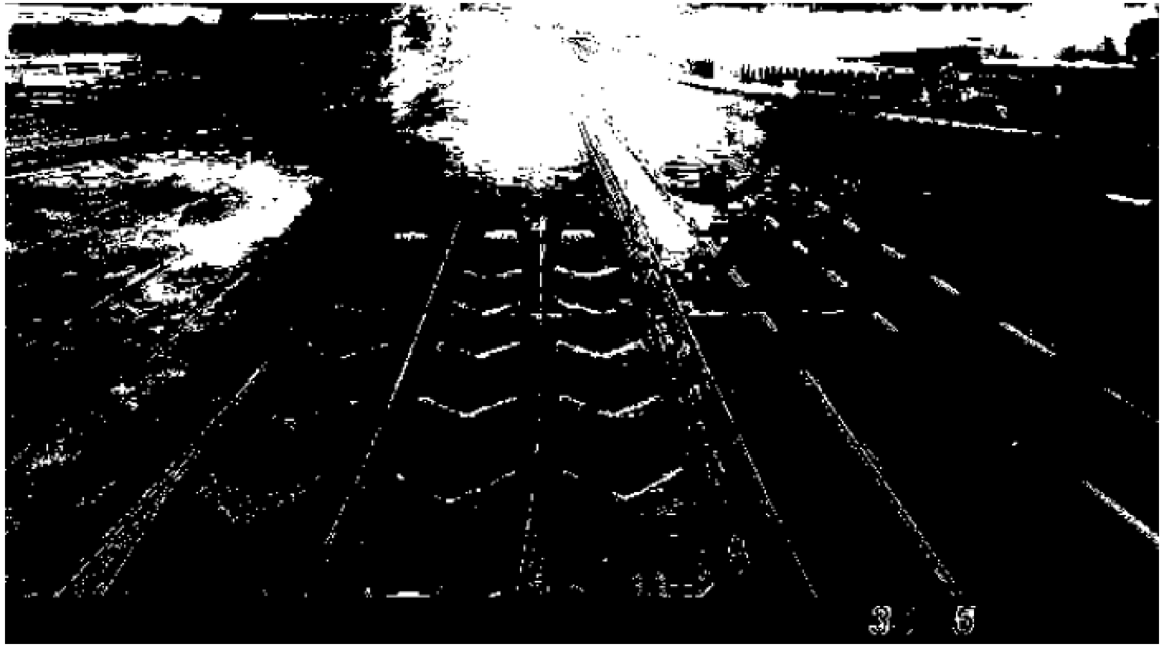
BACKGROUND DETECTED

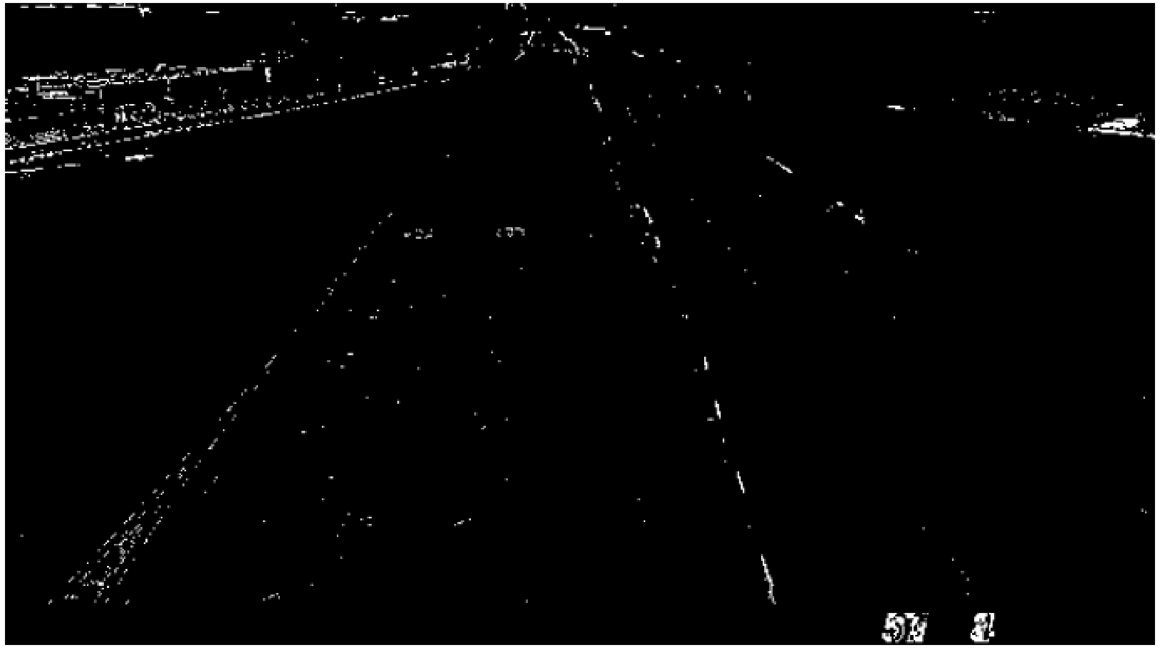
ORIGINAL FRAME

Results analysis

- Check pixel per pixel differences between the two images
- Use a little threshold
- Results:







Results analysis

- From detected background and original frame extract a matrix made up by differences
- Create an image starting from this matrix
- This image represents the error

Thank you

Andrea Mazzeo
Daniele Moltisanti