Dpt Informatique

Project report: Introduction to image analysis

Realized by:

- Doriane AJMI p1616047
- Mohamed MAKHLOUF p1808420
- Ilyess BESSAADI p1707782

Supervised by:

- Saida BOUAKAZ BRONDEL - University Professor

1)Introduction

This project is part of the UE Introduction to image analysis of the M1 computer science of the University Claude Bernard Lyon 1 supervised by Mrs Bouakaz Saida.

Image analysis is the recognition of elements and information contained in an image. It can be automated when the image is recorded in digital form, using computer tools. The tasks related to image analysis are multiple, from barcode reading to facial recognition.
The use of computers allows the automation of the analysis of the content of images.
These softwares implement algorithms such as convolutions, thresholding, operations (addition, subtraction, multiplication, division) of images, binary operations of images... The programming of these operations allows the user to carry out quickly the analysis of a great number of images according to his problematic.

The project is divided into 3 parts:
- Basic image processing : a first exploratory part which allows to get familiar with OpenCV tools and to get familiar with filters and their effects on images.
- Background removal : a second part which consists in doing background extraction on a video.
- Region growing : a last part which allows to do segmentation using the region growing method on a character image to assign a region to each body part.

2) Basic image processing

This part aims to make us discover the tools proposed by OpenCv which are useful for image analysis. We made the choice to realize this part in a jupyter notebook in order to appreciate the comfort of display and comparison that this tool offers us. We were thus able to alternate the cells of code, of markdown to make explanations and to preserve the results of each of our tests in output display. We provide in annex of this report the content of our notebook.

3)Background removal

a) Introduction

In the field of computer vision, the detection consists in perceiving a static or dynamic scene, and to know the appropriate change of movement, thus deducing the kinematics of all moving objects with respect to the camera. Detection is a difficult and crucial task, it usually depends on the nature of the scene and the corresponding application.
Background subtraction is a widely used approach for the detection of moving objects in video sequences taken from a fixed camera. The basic idea is to detect the difference between the current frame and the reference frame, often called the background image, or the background model.

Several techniques are used to implement the idea of background subtraction. Among the most known are:
- **The image subtraction technique.**
- **The adaptive temporal difference technique.**

b) The image subtraction technique

The simplest way to implement this algorithm is to take an image as a background image and take the frames obtained at time t, which are noted I(t) and compare them with the background image noted B using simple arithmetic calculations, we can segment the objects simply by using the subtraction technique for each pixel in the image I(t). By taking the value of the pixel noted by P(I(t)) and subtracting it from that of the pixel corresponding to the same position in the background image noted by BG.

Although we have now removed the background, this approach will only work for cases where all background pixels are static. A threshold" is applied for this difference between the background and the current image to improve the subtraction. This means that the pixel intensity of the difference is thresholded or filtered by the threshold value.

The efficiency of this technique will depend on the speed of motion in the scene. Faster movements can necessitate a higher threshold.

Advantages:

- The image difference technique is easy to implement.
- The execution time is relatively the lowest compared to the other techniques.
- The computational cost is minimal due to the simplicity of its algorithm.

Disadvantages :

- The background must be static which is a bit unrealistic.
- Sensitive to changes in lighting conditions.
- Sensitive to geometrical changes of the image.

c) The adaptive temporal difference technique

The background adaptive temporal difference technique fills the gap of the image subtraction technique. Where we Initialized at the beginning the background and stays fixed throughout the sequence. The adaptive background time difference technique corrects this issue by updating the background on predefined time steps.

Most popular methods are based on Gaussian mixture models (GMM). Four methods based on GMM were used: GMG, KNN, MOG, MOG2. These few popular background subtraction methods are implemented in known open computer vision library called OpenCV

-GMG :

The GMG method is based on study [1]. It employs a probabilistic foreground segmentation algorithm that identifies possible foreground objects using Bayesian inference. The estimates are adaptive, newer observations are more heavily weighted than old observations to accommodate variable illumination, several morphological filtering operations like closing and opening are done to remove unwanted noise.

-KNN(K-nearest neighbours):

This method is based on study [2]. They presented recursive equations that are used to constantly update the parameters of a Gaussian mixture model and to simultaneously select the appropriate number of components for each pixel. They implemented KNN method for better kernel density estimation [2].

