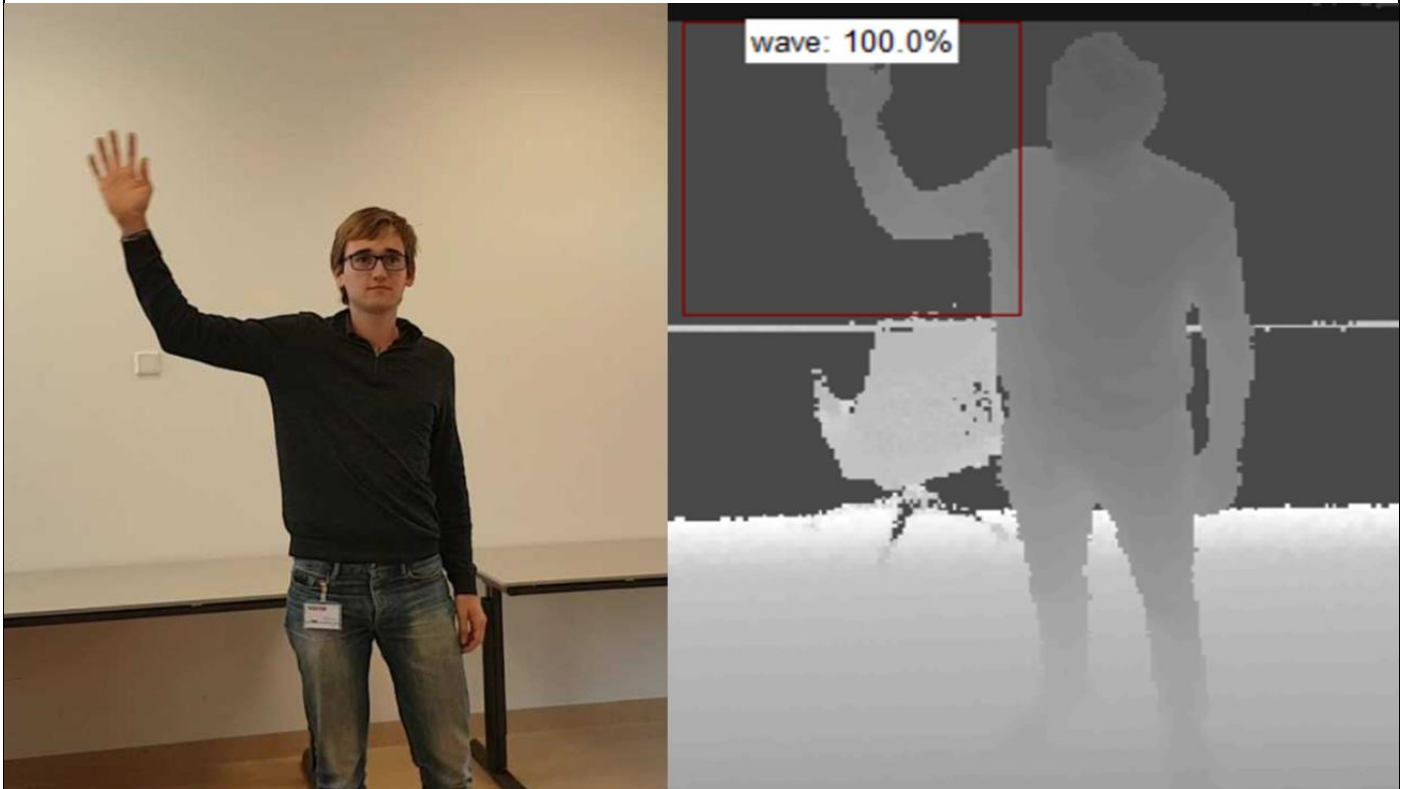


TNO innovation for life



Internship report

Human behaviour recognition on video



VALERY Louis
louis.valery@ensta-bretagne.org

ENSTA Bretagne
2 rue F. Verny
29806 Brest Cedex 9,
France

Acknowledgment

I would like to thanks Gertjan who has supervised me for 3 long months, explaining the job and answering my question, always smiling and joking.

