UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

People counting and runner identification in athletic races

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Felipe Gasel Cunha

In partial fulfillment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisors: Josep Ramon Morros Rubió and Javier Ruiz Hidalgo

Barcelona, July 2019

Title of the thesis: People counting and identification in athletic races

Author: Felipe Gasel Cunha

Advisor: Josep Ramon Morros Rubió and Javier Ruiz Hidalgo

Abstract

The objective of this project is to create software capable of analyzing a video sequence of running competitions. The analysis consists of detecting the runners, tracking them with the intention of knowing their position when they cross the finish line and counting them. Another functionality of the system will be recognizing the bib numbers, thus making it possible for every runner to get their time. The software was developed studying different techniques of object detection, tracking and character recognition to try to choose the best for this specific application. A set of experiments has been performed to validate the proposed system.

Acknowledgments

First, I would like to thank Prof. Ramon Morros and Prof. Javier Ruiz who has supervised my thesis and believed in my capacity to develop this project. I would also like to thank my colleague Berta Nuñez that collaborated in the same project as well the DAPCOM company that supported the project.

Secondly, my family always support my dreams and made all of that being possible, with patience, help, and confidence.

Overall, my experience in Barcelona was amazing, thanks to the amazing people I have met there in these years that I pursued this degree. I would not be here if it was not for a long list of individuals who influenced me, those are anonymous at this moment because the list will be too big to be written here.

Revision history and approval record

Revision	Date	Purpose
0	25/06/2019	Document creation
1	09/07/2019	Document revision

Written by:		Reviewed and approved by:	
Date	12/07/2019	Date	14/07/2019
Name	Felipe Gasel Cunha	Name	Ramon Morros
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation and contributions

Running is known for being one of the oldest sports in human history, and it's in constant evolution. Currently, there is a lot of races and the organizers and runners want to record the time of arrival, position, velocity and many other statistics. Nowadays the cost of signal processing is being reduced and becomes more and more popular to be used in all kind of sports to help to diminish human errors and provide information about the competition.

Many competitions offer the option of buying a chip that registers the personal time from the start to the finish line. This method is pretty cheap and precise but if not all competitors get the chip, the organization cannot assure the complete order of arrival. Another problem with using the chip, is that it is an intrusive method as it must be locked in the shoes.

To solve that problem, the company DAPCOM came with the idea of using a video system to register the competitors that cross the finish line using a simple camera behind the line. The camera will take a shot over the last minutes of every person that manages to finish the race assuring that the number and order of the participants, who have finished the race, is known.

In this work we study how computer vision can assist with that, using an automatic algorithm for detecting and tracking people in the finish line providing timestamps for each competitor. Moreover, a character recognition system will allow identifying each participant by their bib number. The system should work by applying person detection on periodic frames followed by a person tracking step that will take the lead, predicting the position of the objects from the initial position given by the detector.

This work is a project supported by DAPCOM in partnership with the UPC which aims to find a new solution using the newest technologies available in machine learning.

1.2. Objectives

The main objective of this work is to create a software prototype with a computer vision system to count the people crossing the finish line in running races from video sequences. The system will also be evaluated to recognize the identification numbers of each runner and assign them to each participant with a timestamp.

More detailed the project focus on three main aspects:

- Study different people detection and tracking techniques to evaluate their performance in the framework of video sequences of running races.
- Similarly, detection and recognition techniques for the identification numbers will be analysed and applied to the prior system.
- Determine the optimal configuration for the complete system, parameters, set up of cameras for recordings, and if the system is capable of performing in real-time.

1.3. Work Plan

This project has been carried out as a Final Master's Thesis of the "Master of Telecommunication Engineering" at Universitat Politecnica de Catalunya. Myself, Prof. Ramon Morros and Prof. Javier Ruiz, have arranged weekly meetings to revise the work's progress.

1.3.1. Work Division

The work division of the project are defined as follow:

- P1: Project Management
- P2: Research about possible algorithms
- P3: Dataset preparation
- P4: Main Architecture
- P5: Validation

1.3.2. Milestones

The Milestones are listed in table 1.1.

Table 1.1 - Milestones

Work Part	Milestone	Date
P1	Project Definition	15/02/2019
P2	Define detector algorithms	07/03/2019
P2	Define tracker algorithms	10/04/2019
P3	Unirun Record	03/03/2019
P3	Cursa Besos Record	09/06/2019
P3	Cursa Barca Record	16/06/2019
P4	Integrate Detection and Tracking	23/04/2019
P4	Reidentification system	18/05/2019

P4	Counting system	25/05/2019
P5	Test on detector	01/03/2019
P5	Test on tracker	05/04/2019
P5	Test on detection + tracking	10/05/2019
P5	Tests on full design software	30/06/2019

1.3.3. Gantt Diagram

The Gantt Diagram with the work division and the different tasks can be seen in figure 1.1.

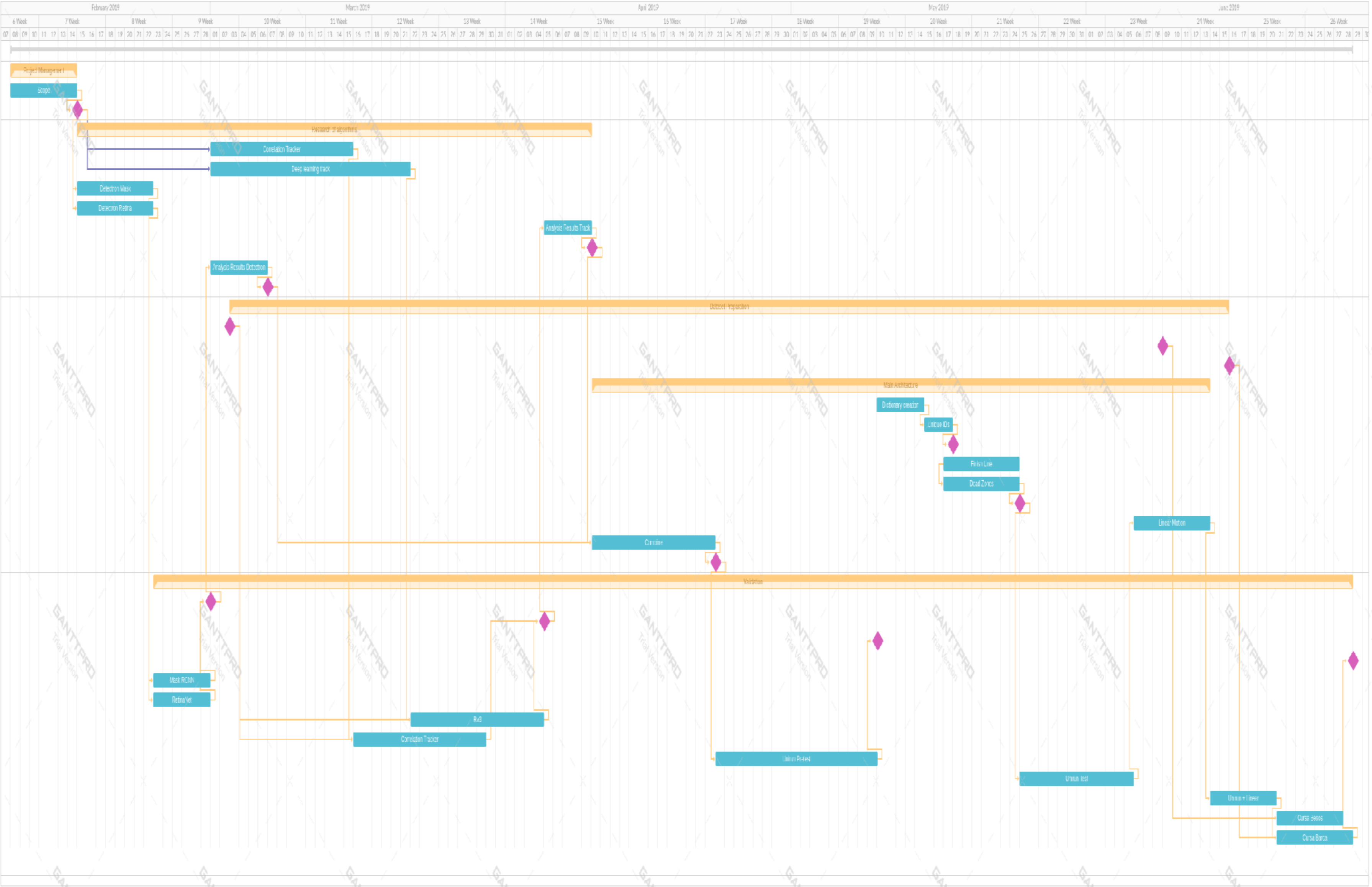


Figure 1.1 – Gantt Chart

2. State of the art

This project includes three sections, object detection, tracking, and character recognition. In each one of them, there is a great variety of techniques to work on, in this chapter, we review some of the techniques.

2.1. Object Detection

The aim of object detection is finding all instances of objects in a cluster of known classes over an image. The detection can be given in the form of a class and/or bounding box that will define the position of the detected object in the image.

Object detection models are typically divided between sliding windows [1] and region proposal classification approaches [2], the second method is the most used in the last years due to a great jump in the accuracy provided by R-CNN [2] later improved by the Faster R-CNN [3]. In the following sections, we are taking a closer look at some of the methods and their approach over the object detection problem.

2.1.1. Faster R-CNN

The R-CNN [2] refers to the convolutional neural networks, focusing on region-based, trying to detect a bounding box of objects. That means generating a rectangle over the identified object, analyzing the original image and generating candidate regions for the objects that will be named as Region of Interest or RoI. These regions will be passed through a convolutional network to have their bounding box extracted to a correspondent object.

Further, the Faster R-CNN [3] method was an improvement that managed to decrease the searching time for the RoI using a Region Proposal Network (RPN). For that, the RPN uses classification of the areas called anchors and proposes the ones with a higher probability of containing objects.

2.1.2. YOLO

This method takes a different direction then the R-CNN and sees the task as a regression problem. Using the features of the whole image to predict each bounding box and with the components separated from the detection are being unified in one neural network, it predicts all bounding boxes through all the classes for an image in a simultaneous way making that an extremely fast method.

The design of YOLO [4] allows training from start to finish in real-time with a high average precision although lags behind the state-of-the-art. The system divides the input image into cells of S x S. If the center of an object is inside one of these cells, that will be responsible to detect the object. Each cell in the grid predicts a bounding box and the confidence score for that cell. They reflect the confidence of the model in which the bounding box contains one object. If it wasn't possible to recognize any object inside the cell the score would be zero.

2.1.3. Mask R-CNN

Mask R-CNN [5] is a method based on the Fast R-CNN [3], which means, it's a deep neural network that serves to segment different objects of a figure in a similar way to the others R-CNN. The major difference is that this method can give us at the same time a bounding box and a mask over the object as the class of it.

The process happens in two different stages. The first is named RPN which analyzes the original image and generates the RoI exactly equal to the Faster R-CNN. The second phase takes each one of these regions and classifies them according to the classes of the objects, generating the masks as we can see in the representation of the process in figure 2.1.