-MOG(Mixture of Gaussians):

MOG is based on study [3], Briefly It uses a method to model each background pixel by a mixture of K Gaussian distributions (K = 3 to 5). The weights of the mixture represent the time proportions that those colours stay in the scene. The probable background colours are the ones which stay longer and more static. It was based on a computational color space that makes use of their background model. Thus, the best result could be obtained on color video sequences [3].

-MOG2((Mixture of Gaussians 2):

This method is based on study [4]. Their adaptive algorithm using Gaussian mixture probability density. Recursive equations were used to constantly update the parameters and also to simultaneously select the appropriate number of gaussian distributions for each pixel. It provides better adaptability to varying scenes due illumination changes etc. As in the previous case, we have to create a background subtractor object. Here, you have an option of selecting whether shadow to be detected or not. If detectShadows = True (which is so by default), it detects and marks shadows, but decreases the speed. Shadows will be marked in gray color.

While coding, we need to create a background object using the function:

$$cv2.createBackgroundSubtractor[NAMEOFTHEMODEl]()$$

It has some optional parameters depending on which background subtractor is used,it is all set to some default values (see Figure 01). Then inside the video loop, use the ‘backgroundsubtractor.apply()’ method to get the foreground mask. It has also a parameter named Learning rate.(See **Figure 01** for explication).

Background subtractor	Parameters of the constructor and default values
GMG	[initialization Frames , decisionThreshold]
KNN	[history , dist2Threshold , detectShadows]
MOG	[history , nmixtures=5, backgroundRatio , noiseSigma]
MOG2	[history , varThreshold , detectShadows]

Parameter name	Explication
history	It denotes the number of frames to be used to model the background.
var Threshold and dist2 Threshold	Threshold on the squared distance between the pixel and the sample to decide whether a pixel is close to that sample. This parameter does not affect the background update. briefly is a threshold to define whether a pixel is different from the background or not. The smaller the value is, the more sensitive movement detection is. And vice versa.
initialization Frames	indicates how many frames the algorithm is going to use to initialize the

	background-modeling method.
decision Threshold	determines the threshold in which pixels are classified as background or foreground
n mixtures	indicates the method how many Gaussians distributions should have during the whole video. Higher values drastically increase processing time.
backgroundRatio	defines the threshold weight for the differentiation between foreground and background.
detect Shadows	If true, the algorithm will detect shadows and mark them. It decreases the speed a bit, so if you do not need this feature, set the parameter to false.
noiseSigma	defines the accepted noise level

Postscript :
When we use the apply() function we have to put in parameter the frame we are trying to subtract it's background, also we can set the learning rate.
learning Rate : value between 0 and 1 that indicates how fast the background model is learnt. Negative parameter value makes the algorithm use some automatically chosen learning rate. 0 means that the background model is not updated at all, 1 means that the background model is completely reinitialized from the last frame.

Figure 01

Advantages :

- One obvious advantage is the modest computational load.
- The other is that the basic model is very adaptive.

Disadvantages:

- Objects with uniformly distributed intensity values, the interior pixels will be interpreted as part of the background.
- Another problem is that the objects must be continuously moving. If an object stays immobile for more than one frame period, it will be part of the background.

d) Post-traitement of the foreground

After several tests of the methods mentioned in the previous section we noticed that the mask does not cover the whole object, this is due to the noise efficiency ratio of the method. The morphological processing of the image counteracts this problem. The detection of the object in the foreground will be improved by the use of two morphological filters, erosion and dilation in order to solve the problems often caused by occlusions.

First a preliminary closing operation is applied from size [5 5], then several successive and alternate erosion and dilation processes in such a way that, we use a (3,3) matrix of ones for the erosion and the dilation kernel. And finally we apply a closing operation of size [10 10].