I also would like to thanks Frank who answered a lot of my questions, and helped me to use RDA. Thank you for your patience.

Abstract

This report presents the result of a project aiming to assess the efficiency of the Kinect2 to detect human behaviours in indoor environments. The Kinect2 is a depth camera; the data of the sensor are computed thanks to a machine learning method based on motion quantification. This report explains the method, present the results and discusses them.

Summary

- 1. The company 5
 - 1.1. General presentation 5
 - 1.2. Activity of the Intelligence Imaging Department 5
- 2. Behaviour recognition with machine learning 6
 - 2.1. Presentation of the method 6
 - 2.2. To train a model: how to get a good model 11
- 3. Theoretical contribution..... 12
 - 3.1. Decision Trees 12
 - 3.2. Random Forest 13
 - 3.3. SVM algorithm..... 14
 - 3.4. Optical flow 16
- 4. Results and limitation of the method..... 17
 - 4.1. MAP 17
 - 4.2 Results 18
 - 4.3 How to detect different behaviors with the same model 21
- 5 Personnel interest 23
- Conclusion 24
- Références Bibliographiques..... 25

Introduction

Autonomous driving vehicles, drones, street surveillance, computer vision has more and more application. With machine learning and artificial intelligence, computers are not only asked to see anymore, but to detect and to understand. Most of human action can be characterized by a specific motion. That is why each behaviour can expect to be detected and correctly understood by the computer. Application field is huge: oversight of the elderly at home instead of in retirement home, detection of aggression in the streets, tracking of people based on their behaviour [1]...

The Intelligence Imaging department of TNO has worked on human behaviour detection for years, for many daily life applications. Until now the sensor used was a RGB camera; the aim of the project presented in this report was to assess the efficiency of a new sensor, the kinect2, to detect specific short term human behaviours in indoor environment. The main particularity of this sensor is that it is a depth camera instead of a RGB camera, a depth camera being supposed to be more efficient to detect the shape of people, so motion.

This project calls for advanced machine learning techniques and algorithms. They are based on motion quantification and are used to compute the videos and to classify specific behaviour. As this project is confidential, not everything will be explained in detail in this report. In particular the details of the code will not be discussed. This report explains in details the method used, presents the results and discusses them.

1. The company

I did my internship in the Netherlands at TNO, an innovative and research-oriented company. The purpose of my internship was to participate in a project that aimed to detect human behaviours on videos in small indoor environment.

1.1.General presentation

TNO is an independent research organisation which was established by law in 1932. It is a not-for-profit knowledge organisation created in order to support companies and the Dutch government with innovative, practicable knowledge. Focused on innovation, applied science and new technologies, TNO is an expert company in several domains which are in line with the challenges and goals of the national economic policy, often linked to social issues and challenges:

- Buildings, infrastructure and maritime
- Environment
- Defence, safety and security
- Energy
- Healthy leaving
- Industry
- Information and communication technology
- Traffic transport

The organisation also conducts contract research, offers specialist consulting services, and grants licences for patents and specialist software. TNO tests and certifies products and services, and issues an independent evaluation of quality. The company is funded

Today TNO is the leading research organization in the Netherlands. It has 2300 employees across the country, but has also branches in Japan, Canada, Belgium and Singapore.

1.2.Activity of the Intelligence Imaging Department

I worked in the Intelligence Imaging department, in The Hague, one of the 82 departments of the company. About 30 engineers and researchers work there, on several different projects:

- Detection and classification of the ships along the coast, thanks to a live video and machine learning.
- Detection and classification of crack and holes in the pavement, thanks to a laser sensor and deep learning. The aim was to indicate which road should be fixed or redo first.
- Live detection of the border between land and sea, land and sky, and sea and sky on video.
- Live tracking of ships on the sea
- Creation of Virtual Reality rooms, to simulate some industrial robots.

These examples show the diversity of the application domains of this department: defence, security, industry, traffic transport...

2. Behaviour recognition with machine learning

During my internship, machine learning was used to detect specific human behaviours. More precisely it was used to quantify motion and to classify actions. For instance we wanted to detect if the person on the video was jumping, waving or falling.