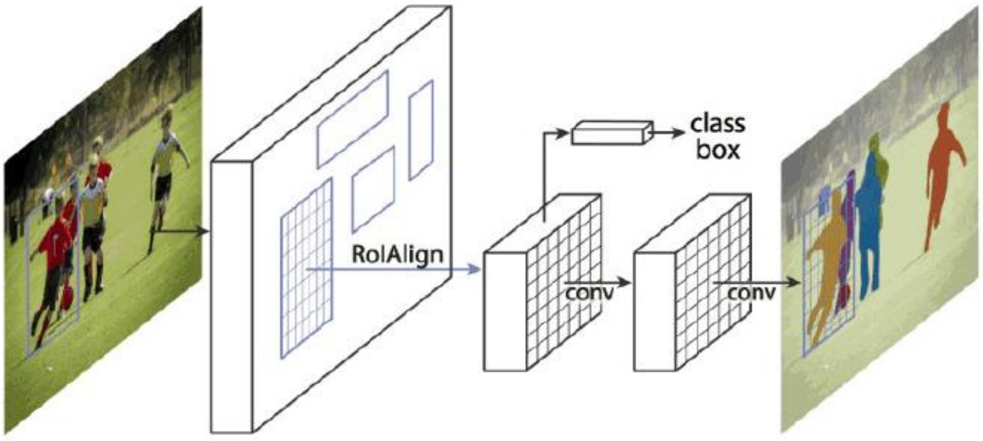


Figure 2.1- Mask R-CNN process representation.

2.1.4. RetinaNet

The RetinaNet [6] is a network which has two sub-networks for specific tasks and a backbone network. While the backbone oversees convolving one array of features from the input image, the two other networks are there to perform the classification of the objects at the output of the trunk network and convolutional regression from the bounding box.

The sub-network of classification is nothing more than an estimation of the probability of the presence of objects in each spatial position for each class of object. There's a matrix of characteristics where usually the input is ResNet 50 or ResNet 101. The sub-network of regression is like the classification network but with non-compatible parameters. The output of this is the place of the object in the image and its respective bounding box.

2.2. Multi-Object Tracking

The Multi-Object Tracking or MOT consists of the act of location all targets of interest in a video track and relating temporally the locations of each object. A track is then the succession of detections of a given object along a video sequence.

2.2.1. Kalman Filter

The Kalman Filter [7] was one of the first methods for tracking objects. Their idea is based on motion estimates, a vector of state that includes the parameters of the object, such the position, and their speed, combining prediction based on a linear dynamic model and a measure over the image. Both prediction and measurement can be affected by noise, that the Kalman filter is considered Gaussian.

2.2.2. Correlation Tracker

The tracker based on filters [8] is responsible for modeling the image of the objects using trained filters with sample images. The object is selected initially based on a small object-focused tracking window in the first frame, from where the tracking and the training work at the same time, while the object is tracked by a correlation filter in a window of search on the next frame. The location corresponding to the maximum value of the output of the correlation corresponds to the new position of the object.

One popular tracker is called MOSSE [9] and works well on objects that only move from one side to the other but fails if the object approaches or moves away from the camera due to the changing of scale. A possible solution for that is a scale pyramid to estimate with precision the scale of an object after the movement, and with that, the tracking of the object can be done even for dislocation of position as well as scale.

2.2.3. Re3

By giving only an initial bounding box we get a serious problem at the generic object tracking and so it represents a big challenge for convolutional neural networks. The major part of the deep-learning algorithms is based on having a million examples to properly work in a way that it can learn the invariant concepts from a high-level perspective. In that way, the object detection learns to differentiate between object, but it cannot distinguish two objects which are in the same category, like one person from two different people.

This tracker works giving one initial example and then specializes in following an object. The adaptation of deep learning to tracking of an object is a difficult task and that’s why they are classified in three different categories. These categories are online training, offline training and hybrid training methods.

The Re3 [10] works as a hybrid training. It works as a regression network in real-time as it says in its full name: Real-Time Recurrent Regression Networks. This tracking system consists of diverse convolutional layers that introduce the form of a given object, recurrent layers which record the form and the movement information, and a regression layer that predicts the location of the object. An example of how it works are showed in figure 2.2.

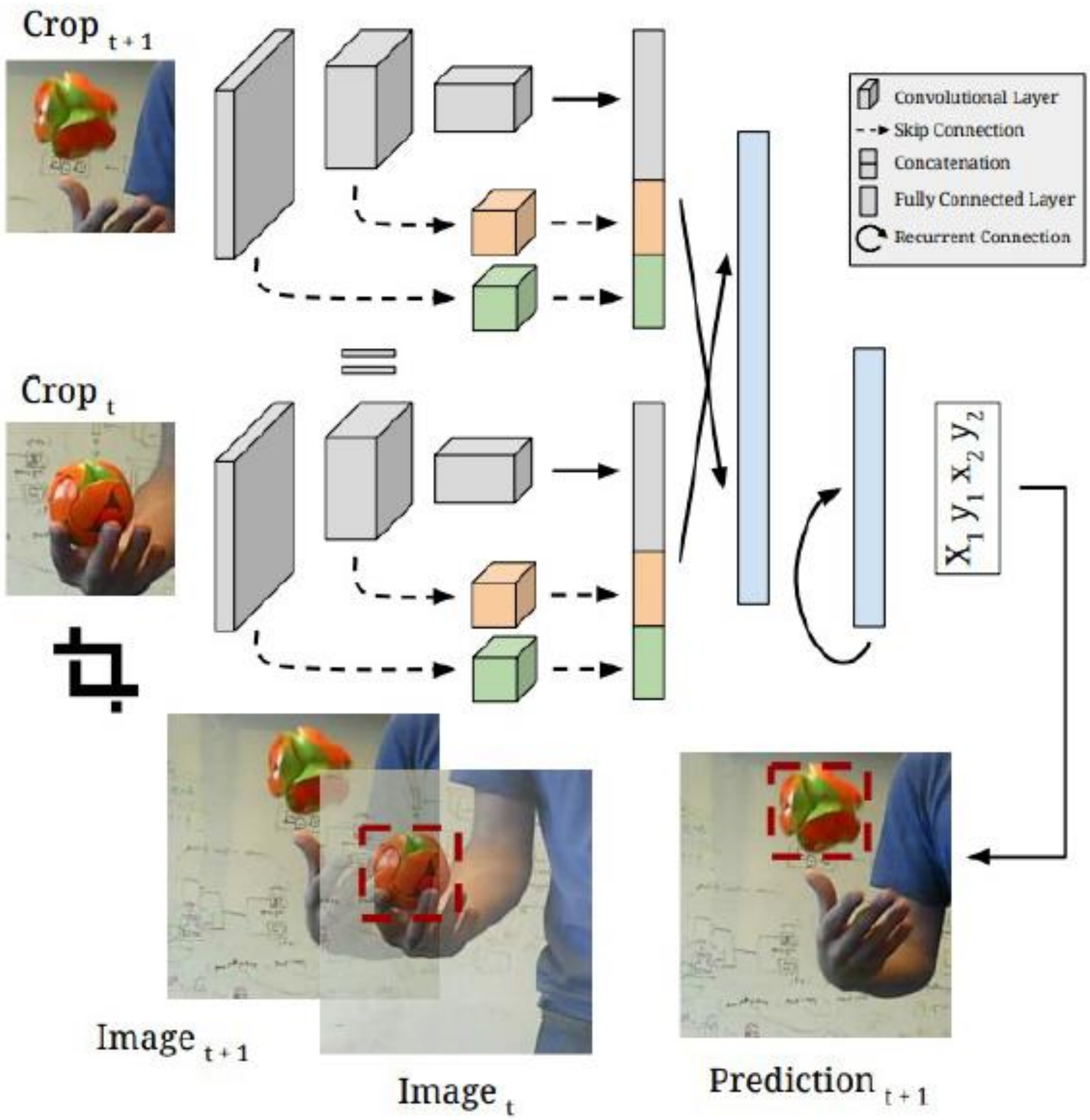


Figure 2.2 – Re3 Process to obtain the predictions of the tracked object

2.3. Character recognition

The optic character recognition is a method that converts images into text. When we visualize an image our brain can recognize the forms and associates that to previously seen forms to identify what we are seeing. However, a computer can only recognize different points in an image and not that in that image exist a text, at least not directly.

The OCR, either adopt Connected Components Analysis (CCA) or Sliding Window based classification [11]. CCA methods extract possible components in various ways, then filtering the non-text components using designed rules or classifiers. Sliding window methods pass a window varying sizes over the image, where each window is classified as text segments or not.

Every line of an image is checked to see whether the different points can represent a letter or a number. It’s important to understand that it is an estimation and the limitations are vast, especially when the quality is affected.

There is a process that uses OCR like Tesseract [12]. That follows a traditional process of step by step with the first of which being an analysis of connected components, memorizing the contours of these components and Kraken [13], which is an implementation of the OCR combined with a neural network, differently from traditional approaches, this technique mimics the way we learn to recognize letters.

Furthermore, one of the most recent methods is the Fast Oriented Text Spotting [14]. This method approaches the problem of detecting and recognizing as a mutual task, while most existing methods treat it as separate tasks. Using a single network that shares computation and visual information among these two tasks, it achieves a more generic feature.

2.3.1. Aruco Code

With movement, diversity of illumination, scales, and distance the OCR can fail as it requires a lot of processing and perfect conditions. A possible alternative for that is the use of Aruco code.

The Aruco code is a binary square marker, as showed in figure 2.3 that allows the registration of the information inside it, using an inner codification [15]. The markers are printed in regular printers having a minimal cost, high robustness, and fast detection.

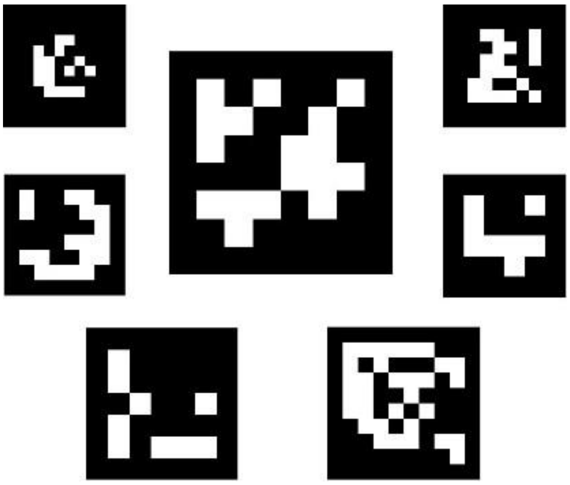


Figure 2.3 – Example of different markers.

3. Methodology

In this chapter the methodology followed by obtaining the experimental results from each part of the full implementation is explained, so is the dataset used, the devices and software used in the validation of the techniques.

3.1. Full Implementation

The complete software prototype methodology is based on an analysis of the video sequence frame by frame to detect, every N frames, and track, in the other frames, each of the people present in the scene. The ratio between detection and track is due to detection systems are much slower compared to track systems. To accelerate the process the track system is used in most of the frames, but as it only relates objects previously detected, a periodic detection is required to capture new people that appear in the video. In the case of occlusions or people disappearing there is a function who tries to re-encounter them after a few frames. The output of this is a bounding box (a set of rectangle coordinates around each person) associated with a Unique ID during the entire time, the person is in that video. After detection/tracking, there is a system of identification of bib number, to get the number inside each bounding box and relate it to a Unique ID.

The following sections describe how the detector and the tracking system work together, and the technique used to count the people and an algorithm section.

3.1.1. General Scheme

The software is composed of five modules. They are as follows: a detection module, a tracking module, and a association module, bib # identification and people counting. The association module, gives a unique ID to new detections and resigns the detected objects to existing tracks when possible.