e) Replacement of the background of the image

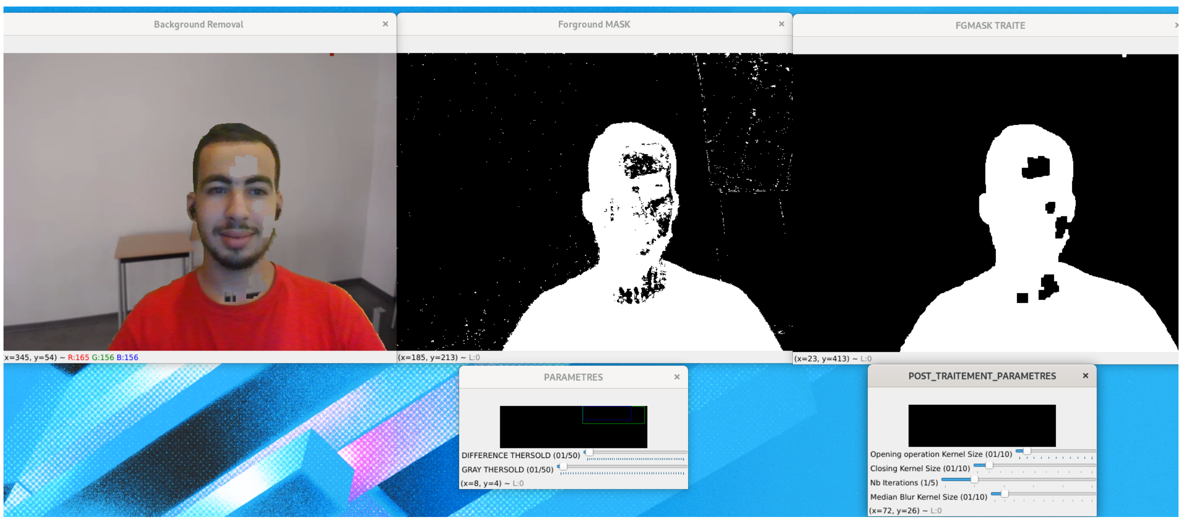
One time we have a good foreground mask FGMASK, to replace the background we calculate the inverse of this mask using the predefined function opencv.bitwise_not called BGMASK.
Then we try to get the foreground from our image by doing a bitwise_and operation between our image and the foreground mask FGMASK, we do the same operation another time between our new background and the inverse of the foreground mask BGMASK.
Our final result will be the addition of the last two results (a part of our current frame and a part of our new background). We can do this operation using the function add of opencv.

f) Implementation and results

We decided in the implementation and the test phase to use the two models available in the last version of Opencv, So we are going to test KNN and MOG2 algorithms.

We made a “Trackbar” with opencv to facilitate the modification of the different parameters of our models. Thus the parameters of our post traitement operations.

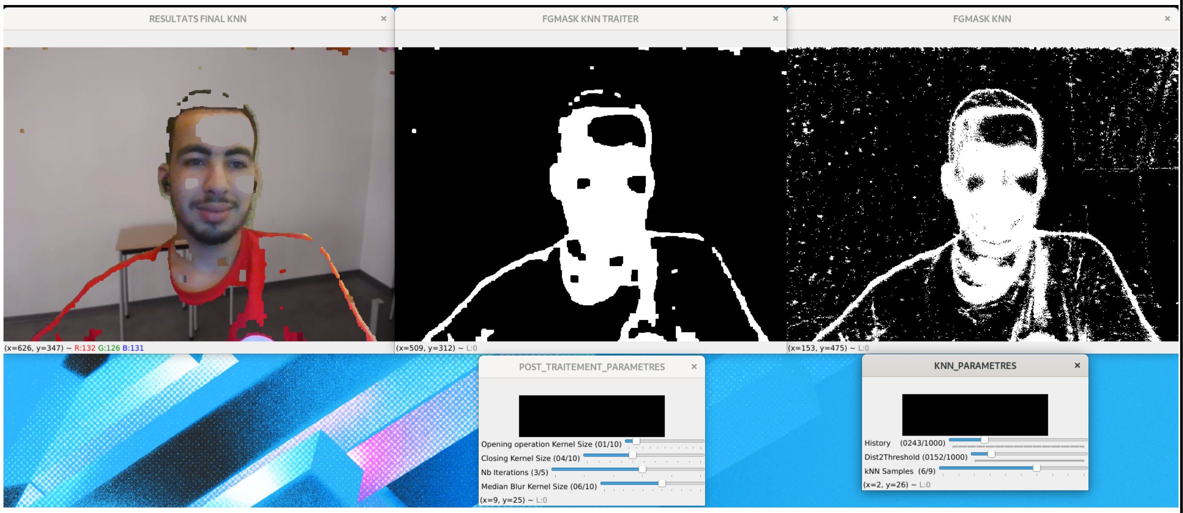
You will find in the following screenshots of the results and we will meet on the day of the presentation for a demonstration.