A machine learning method is always separated in two parts: the first one, the tricky one, is the creation of the model. Then the model can be use on examples that it doesn't know. All the theoretical points developed in part 2 are used to train the model. This part explains step by step how to create a good model.

The advantage of machine learning is that once you have a model it can be used forever, on not yet known examples and without human supervision. This part will present the method used to get such a model. The explanations of the theoretical points and specific algorithms are developed later, in part 3.

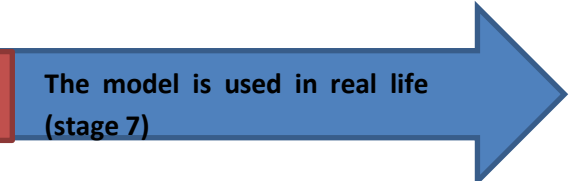
2.1.Presentation of the method

In this part, the whole method to create a model will be explained using an example. In this example we assume that the behaviors of interest are "wave" and "jump". All the other behavior will be called "others".

A training video is recorded on which there are the different behaviors of interest (stage 1). The algorithms will use this video to learn how to distinguish the different behaviors. After a calibration stage (stage 2 and 3), a detector indicate which part of the video is interesting (stage 4). Then the optical flow and random forest are used to quantify motion and the SVM algorithm classifies the behaviors (stage 5 and 6). It creates a model which will be used later to detect behavior on live videos (stage 7).

Creation of the model with human supervision (stage 1 to 6)

The model is used in real life (stage 7)



Stage 1:

Firstly, a video is recorded. On this video “wave”, “jump” and “others” can be seen. “Others” are essential to explain to the future model what another behavior can look like. It is the video used to train the model; it will be called “training video” or “dataset” (figure 1).

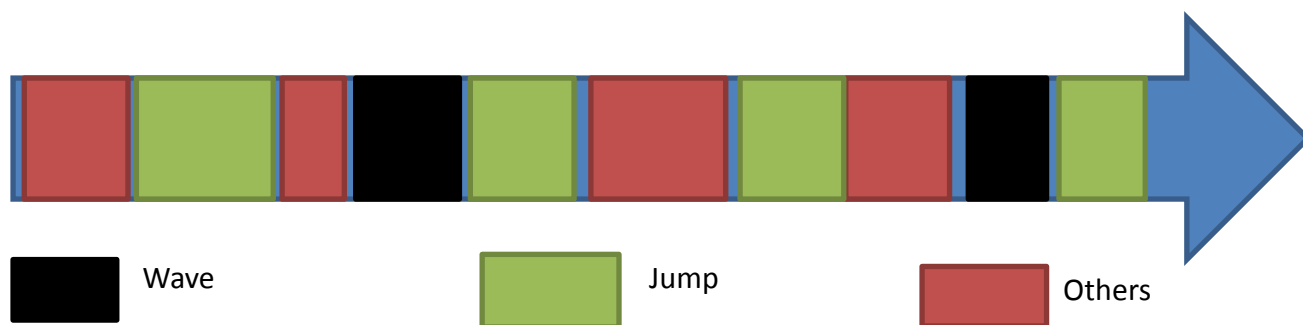


Figure 1: Interval repartition on the training video

Stage 2 (annotation, calibration and choice of the parameters):

Then the video is annotated, it is the part with human supervision. The behaviors of interest on the training video are clearly defined. On each frame a rectangle is drawn to explain where the motion is.