A high-level block diagram is presented in Figure 3.1. The system is composed of a people detection system, a people tracking system, an association system and an identification module. Every frame starts going through the detection or the tracking system depending on the frame number. At the detection system, new people can be detected and will have a new ID associated with them. The tracking system catches the objects detected and follows their movement through the video frames. Note that every frame, independently if it is detection or a tracking frame, passes through the relation and identification system. The association of objects is designed to relate detections to previously tracked people. An already detected object will be marked in order to avoid double counting when it moves within the finish zone.

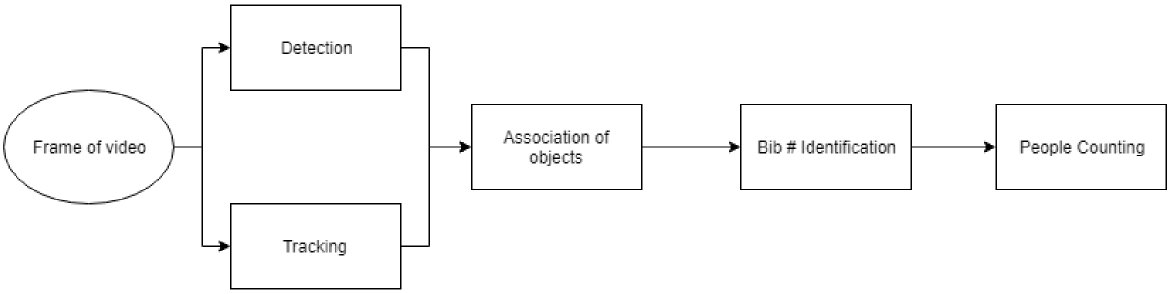


Figure 3.1 – High-level block diagram

3.1.2. Detection and Tracking

The objective of detection and tracking systems is to determine the spatial position of the objects for every frame. In order to do that the detection system chosen is an implementation by the Facebook AI Program Detectron[16]. The only change on the system was to discard any detected objects that were not classified as people. The tracking system is a deep learning network Re3[17] created by Daniel Gordon.

3.1.3. Association of objects to tracks

In every frame, being detection or tracking, every bounding box must be associated with one existing object or verified if it’s new and then associated with a new unique ID. To implement that idea, we calculate the Euclidian distance between the centroids from all existent objects and the objects that appear in the following frame, that the centroid has calculated using a list of bounding boxes that correspond to each object detected and is represented in figure 3.2.

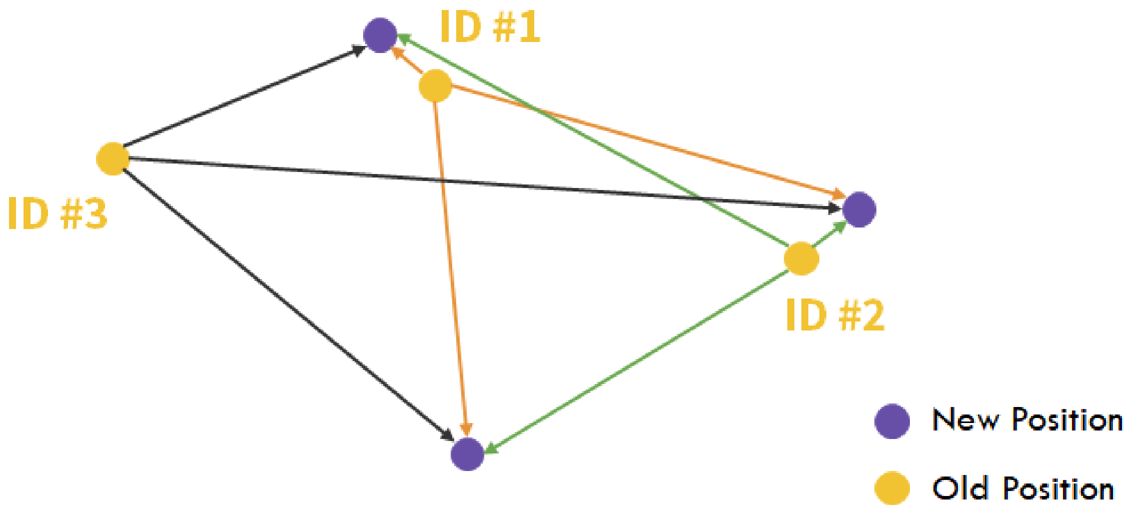


Figure 3.2 – Graphical representation of the Euclidian distance calculation.

This technique is used every time, but to register/deregister a new object that must happen in the frames that will pass through the detector. Moreover before that the algorithm will have a max distance from the old centroid to the new centroid and if one object centroid is inside this area it will be associated with the previous unique ID and not generate a new ID, in the case of more than one centroid is in that area, the closest one will be the assigned as represented in figure 3.3.

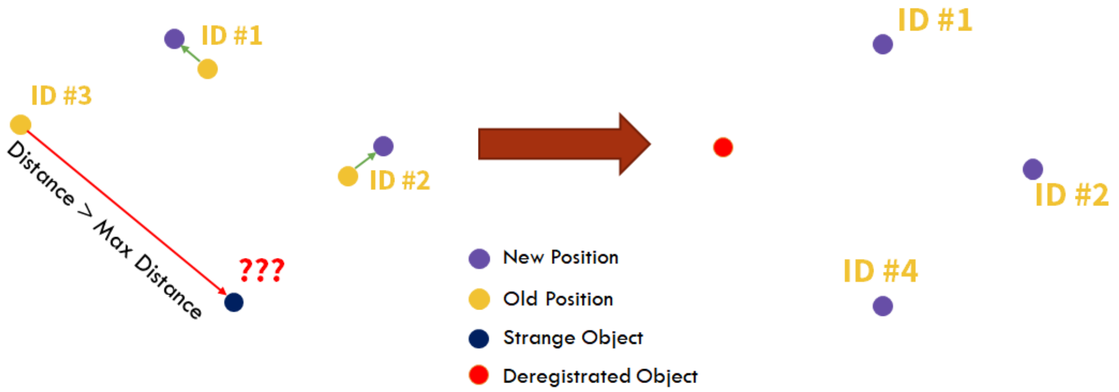


Figure 3.3 – Graphical representation of register/deregister of objects.

A simple flow chart explaining this function is described in figure 3.4

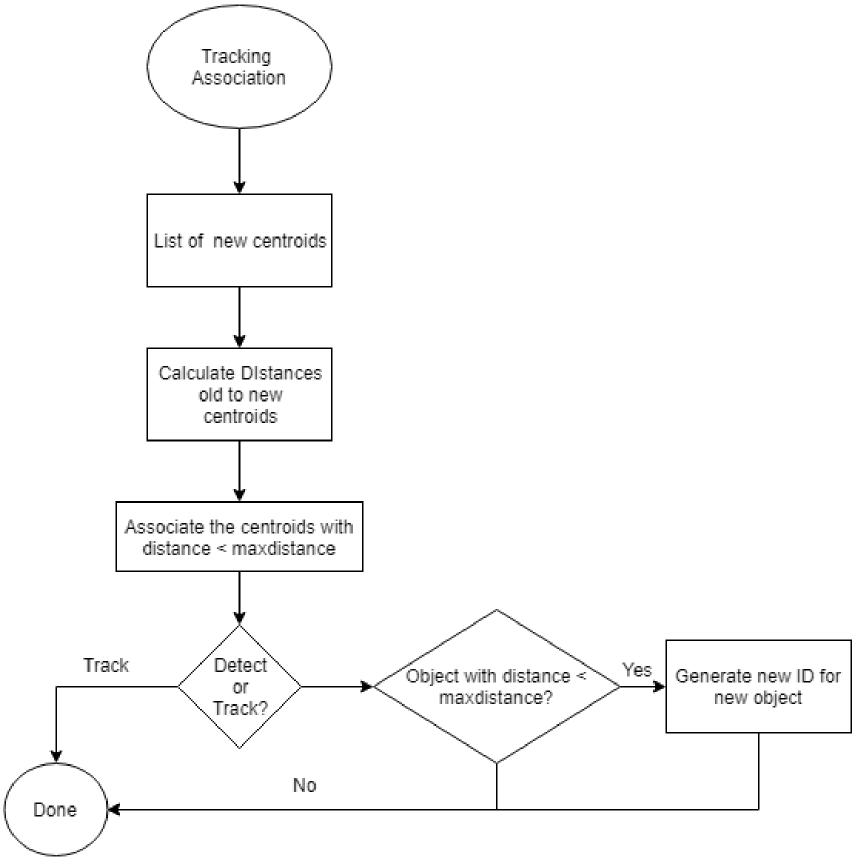

Tracking Association
List of new centroids
Calculate DIstances old to new centroids
Associate the centroids with distance < maxdistance
Detect or Track?
Track
Object with distance < maxdistance?
Yes
Generate new ID for new object
No
Done

Figure 3.4 – flow chart of association track

3.1.4. Counting

For people counting, we must count the people that cross the finish line and exclude the ones that are in the field of view and are not runners. A three polygon was designed to be placed using the first video frame (figure 3.5) in order to perform people counting. It consists of one polygon for the finish zone (green, figure 3.5) which will detect objects crossing the finish line, two polygons (red, figure 3.5) which is responsible for the dead zones, zones outside of the race track, and they should not count any people.

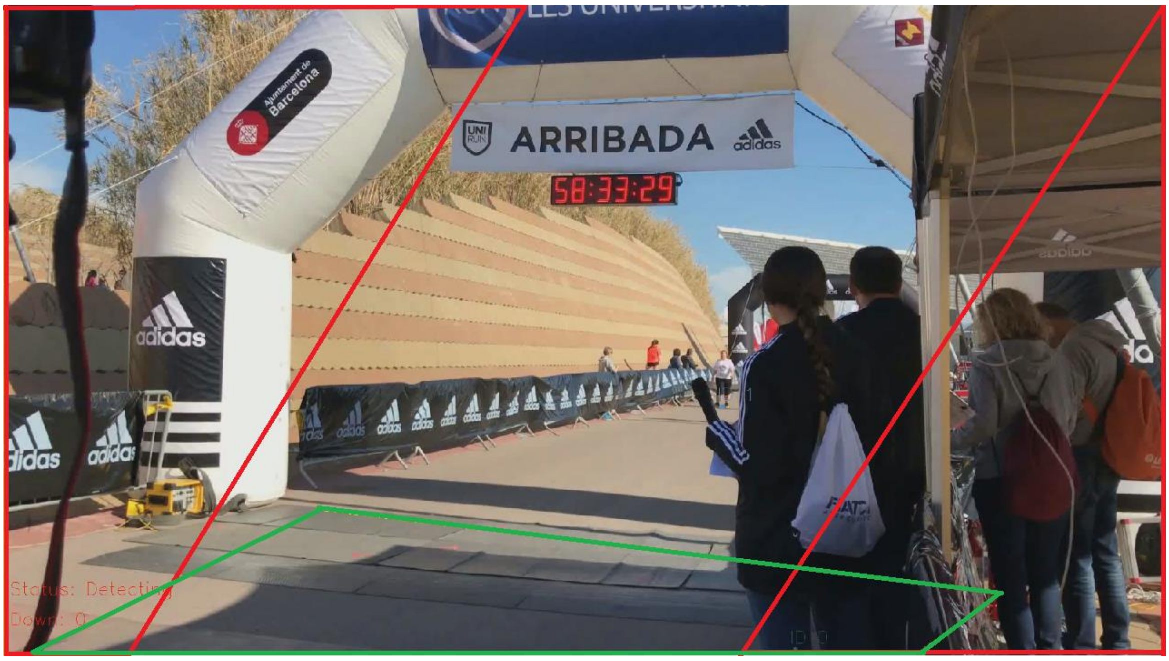


Figure 3.5 – example of the design of zones on the first frame

The exact way of how these zones work with the program is explained in the following flowchart (figure 3.6).