DIFFERENCE THERSOLD (01/50)
GRAY THERSOLD (01/50)
(x=8, y=4) ~ L:0

Opening operation Kernel Size (01/10)
Closing Kernel Size (01/10)
Nb Iterations (1/5)
Median Blur Kernel Size (01/10)
(x=72, y=26) ~ L:0

Result of the image subtraction technique



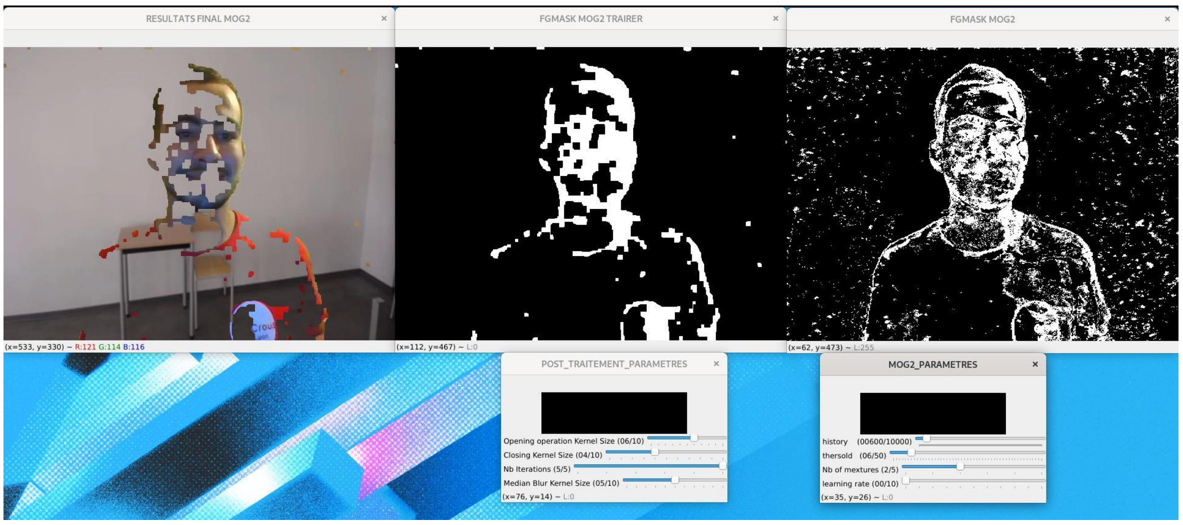
POST_TRAITEMENT_PARAMETRES

Opening operation Kernel Size (01/10)
Closing Kernel Size (04/10)
Nb Iterations (3/5)
Median Blur Kernel Size (06/10)
(x=9, y=25) ~ L:0

KNN_PARAMETRES

History (0243/1000)
Dist2Threshold (0152/1000)
kNN Samples (6/9)
(x=2, y=26) ~ L:0

Result of KNN Algorithm



POST_TRAITEMENT_PARAMETRES

Opening operation Kernel Size (06/10)
Closing Kernel Size (04/10)
Nb Iterations (5/5)
Median Blur Kernel Size (05/10)
(x=76, y=14) ~ L:0

MOG2_PARAMETRES

history (00600/10000)
thersold (06/50)
Nb of mextures (2/5)
learning rate (00/10)
(x=35, y=26) ~ L:0

Result of MOG2 Algorithm

4) Region growing

a) Introduction

Segmentation subdivides an image into a constituent regions or objects. The level of details to which the subdivision is carried depends on the problem being solved. That is, segmentation should stop when the object or regions of interest in an application have been detected. Image segmentation is useful in several contexts, such as the detection of anomalies on products in quality control for example, or can help in the detection of pathology such as cancer. Segmentation of nontrivial images is one of the most difficult tasks in image processing.