Human supervision is also to indicate:

- Size of human in back (in green on figure 2)
- Size of human in front (in green)
- Area of interest (where the behavior are supposed to be)

It is the calibration of the video

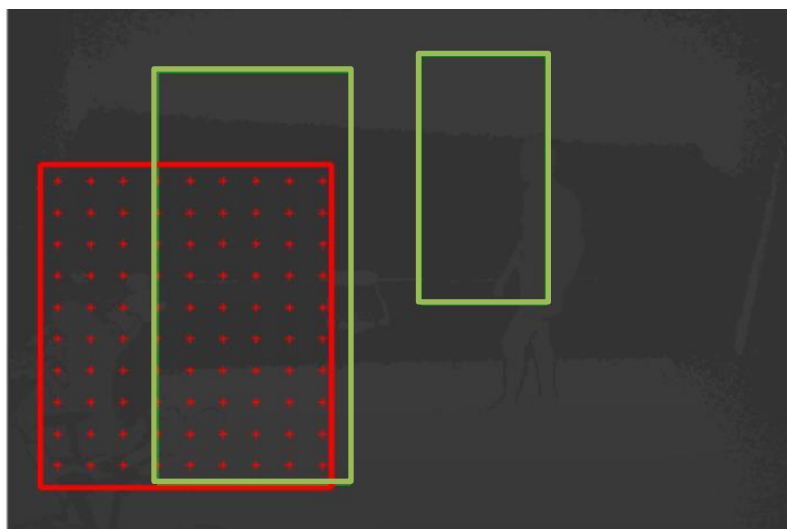


Figure 2: Results of the calibration of the environment (size of people in the front, in the back, and size of one bounding box, in red)

Finally many parameters are to be chosen by user. These parameters have a huge influence on the efficiency of the final model. One of the most important is the size of the box, named bounding box, which will be used to localize the behaviors in space and time. The red box on figure 2, is a box of 1.6 m * 1.6m (time dimension is not represented).

Stage 3 (draw grid):

Potential bounding boxes are drawn on each frame all over the frame creating a kind of grid (figure 4). Motion and behaviors will be analyzed and computed in each box. A box has 3 dimensions: height, width and time (figure 3). It means that each box contains a whole behavior, or at least a part

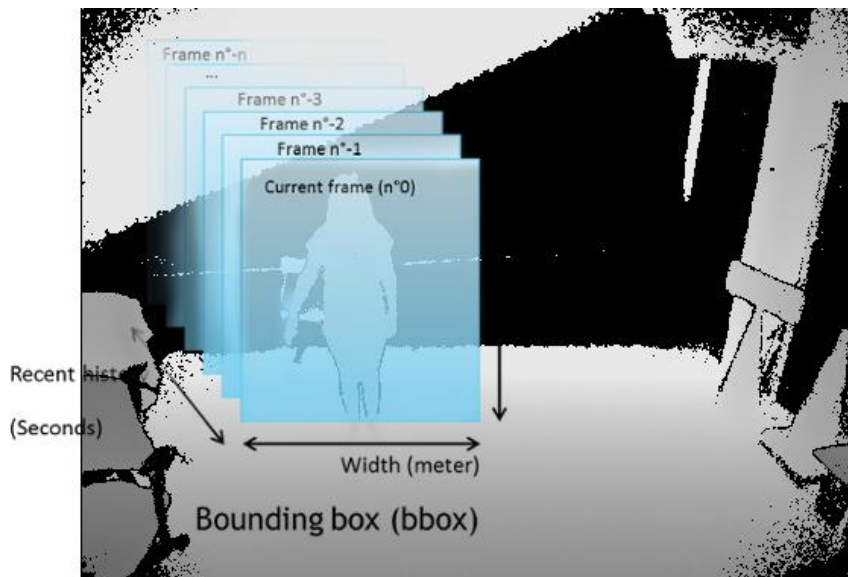


Figure 3: Bounding boxes (Bbox)