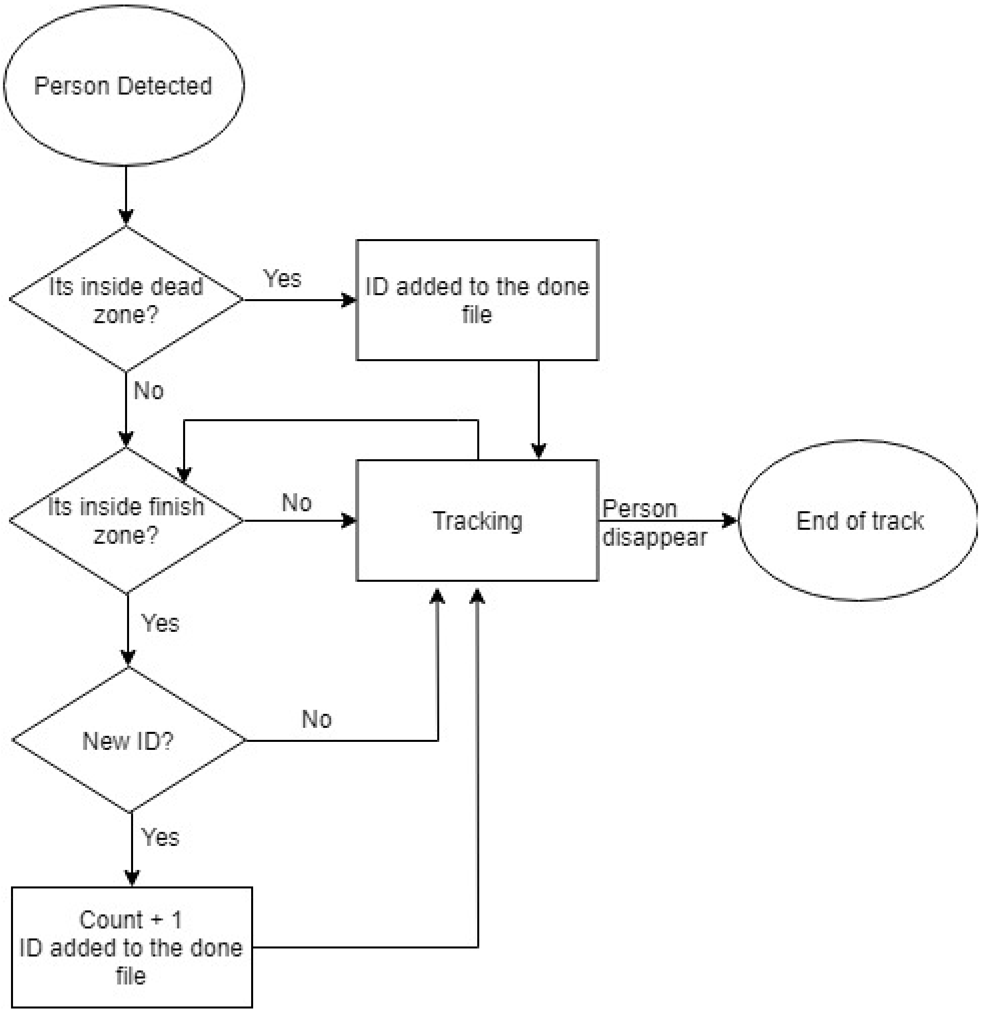


Figure 3.6 – Flowchart of the counting technique

3.1.5. Bib number recognition

For recognition of bib number, the objective is to determine the numbers of each runner. Using the OCR Tesseract in every bounding box from every frame the text would be extracted and associated with the object. If there are different numbers being recognized to the same object in different frames, the number that appears most will be the one assigned. Another possibility for the bib number recognition is to add and use Aruco codes with the number in their codification, making this process faster and more reliable. Both systems can work together.

3.1.6. Algorithm

The full algorithm of the software consists of the combination of the detection, tracking and counting system. At each time instant, the system maintains a list of tracked people, with an associated ID that is maintained along with the video. For every new frame the position of the tracked people is updated to a new position in the image, which occurs only when the people are already being tracked. For new people the detector part will take a step in every N frames to try to identify new people to track. When the tracking module cannot predict the next location of the object, a new marker (“disappear”) is applied to this object. If the object contains this marker, a linear motion model tries to predict the position using the velocity of the object. A simplified algorithm of the software is described below.

```
counted_objects = 0
trackedObjects = {} # Dictionary of tracked persons, key = ID

for each frame:
     rectangles = []
     if t%N == 0:
           rectangles = detect(frame)
     else
           rectangles = track (trackedObjects, frame)

     trackedObjects = associate (rectangles, trackedObjects)

     if disappear != 0 and disappear < maxdissapear
           rectangles = linearmotion (trackedObjects,frame)
           disappear++
     elsif disappear > maxdissapear
           delete object

     for object in trackedObject:
           if object->centroid > finish_line and object->counted =
False:
                 counted_objects = counted_objects + 1
                 object->counted = True
```

4. Results

This chapter presents and analyses the results of the different parts of the complete program as well as the full implementation, in terms of accuracy and timing; It also discusses all challenges found during the whole process.

4.1. Dataset acquisition

The dataset used in this master thesis was created by making video records from 3 different runs in Barcelona using a mobile phone camera, during the period of the thesis. It’s composed of videos with a resolution of 3840 x 2160 pixels. A total number of 18 videos in three different height (1.5m, 2m, and 2.5m) with a sum of over 800 seconds of useful video with different density of persons, different illuminations and angles.

Our first session took place on the Unirun on the 3rd of March in Parc del Forum, followed by Cursa Besos on the 9th of June in CEM Maresme and the last session recorded at

Camp Nou on the 16th of June in the Cursa Barça. These three mentioned locations are marked in figure 4.1.

In the first session, we recorded clips at 4K and with lower quality, to determine the quality necessary for the system. After analysing the data obtained it was decided to record in 4K because of the bib number recognition. At this session, it was being possible to obtain 183 seconds in 3 different clips.

During the second session, we got videos recorded from two parallel positions close to the finish line trying to obtain fewer occlusions than before. Using a selfie-stick the records were done at a height of 250cm. As a result, we achieved a total of 278 seconds divided into 8 different clips. This session was a familiar run with a small number of people and almost zero occlusions.

Furthermore, during the third session, we acquired 7 clips with 309 seconds in total. That day we couldn't use the selfie-stick for the height, which results in only 200cm this time. Because of a platform in the middle finish line, these videos have a lot of occlusions.

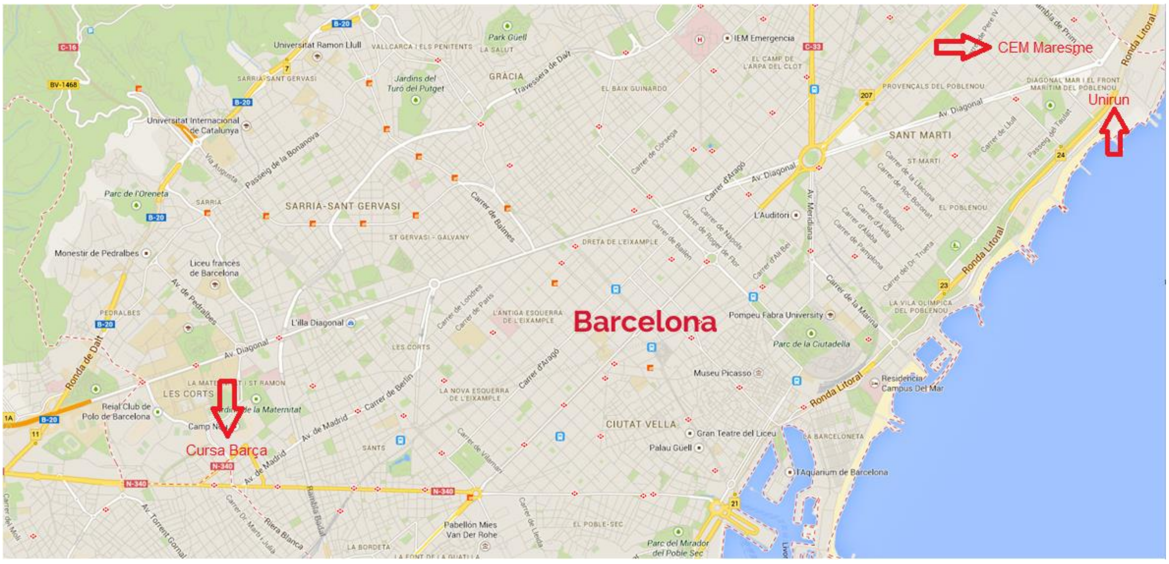


Figure 4.1 – Location of the runs

After each session, a manual person counting was done to obtain a ground truth and posterior analysis of the results of the project.

4.2. Detection model analysis and decision

The test was proposed to try different networks and check if the RetinaNet would be better in our images than the Mask R-CNN the results are presented at table 4.1. To perform this test we run the Detectron program on the UPC Imatge group cloud server using a GPU and 6GB of ram, over 10 frames of our dataset using one network of Mask R-CNN (model id: 35861858) and 4 networks of RetinaNet (RetinaNet R-50-FPN model id: 36768677, RetinaNet R-101-FPN model id: 36768907, RetinaNet X-101-64x4d-FPN model id: 36768907, RetinaNet X-101-32x8d-FPN model id: 36769641) that can be obtained at [3]. All of them were trained, by the Facebook research group on the COCO Database. These images were randomly selected from the videos recorded in the first session of our database.

Table 4.1 – Number of people separated in runners and spectators in all the frames analyzed

	Total	Runners	Spectators
Unirun1	24	15	9
Unirun2	20	12	8
Unirun3	13	6	7
Unirun4	16	12	4
Unirun5	16	8	8
Unirun6	21	16	5
Unirun7	22	13	9
Unirun8	22	13	9
Unirun9	21	11	10
Unirun10	20	15	5
TOTAL	195	121	74

Using the five different models we generated table 4.2 which provide an analyzed result from each one of the models.

Table 4.2 – Analysed results obtained with different models

	Mask	Retina 50	Retina 101	RetinaX 64	RetinaX 32
TP	141	25	20	24	35
TN	54	170	175	171	160
FP	0	0	0	0	0

While TP is the number of people that were properly identified as people, TN is the number of people that conversely not identified and FP represents a wrong identification of people.

From table 4.2, we can calculate the accuracy of the detection system. The accuracy is a statistical measure that quantifies true value. These values can be checked in table 4.3 for each one of the tested models.

Table 4.3 – Model Characteristics.