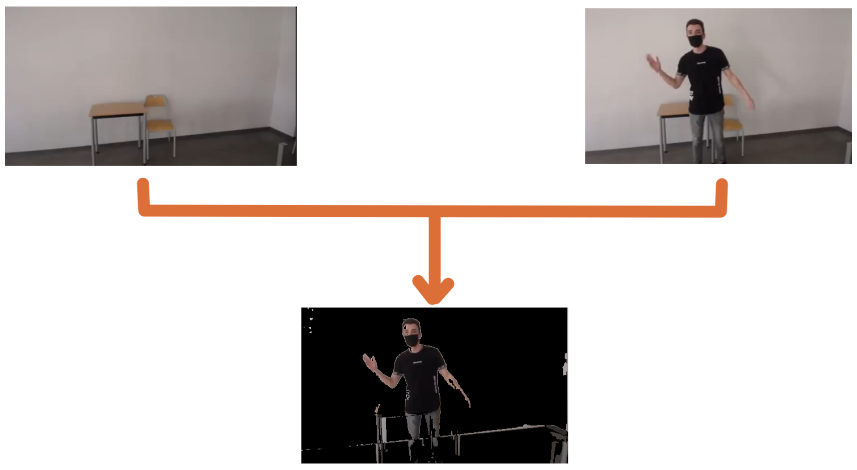
The objective of our project is to make a region segmentation on the different parts of the body of a human character on an image. To do this, we applied the region growing method. In this report, we will explain the different steps realized in our project to realize this segmentation.

b) Steps of the implementation of the project

i) study picture

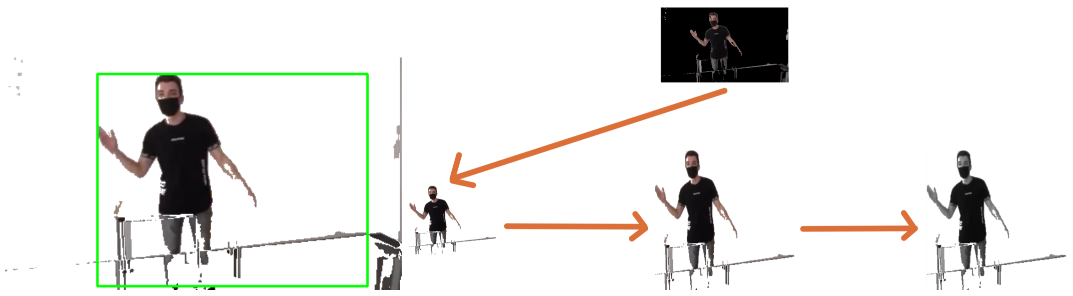
To test our code, we used the image of one of our classmates that was provided in the database for the project.

The first step of our segmentation consists in a background suppression of the image. For this, we have two images: one that corresponds to the "background" which is in fact a blank image of character. The second image has the same background, with the addition of a character. We have made images with different position of the character to make tests to ensure the adaptability of the program. So, in a first step, we try to remove the background of our image to keep only the character. To do this we use the "substract" function of openCV which calculates the foreground mask by performing a subtraction between the current image and a background model, containing the static part of the scene or, more generally, everything that can be considered as a background given the characteristics of the observed scene.



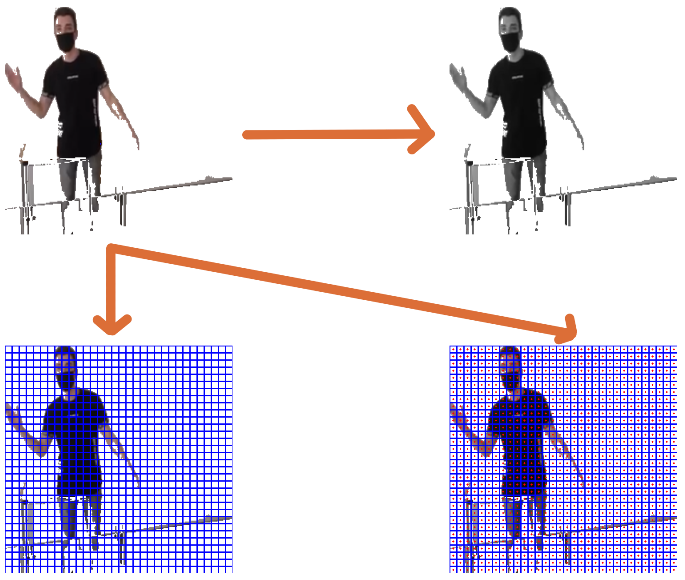
ii) bounding box

This step consists in keeping only the element of interest on our image : thus the character. To do this, we build bounding boxes and we select the one that encompasses our character. To do this we convert our image into coarse levels and then we apply the Otsu threshold *(ref wikipedia : In computer vision and image processing, the Otsu method is used to perform automatic thresholding based on the shape of the image histogram1, or the reduction of a grayscale image to a binary* *image.)* to obtain a binary image. This allows us to find the contours, then we use the boundingRect() function of openCV to obtain the coordinates of the bounding box, we can draw it with the rectangle() function of openCV to visualize it but also cut the image to keep only the content of the bounding box.