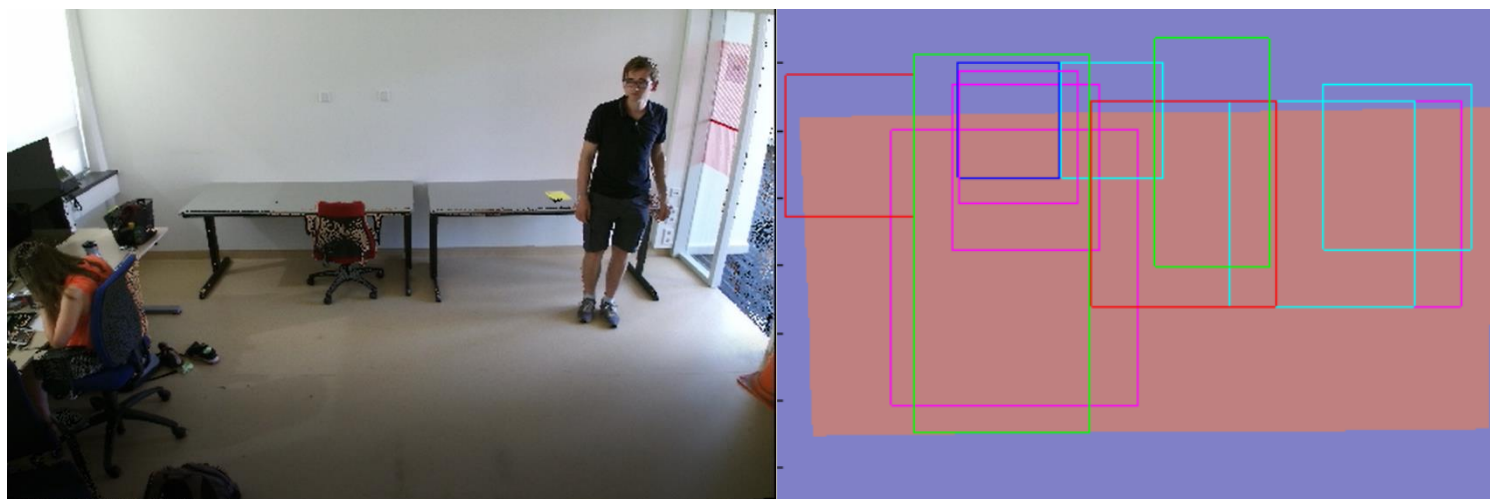


Figure 4: Grid created after calibration. In green: size of people in the front and the back. The squares are examples of Bbox created all over the area of interest (the red rectangle)

Stage 4 (detector): The aim of the detector is to reduce the number of relevant bounding boxes, in order to reduce computation time. The idea is to find a threshold based on motion detection able to detect all the “wave” and “jump”, and as few “other” as possible.

A score about motion is given. The higher the score is, the more important the motion is. Then the user needs to choose a threshold (figure 5). If the detection score is higher than the threshold, the motion is considered as relevant, if it is not, the motion is not computed. If the behavior is considered as relevant, it means that it will need to be quantified and classified later between wave, jump or other.