	Accuracy
Mask	0.723
Retina50	0.128
Retina101	0.102
Retinax64	0.123
Retinax32	0.179

As we can see the Mask Model shows a much better accuracy when compared with any other model. Comparing the brute numbers of each model we can see that the best model of RetinaNet detects in total 35 people, including runners and spectators, while the Mask model detects 141 persons in a pool of 195 persons. An image example of the difference between the best Retina Model and the Mask model is shown in figure 4.2

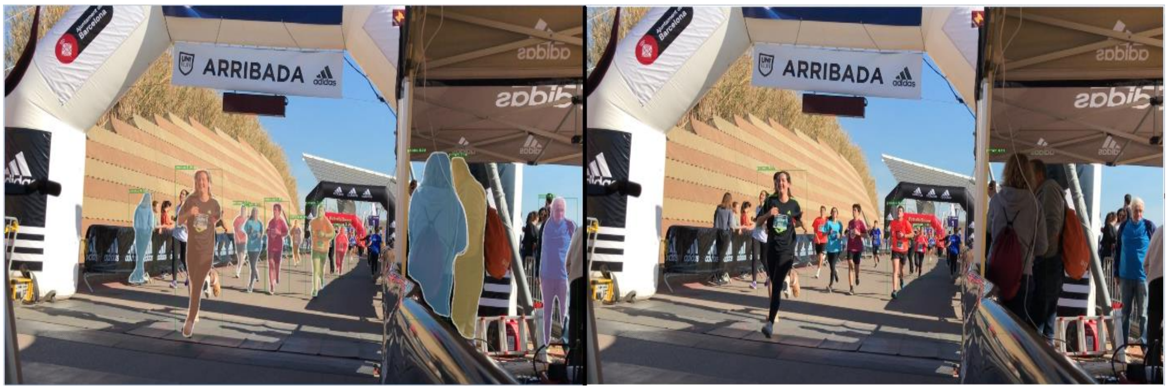


Figure 4.2 – Example of detection using Mask model (right) against RetinaNet model (left) at the same frame

We don’t have any distinction between a spectator and a runner for the detecting part of the project. The Mask model is the best possible solution for it and will be used in the full implementation.

4.3. Tracking analysis and decision

This test was supposed to be performed in the whole dataset. After preliminary tests over only two videos, in all their qualities, we verified a difference of over 6 times more computational timing to process the tracking using the correlation tracker against the Re3 tracker. Due to that, we decided to abandon the correlation tracker and perform the tests and the full implementation using the Re3 tracker.

At the Re3 tracker test, we decided to downscale the videos from the original resolution (4K) by 2x, 4x,5x,6x checking how that affects the time and the accuracy of the tracker. These tests were running at the same platform used for the people detection tests with an integration of the tracker’s technique with the Detectron to perform automatic detection of the people.

The Re3 tracker results from the videos are shown in table 4.4. Two other tables are presented for timing (Table 4.5) and preliminary people counting (Table 4.6). These tables were generated by analyzing videos one by one after processing and checking the misses and false.

Table 4.4 – Videos used for the validation of Re3

Recording ID	Session	People crossing the line	Duration	Height of the camera
5972	1	69	62 s	150cm
5977	1	16	31s	150cm
5978	1	57	90s	150cm

Table 4.5 – Timing using Re3 tracker

Timing on all videos (183 s)					
Timing (s)	Original Resolution	x2	x4	x5	x6
Detection	1424,74	300,33	98,11	84,22	68,73
Track	3214,16	900,31	342,96	281,83	234,48
Program	1281,98	370,54	128,79	99,31	80,56
Movie	4778,22	1199,6	309,99	200,06	134,95
Total	10699,1	2770,78	879,85	665,42	518,72

The Program is the time elapsed outside the track and the detection system, that means the time from all the other functions as Association of IDs, and people counting. While the Movie time is the time elapsed to generate an output video for visualization from the whole set of frames obtained during the program run.

Table 4.6 – People counting using Re3 tracker

People Counting over all videos (142 people)					
	Original Resolution	x2	x4	x5	x6
Total	115	114	114	114	117
Correct	112	112	112	114	116
Misses	30	30	30	28	26
False	3	2	2	0	1

Using the Re3 as a tracker we achieve a failure rate of 21%, this number is a result of the sum of misses and false positives. The misses are in the majority due to an occlusion of the people during the entry in the finish zone, while the false positives are originated by people who are not runners but are in the field of view, including commentators, or people that randomly enter in the race zone.

4.4. Full implementation

The tests over full implementation are realized due to the new functions integrated from correlation tracker on Re3, the addition of bib number recognition. The results are obtained from a partial dataset presented in table 4.7. The results are done only in a part of the total dataset since after checking the videos some of them had a huge number of spectators in front of the camera and in the trajectory to the finish line. As in the tracker analysis, a table of timing (Table 4.8) and also the one for people counting (Table 4.9) were created for analysis.

Table 4.7 – Dataset of videos used for validation of the program

Recording ID	Session	People crossing the line	Duration	Height of the camera
5972	1	69	62 s	150cm
5977	1	16	31s	150cm
5978	1	57	90s	150cm
6491	2	9	30s	250cm
6492	2	4	30s	250cm
6493	2	6	30s	250cm
6494	2	11	30s	250cm

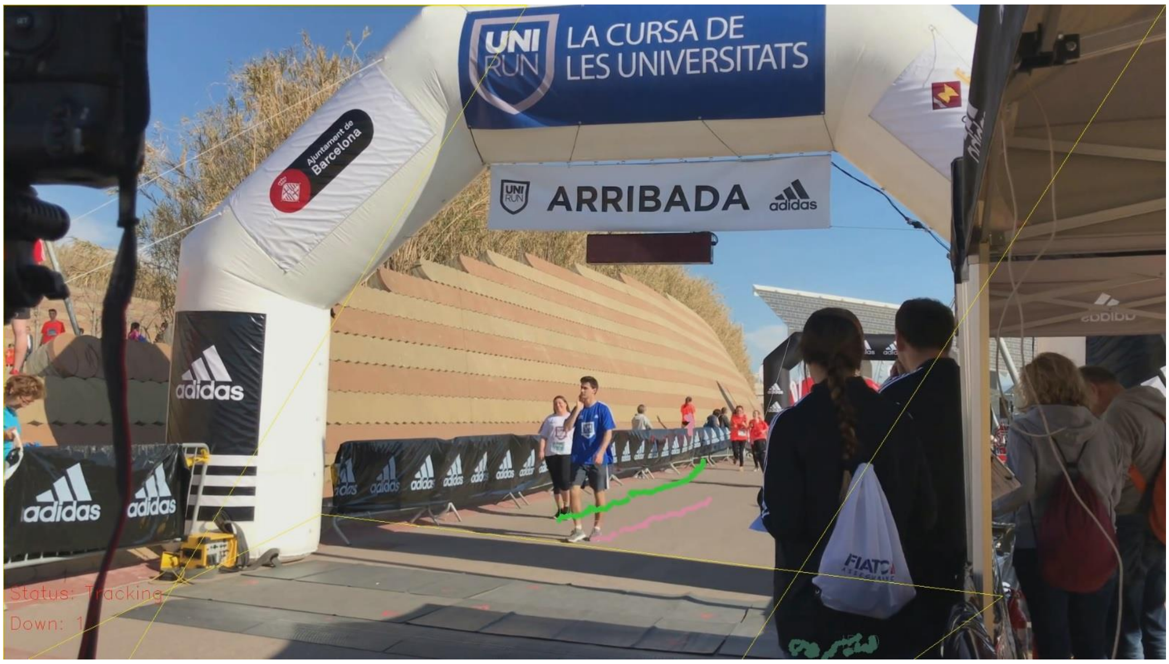
Table 4.8 – Timing using the full implementation

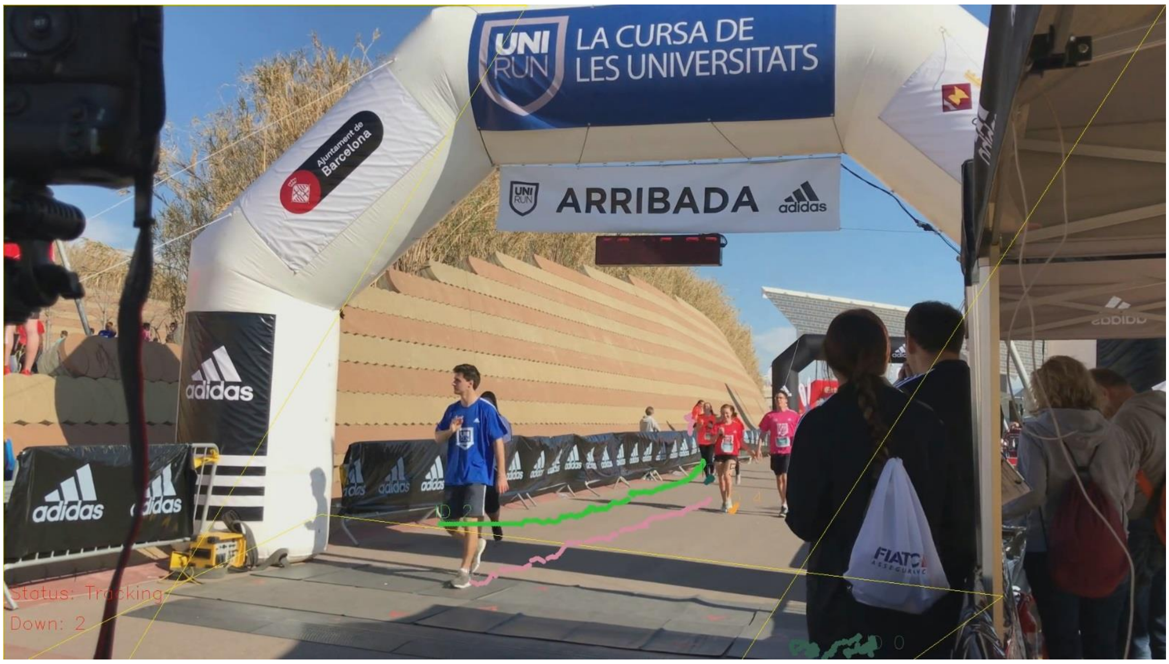
Timing on all videos (303 s)					
	FR	x2	x4	x5	x6
Detection	2594.29	308.26	165.58	146.98	137.03
Tracking	3303.57	956.99	401.24	322.42	277.47
Program	8308.17	2252.94	635.48	375.22	265.14
Total	14206.03	3518.19	1202.3	844.62	679.64

Table 4.9 – People counting using the full implementation

People Counting over all videos (172 people)						
	Ground Truth	FR	x2	x4	x5	x6
Total	172	171	165	159	160	158
Correct	172	161	155	152	153	151
Misses	0	11	17	20	19	21
False	0	10	10	7	7	7

With the full implementation, using the Re3 as a tracker mixed with the correlation tracker functions we managed to have great improvement in both aspects, timing, and counting. We worked with around 60% more time on the videos to be processed than before and got results equivalent to 75~80% faster for all resolutions. The people counting, with all functions implemented lead to diminish the misses in a great amount and the number of false positive didn't raise so much. An example of perfect tracking is showed in the below sequence of frames in figure 4.3.