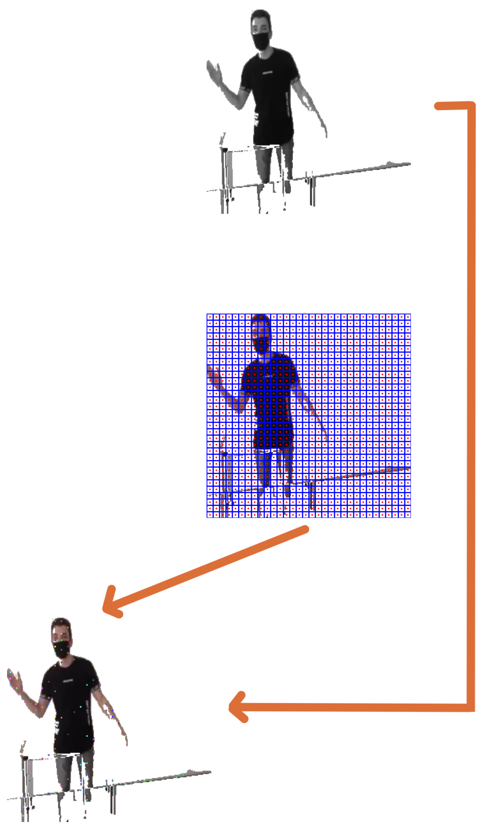
iii) Construction of a grid and seed placement

At first we placed our seeds arbitrarily "by hand" in our code but then we made sure to automate this placement. To do this, we first built a grid on our image. This will allow us to study defined portions of the image, which will help us for the next step. Before, our image has been converted to gray and resized to facilitate the construction of the grid. Following the construction of the grid, we have placed a seed in the center of each of our cells. This makes a seed pose which is very systematic and not very adapted to our problem. That's why we realize a reduction of seeds.



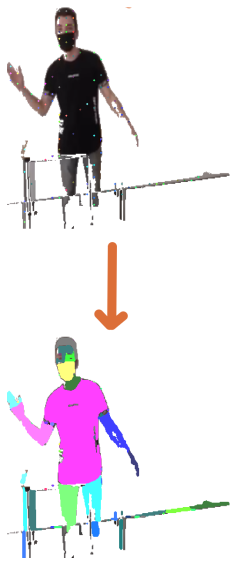
iv) seed reduction

To do this, we define a certain threshold and we calculate the different value of gray levels of 2 seeds that follow. If the result is lower than the defined thresholds, we realize a reduction . This consists in placing a new seed between the two seed which were compared, then to remove them. We start again that as long as there are seeds which are able to merge. We also delete all the seeds that have been deposited in the background (that have a value of 255). So we keep only some seeds that are deposited on the different parts of the character's body.



v) seed expansion and fusion

Each of the seeds will expand to form a zone. In fact, each seed will compare its level of grey with the 8 neighbors which surround it and continue its expansion if the difference does not exceed a certain threshold. Thus, the stop of an expansion is done either by the contact of another region or in the case where the difference exceeds the threshold. It should also be noted that the seed do not expand on white pixels.



5) Conclusion

To conclude, this project allowed us to explore OpenCV tools and their use in the domain of image analysis. This made us realize how difficult it is to get good results. Moreover, from one image to another, the settings necessary for the proper functioning of the algos are not always the same, which prevents us from having universal rules that will work every time but rather try to vary the parameters until finding something satisfying.

6) References

[1] Godbehere A. B.,, Matsukawa A., Goldberg K., Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation, In American Control Conference (ACC), 2012, pages 4305–4312. IEEE, 2012.
[2] Zivkovic Z. F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern recognition letters, 27(7):773–780, 2006.
[3] KaewTraKulPong P., Bowden R., An improved adaptive background mixture model for real-time tracking with shadow detection. In VideoBased Surveillance Systems, pages 135–144. Springer, 2002.
[4]Zivkovic Z., Improved adaptive gaussian mixture model for background subtraction. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 2, pages 28–31. IEEE, 2004.

7) Annexes