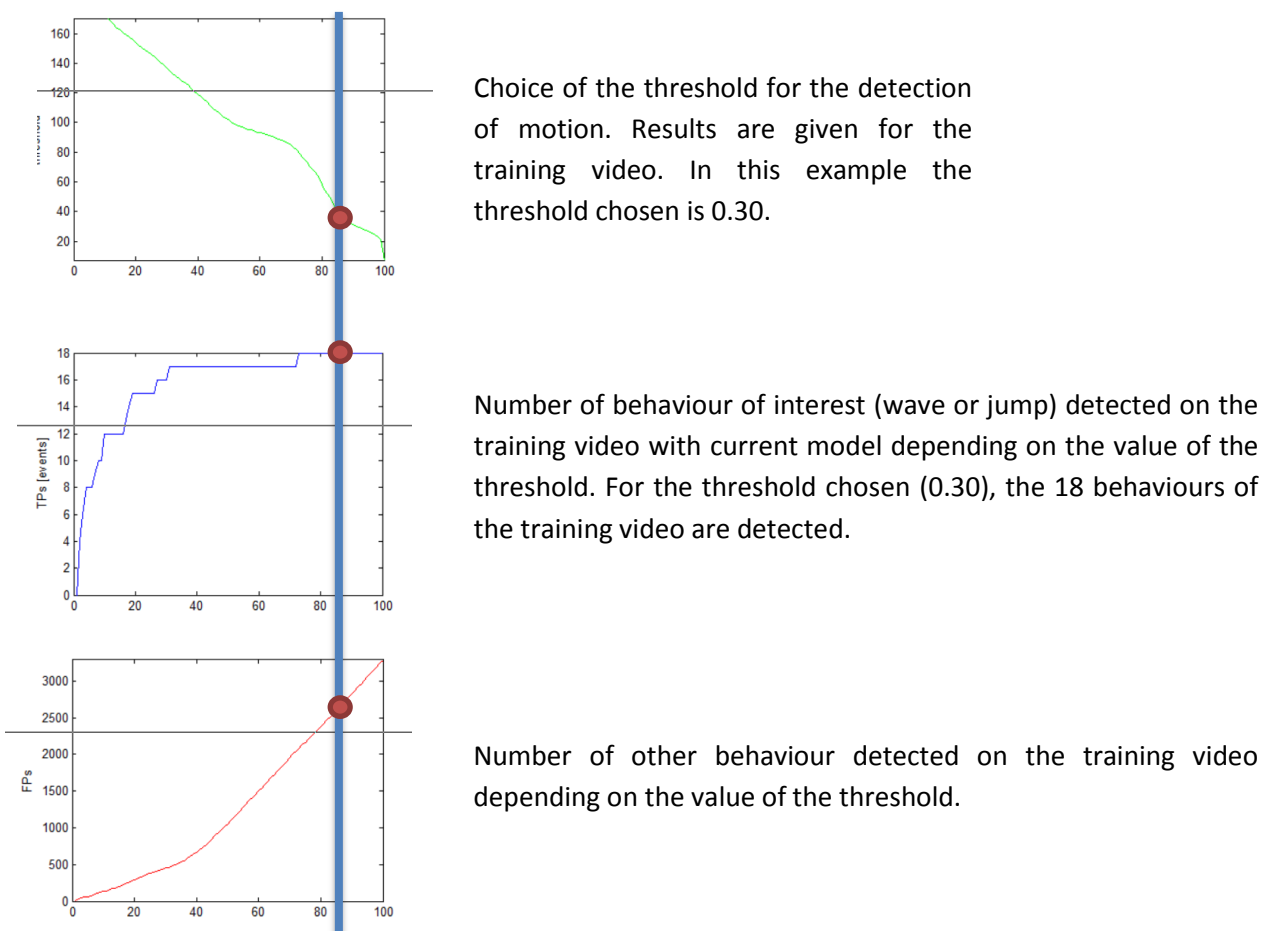


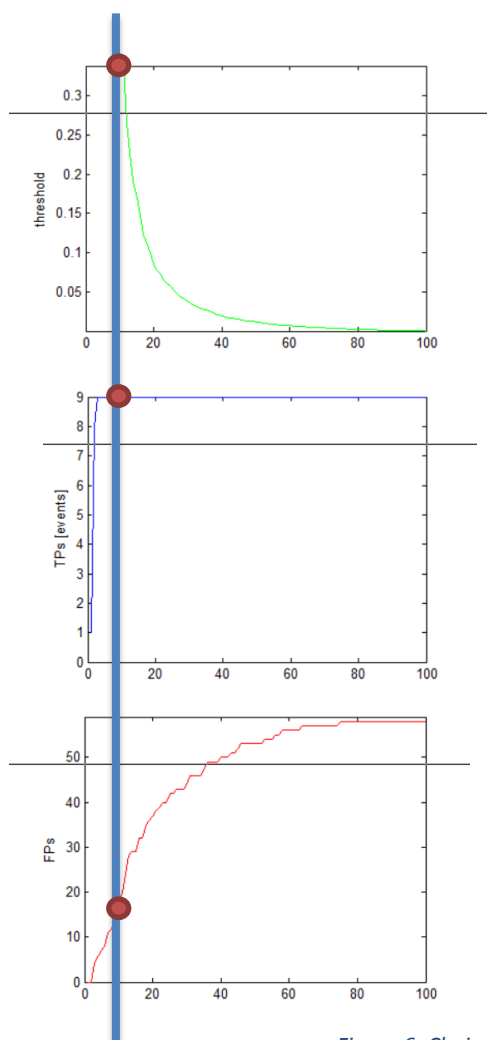
Figure 5: Choice of the threshold for the detector

Stage 5 (quantifier and classifier):

While there were hundreds of potential bounding boxes on each frame, there is now only one or two, thanks to the detector. **The aim of this stage is to classify the behavior linked to each bounding box.**

Optical flow is used to quantify motion in several points of each box, then a random forest uses the results to create a pattern.