LA CURSA DE LES UNIVERSITATS
ARRIBADA
Status: Tracking
Down: 1


LA CURSA DE LES UNIVERSITATS
ARRIBADA
Status: Tracking
Down: 2

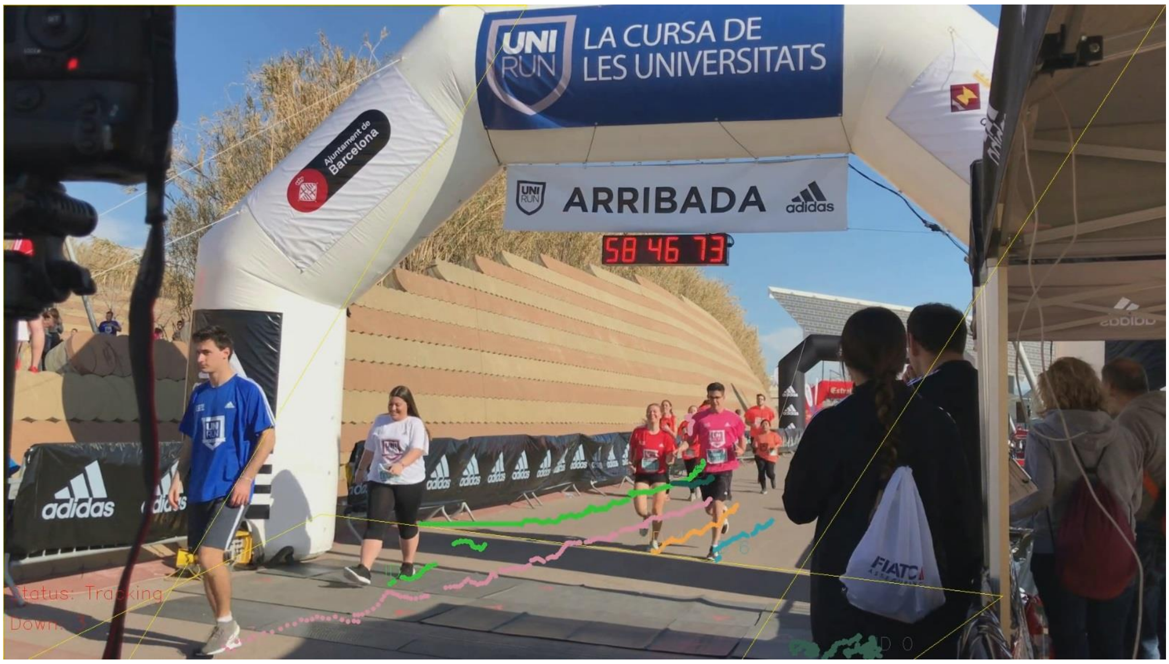
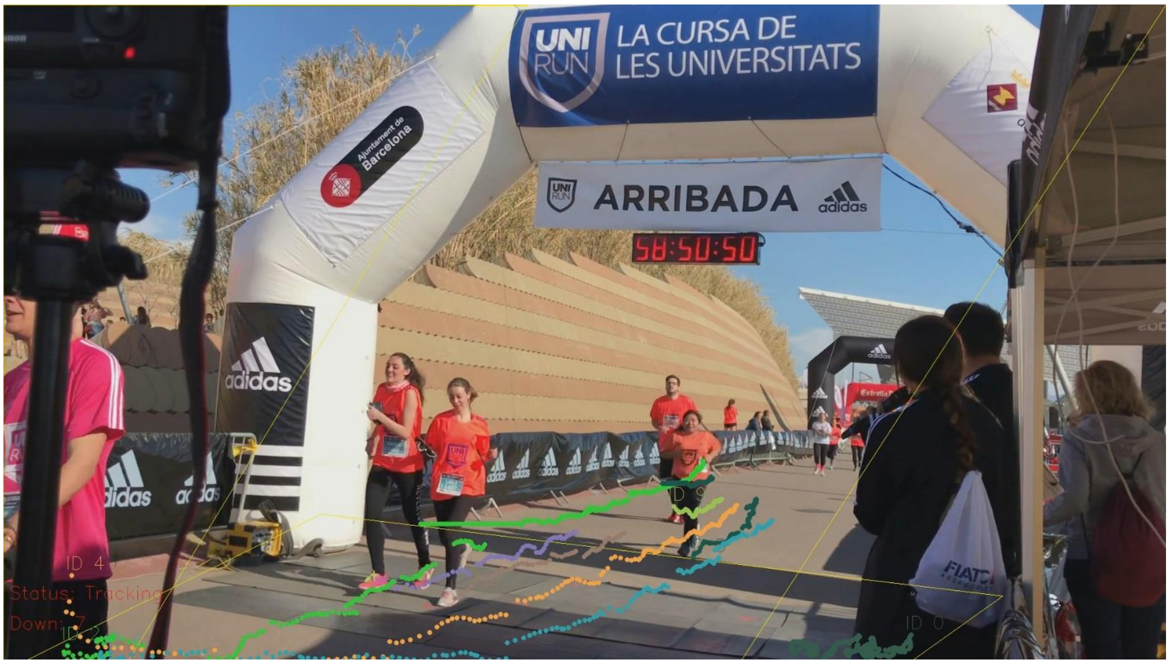
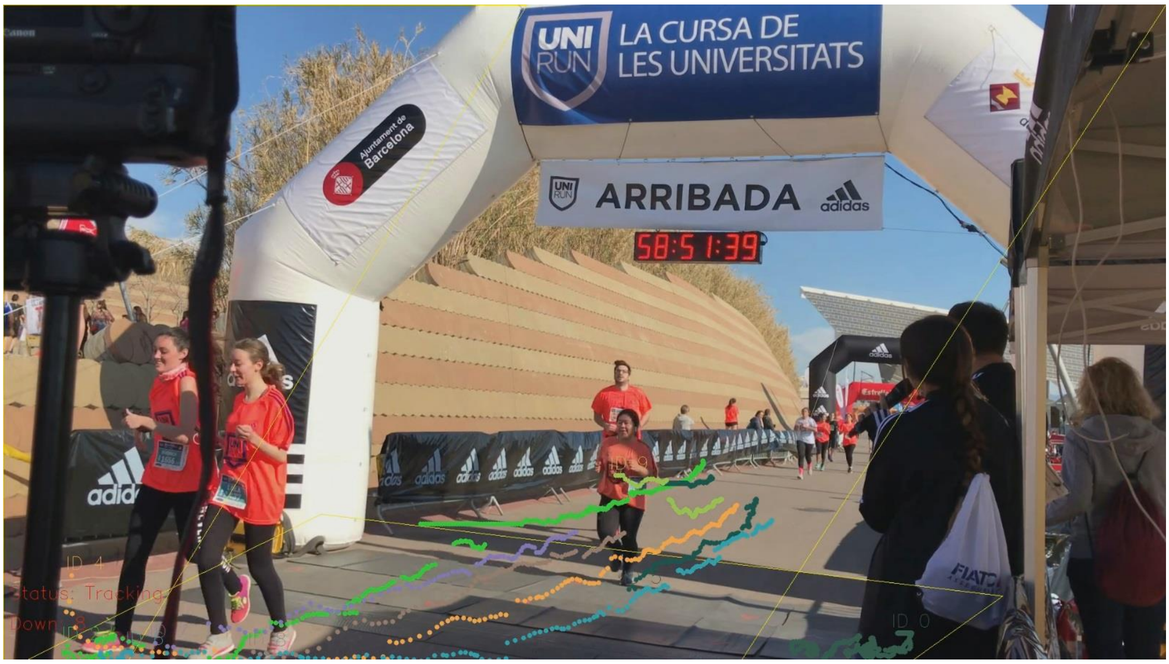


Figure 4.3 – Sequence of frames of good tracking and counting with re-identification

At figure 4.3 we can see 3 frames of the video after the processing showing the paths of the person ID 3 (green) and ID 4 (pink). In the first figure (upper) both of them are outside of the finish zone and the counter shows the quantity of 1, the moment after the ID 4 crosses the line at the second figure (central) and the track goes to 2 and at least with the occlusion before of the ID 3 what possible to re-detect. Separated tables for each video can be found at Annex II.

The people misses are in the majority of people that are occluded near the finish line. The false positives are originated by persons who are not runners but still in the running area. These errors can be checked in the sequence of frames in figure 4.4 and 4.5. The errors can be reduced with a better position for the camera, which means centered recordings.


LA CURSA DE LES UNIVERSITATS
ARRIBADA
58:50:50
ID 4
Status: Tracking
Down: 7
ID 2
ID 9
ID 0


LA CURSA DE LES UNIVERSITATS
ARRIBADA
58:51:39
ID 4
Status: Tracking
Down: 8
ID 2
ID 0

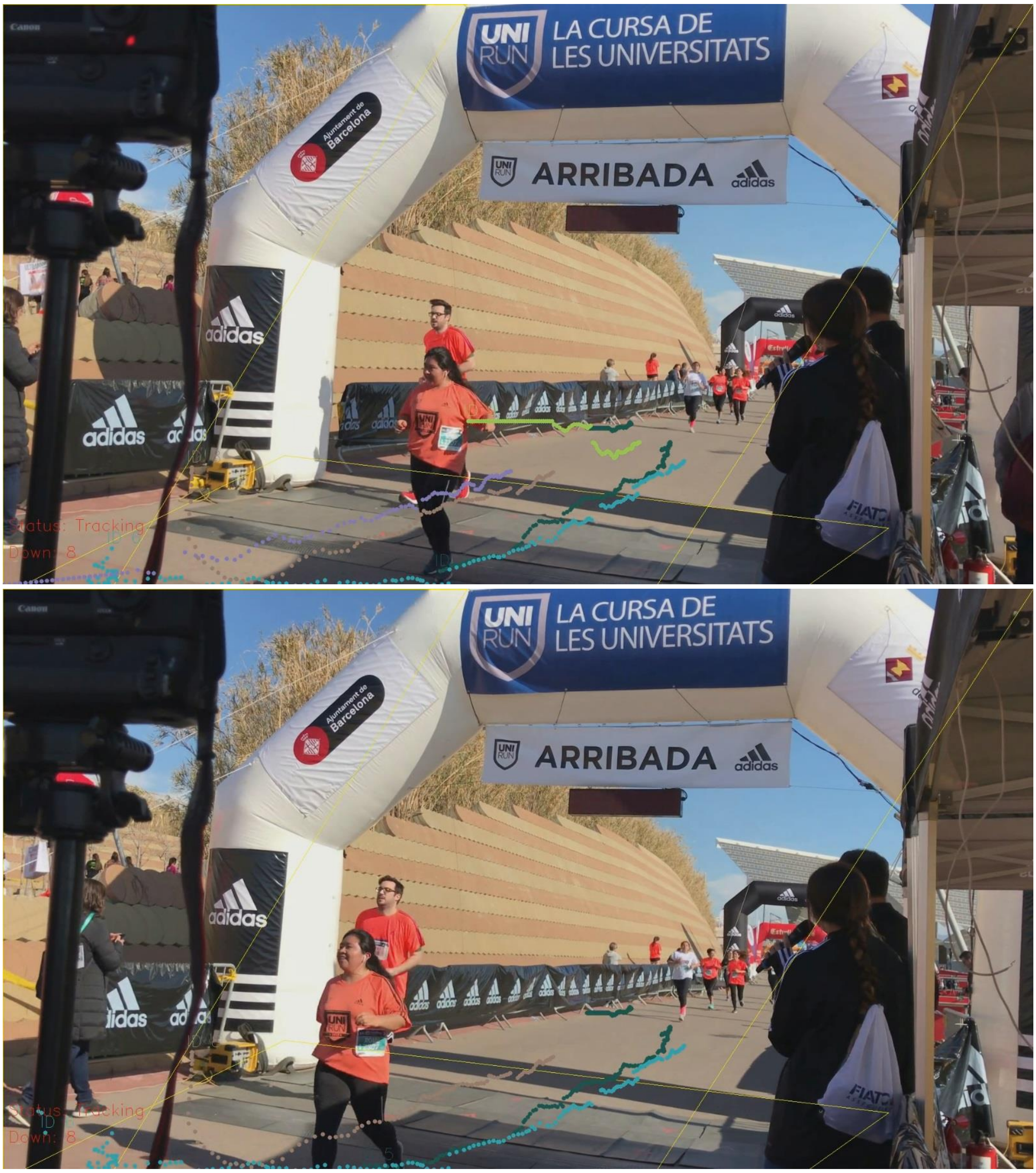


Figure 4.4 – Sequence of frames for an occlusion miss

Looking at the man in orange identified as ID 9, a partial occlusion happens between frame 1 and frame 2 (first two frames) there is poor detection/tracking. Resulting in the sequence tracked on frame 3 that will lead to him not be reassociated after the occlusion.

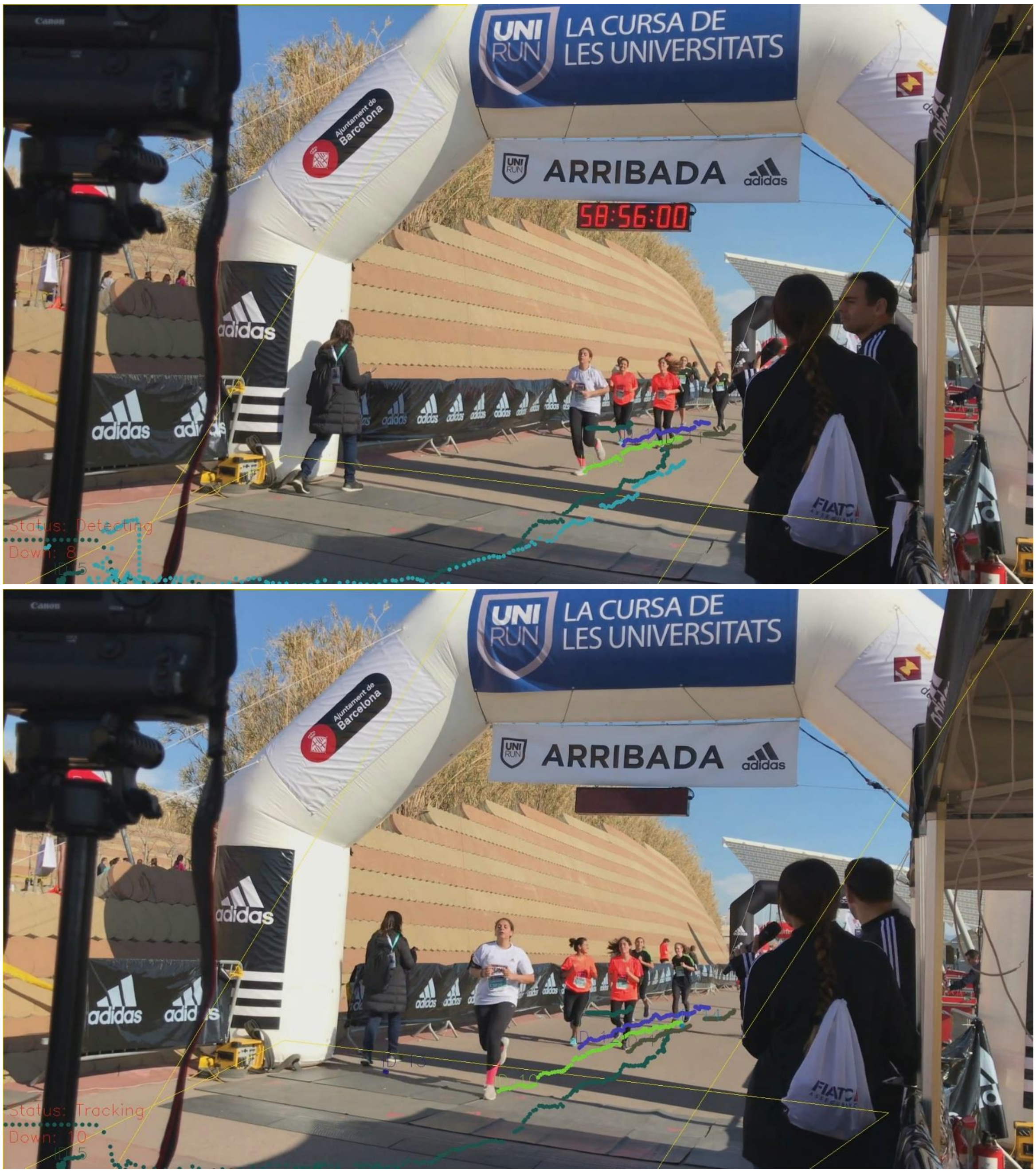


Figure 4.5 – Sequence of frames for a false positive.

The other type of error presented in figure 4.5 is a false positive. A person that isn't a runner enters in the tracking area and is identified as a person receiving a unique ID, in this case, 13 in dark blue, and being counted.

4.5. Number Recognition analysis

For that, a manual segmentation over the dorsal of different people in different videos was made. In case of the manual segmentation had good results, we would apply some automatic segmentation over the bounding boxes using text detectors on the bounding boxes areas. The number recognition analysis was done over 57 images of 14 different people. There are different possibilities of pre-processing that we could use, 4 different processes were designed, always relying on the previous process results.

The first processing was getting the original image pass by a to resize interpolation, change from RGB to Grayscale, binarize it, invert the color and then closing to eliminate the interferences. The second process adds a dilatation before the closing to make the lines a bit thinner, and after the closing, an erosion is done to redefine the lines. The third process is equal to the second changing only the order of resizing with the change of color to grayscale. The last process tries to fix the difference of illumination doing an equalization over the grayscale. This part was implemented by another member of our team but is included here for completeness. The complete details are graphically detailed in Annex I. We obtained very disappointing results with all the pre-processing, as showed in table 4.10

Table 4.10 – Character Recognition results

	Process 1	Process 2	Process 3	Process 4
Correct Detection	2	9	8	2
Incorrect Detection	0	3	6	3
No Detection	55	45	43	52

From table 4.10, it's possible to see that process 2 has the best results, 15% accuracy. Looking at the different results obtained we can say that it will not be effective and for the moment won't be integrated into the full system. Figure 4.6 below shows one example of the dorsal segmentation and process over a correct detection of number.

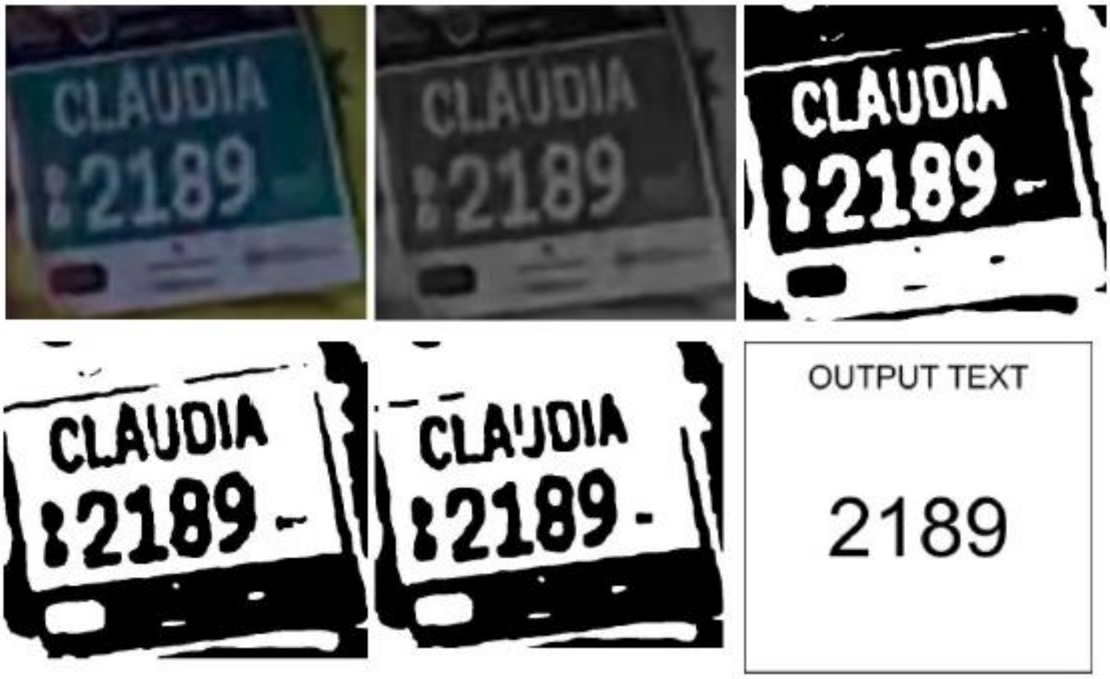


Figure 4.6 – Dorsal with a correct number read by OCR

5. Budget

The costs associated with this project are mainly personnel costs of one junior engineer working as a full-time worker for the master thesis, one junior engineer working as a full-time worker for the graduation thesis, and two senior engineers to supervise the work.

Table 5.1 – Budge table

	Wage	Hours/Week	Total Weeks	Total (€)
Junior engineer x2	8€/h	25	21	8.400
Senior engineer x2	60€/h	2	21	5.040

Finally, the personal cost goes around 13.440 €, but the resources used from the group of Imatge from UPC are not included but can be assumed that rent a similar server on Amazon Web Services would cost 0.33€ per hour estimating the usage of that for approximately 100 days we will add to the sum 792€ as well as a coworking place in Barcelona for 8 hours a day will add another sum of 420€.

By that, the total cost of the project would be 14.652€.

6. Conclusions and future development

The main goal of this study was to design and implement a computer vision system capable of counting and identifying the participants in any kind of running competition. The proposed system can detect, track, and count the competitors when they cross the finish line without using any special device. Furthermore, it is able to extract the bib number from each runner so the order of the arriving can be known. This thesis has focused on the study of different methods related to each part of the complete system. The tests that were done and showed in the chapter of results, divided the whole system into four main sections, detection, tracking, counting, and character recognition.

At the detection part, we managed to test two different networks, Mask R-CNN, and RetinaNet, obtaining much better results using the Mask R-CNN. The Mask R-CNN was the model implemented at the final system.

Furthermore, two different methods of tracking were being tested, a correlation tracker and a deep learning algorithm. In this case, the deep learning method was presenting much better timing results than the correlation tracker, overpassing it on a scale of 1:10 as it works with GPU. For the final system, the deep learning method was chosen thinking of having a real-time system.

For the counting, the test was made with both tracking methods. It works better with the functions predefined on the correlation tracker. Integration between the Re3 tracker and the functions available in the correlation tracker was done to obtain results with only 6~7% of errors caused by occlusions or when people get amounted in the finish zone.

Finally, character recognition didn’t give the results that we expect. Tested in the best situation possible, taking the segmentation by hand in good frames without any occlusion of the dorsal so this part isn’t implemented in the final design for now.

For possible next steps for the system, first is necessary to find a better record spot, probably a centralized one in a higher position to diminish the occlusions and by consequence lead to fewer errors. Another problem is character recognition, for that, must be develop an automatic detection only over the numbers of the dorsal to exclude manual segmentation. The recognition can be done using other methods as adding an Aruco code [19] that will have in its content the number of the competitor as it’s a well-known system with pretty solid results that can raise the correct results.