Finally SVM algorithm classifies the behavior detected inside the bounding box, using the results of the random forest. For each bounding box, it gives 2 scores, one for each behavior of interest: “wave” and “jump”. If the score of the current behavior is higher than the threshold, the behavior is positive. To choose a threshold (figure 6), the model is tried on the training video, as you can see on figure 12, we just have to choose the threshold which give the maximum of “true positive” with a minimum of “false positive”.



Choice of the threshold for the wave classification. In this example the threshold chosen is 0.35.

Number of wave correctly classified on the training video with current model depending on the value of the threshold. For the threshold chosen (0.35), the 9 waves of the training video are correctly classified.

Number of False Positive depending on the value of the threshold. Number of behaviour classified as wave on the training video while they are not. With this choice of threshold we have 15 false positive.

Figure 6: Choice of one of the 2 thresholds for the classifier (wave threshold)

Stage 6 (live video):

After stage 6, a model has been created, meaning it can be use in real life. The user can now verify if the model that he has created is efficient. Indeed, a correct model is always good with the training data, that is to say that it is able to classify correctly all the behaviors of the training video, but the question is to know if it is able to classify behavior he never saw.

2.2.To train a model: how to get a good model

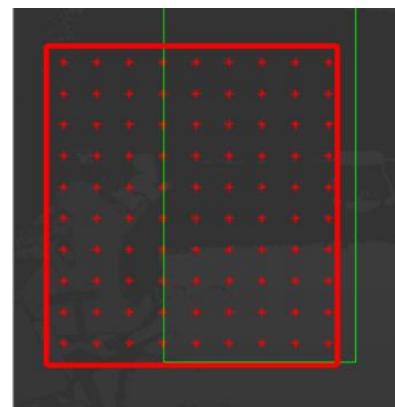
A lot of parameters are to be chosen by the user and have a stong influence on the efficiency of the model. In this part the main parameters will be presented. The choice of parameters is a balance between speed and accuracy. Indeed, even if we are creating a model, and not using it (so we do not really care about speed), these parameters will later have an influence on the live detection. If the processing time is to slow, detection will not be efficient, even if the model is supposed to be accurate.

The size of BBoxes:

There is always one optimal size to detect one specific behavior, but in our case we are trying to detect 2 different behaviors, so we need to find a compromise between the optimal size to detect wave and the optimal size for jump. The model will be less accurate than with just one behavior, but results show that it is still possible to find an efficient model.

Number of features:

During the stage of quantification, the optical flow is not computed everywhere in the Bbox. The number of feature is the number of time the optical flow is computed. The higher it is the more accurate the model is and the slower computation will be in life video. Here the red square represents a bounding box in 2 dimensions and the red points represent each feature.



The depth of the random forest:

The depth of the random forest can be chosen. The higher the depth, the more accurate the model, but the longer the computation. An over-fitting effect can also be seen. So we have to find the balance between accuracy and speed. If the computation is to slow, some frames are missed and the result becomes really bad. We have the same problem with the SVM algorithm.

3. Theoretical contribution

This part explains some theoretical points linked to the method developed in part 2. It explains what random forest or SVM algorithms are. A quick explanation of optical flow is also given.

3.1. Decision Trees

Decision trees are essential for random forest. So before talking about random forest, let us understand what a decision tree is.

A decision tree is a decision support tool representing a set of choices in the graphical form of a tree (figure 7). The different possible decisions are located at the ends of the branches (the "leaves" of the tree), and are reached according to decisions made at each stage. Decision trees are used for classification [4].