Bibliography

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D.Ramanan, “Object detection with discriminatively trained part-based models” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[2] R. Girshick, J. Donahue, T. Darrell, and J.Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation” *Computer Vision and Pattern Recognition*, 2014.

[3] S. Ren, K. He, R. Girshick and J. Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks.”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You Only Look Once: Unified, Real-Time Object Detection.”. IEEE Conference on Computer Vision and Pattern Recognition, 2016

[5] K. He, G. Gkioxari, P. Dollar, and R. Girshick. “Mask R-CNN” IEEE International Conference on Computer Vision, 2017.

[6] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. “Focal Loss for Dense Object Detection” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[7] Kenshi Saho, "Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design," Book Chapter 12 in Kalman Filters - Theory for Advanced Applications, 2018.

[8] M. Danelljan, G, Hager, F. Khan, and M. Felsberg. “Accurate Scale Estimation for Robust Visual Tracking.” British Machine Vision Conference, 2014.

[9] “Mosse Tracker”. [Online]. Available: https://docs.opencv.org/master/d0/d02/classcv_1_1TrackerMOSSE.html [Accessed: 30 June 2019]

[10] Bolme et al. “*Visual Object Tracking using Adaptative Correlation Filters”*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[11] Shangbang Long, Xin He, and Cong Yao “Scene Text Detection and Recognition: The Deep Learning Era”. ArXiv, 2018.

[12] “Tesseract open source OCR engine”. [Online] Available: https://github.com/tesseract-ocr/tesseract

[13]“OCR engine for all languages”. [Online] Available: https://github.com/mittagessen/kraken

[14]Xuebo Liu et al. 2018, “FOTS: Fast Oriented Text Spotting with Unified Network”. ArXiv, 2018.

[15] Francisco J. Romero-Ramirez, Rafael Munoz-Salinas, Rafael Medina-Carnicer. “Speeded up detection of squared fiducial markers”. Image and Vision Computing, vol 76 pages 38-47, 2018.

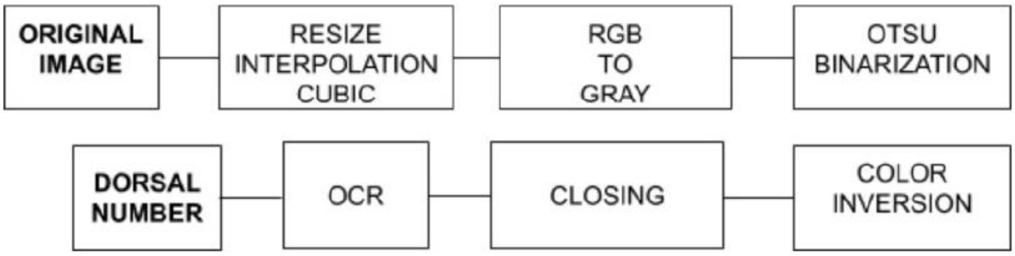
[16] R. Girshick, I. Radosavovic, G. Gkioxari, P. Doll, K. He, “Detectron”. Available: https://github.com/facebookresearch/Detectron, 2018.

[17] Daniel Gordon. “Re3: Real-Time Recurrent Regression Networks for visual Tracking of Generic Objects”, IEEE Robotics and Automation Letters, vol. 3, no. 2, pp. 788-795, 2018.

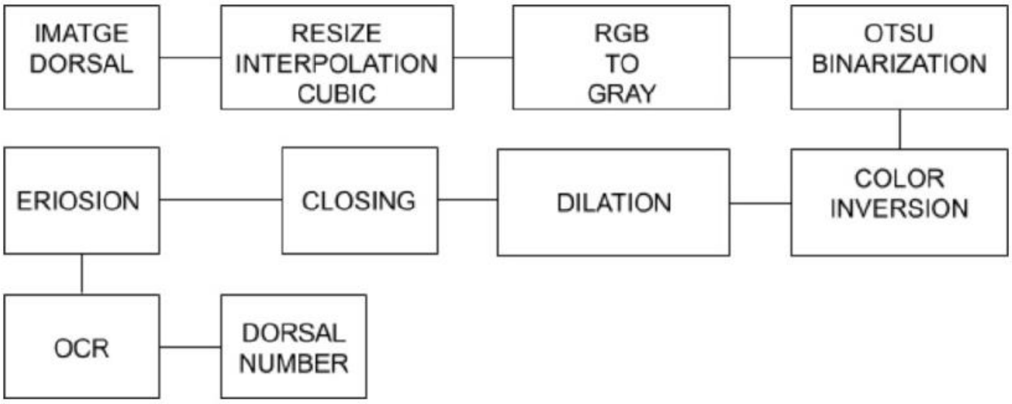
Annex I

The following diagrams show every pre-process tried for the Dorsal Recognition Function.

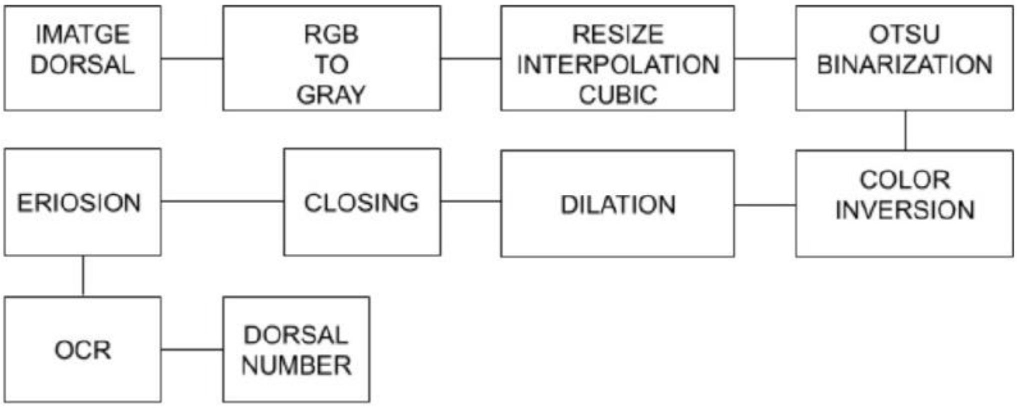
Process 1:



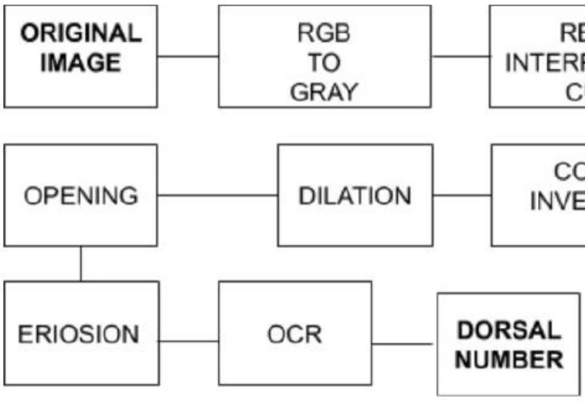
Process 2:



Process 3:



Process 4:


ORIGINAL IMAGE
RGB TO GRAY
RESIZE INTERPOLATION CUBIC
EQUALIZATION
OTSU BINARIZATION
COLOR INVERSION
DILATION
OPENING
ERIOSION
OCR
DORSAL NUMBER

Annex II

The following tables show the Re3 tracker people counting results and the duration of the tests, detailed for each video.

Timing on video 5972 (62 s)					
Timing (s)	Full Res.	x2	x4	x5	x6
Detectron	646.50	73.17	35.46	31.17	27.91
Track	736.62	204.81	81.37	64.99	54.71
Program	1858.88	476.27	120.00	81.04	62.78
Total	3242.00	754.25	236.83	177.20	145.40

Timing on video 5977 (31 s)					
Timing (s)	FR	x2	x4	x5	x6
Detectron	257.82	31.28	17.77	15.82	14.25
Track	355.36	95.17	40.17	33.51	30.19
Program	902.21	236.77	59.37	38.95	27.91
Total	1515.39	363.22	117.31	88.28	72.35

Timing on video 5978 (90 s)					
Timing (s)	FR	x2	x4	x5	x6
Detectron	654.15	84.26	45.85	41.65	38.39
Track	986.29	269.42	113.31	90.7	80.14
Program	2767.19	704.39	173.7	110.24	77.65
Total	4407.63	1058.07	332.86	242.59	196.18

Timing on video 6491 (30 s)					
Timing (s)	FR	x2	x4	x5	x6
Detectron	270.25	31.41	16.55	14.88	14.27
Track	387.21	107.76	43.86	34.81	30.81
Program	827.88	225.07	55.37	36.01	27.02
Total	1485.34	364.24	115.78	85.70	72.1

Timing on video 6492 (30 s)					
Timing (s)	FR	x2	x4	x5	x6
Detectron	244.41	29.64	13.35	12.48	13.39
Track	165.99	81.96	24.64	21.69	25.83
Program	532.39	218.66	34.83	22.87	25.50
Total	942.79	330.26	72.82	57.04	64.72

Timing on video 6493 (30 s)

Timing (s)	FR	x2	x4	x5	x6
Detectron	267.82	23.40	17.22	15.72	15.39
Track	408.31	68.72	44.55	33.14	27.83
Program	858.63	144.84	121.44	45.37	25.5
Total	1534.76	236.96	183.21	94.23	68.72

Timing on video 6494 (30 s)

Timing (s)	FR	x2	x4	x5	x6
Detectron	253.34	35.10	19.38	15.26	13.43
Track	263.79	129.15	53.34	43.58	27.96
Program	560.99	246.94	70.77	40.74	18.78
Total	1078.12	411.19	143.49	99.58	60.17

People Counting video 5972

	Ground Truth	FR	x2	x4	x5	x6
Total	69	65	59	58	58	57
Correct	69	63	57	56	56	56
Misses	0	6	12	13	13	13
False	0	2	2	2	2	1

People Counting video 5977

	Ground Truth	FR	x2	x4	x5	x6
Total	16	17	18	17	17	18
Correct	16	16	16	16	16	16
Misses	0	0	0	0	0	0
False	0	1	2	1	1	2

People Counting video 5978

	Ground Truth	FR	x2	x4	x5	x6
Total	57	59	55	55	56	54
Correct	57	55	55	53	54	52
Misses	0	2	2	4	3	5
False	0	4	3	2	2	2

People Counting video 6491

	Ground Truth	Original Resolution	x2	x4	x5	x6
Total	9	11	11	10	10	10
Correct	9	9	9	9	9	9
Misses	0	0	0	0	0	0
False	0	2	2	1	1	1

People Counting video 6492						
	Ground Truth	Original Resolution	x2	x4	x5	x6
Total	4	4	4	4	4	4
Correct	4	4	4	4	4	4
Misses	0	0	0	0	0	0
False	0	0	0	0	0	0

People Counting video 6493						
	Ground Truth	Original Resolution	x2	x4	x5	x6
Total	6	6	6	6	6	6
Correct	6	6	6	6	6	6
Misses	0	0	0	0	0	0
False	0	0	0	0	0	0

People Counting video 6494						
	Ground Truth	Original Resolution	x2	x4	x5	x6
Total	11	9	9	9	9	9
Correct	11	8	8	8	8	8
Misses	0	3	3	3	3	3
False	0	1	1	1	1	1