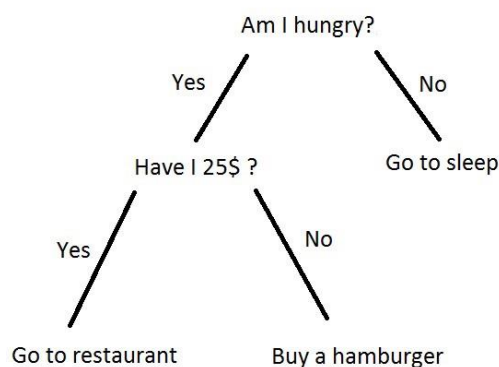


Figure 7: Decision tree example

They are commonly used in machine learning, in deep learning but are also a popular tool in operations research, especially in decision analysis, to help identify a strategy most likely to reach a goal.

Most of the time decision trees are not built manually, but created from to a data set. An algorithm analyses the data to find the good question to ask at each stage, to finally build the final tree. The result is most of the time perfect for the data set (if we use it on the data set, classification will always succeed), but the difficulty is to create a tree efficient for other data.

Let us illustrate that with an example: A bank want to create a decision tree to know if she can safely lend money to its client. It has a huge data set thanks to the former clients, with a lot of information on each client. Of course, it also knows if the formers clients were able to pay back the loan. Thanks to this data, the bank creates a decision tree, that is to say questions to ask to new clients in a specific order to know if they are reliable or not. If this tree had been used on previous clients it would have been able to predict its reliability, so we hope it will be the case for the next one.

3.2.Random Forest

Decision trees are a popular method for various machine learning tasks; however, they are seldom accurate. Indeed, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets. As a result, they have low bias, but very high variance (figure 8).

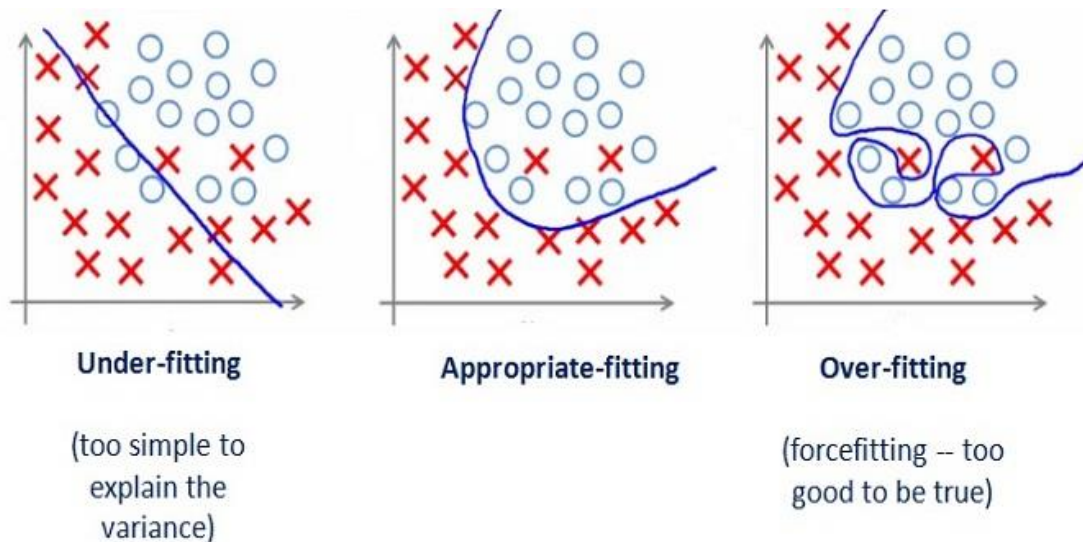


Figure 8 : Over-fitting illustration

The solution to such a problem is random forests. They are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

The algorithm creates a forest of decision trees trained on different subsets of the training data set and makes it somehow random (figure 9). Thus, this method adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds [3].

Let us come back to the bank example. Instead of using all the data set to create one single data set. This data set is split randomly in several subsets. Each subset is used to create a decision tree. You get a forest of decision trees. This is the model. Then try it one a new client. You will get a result for each tree, the can be different, but the final result will be the most recurrent one. On figure 9, the final result is that client is part of class B: he is not reliable although one of the trees said differently.

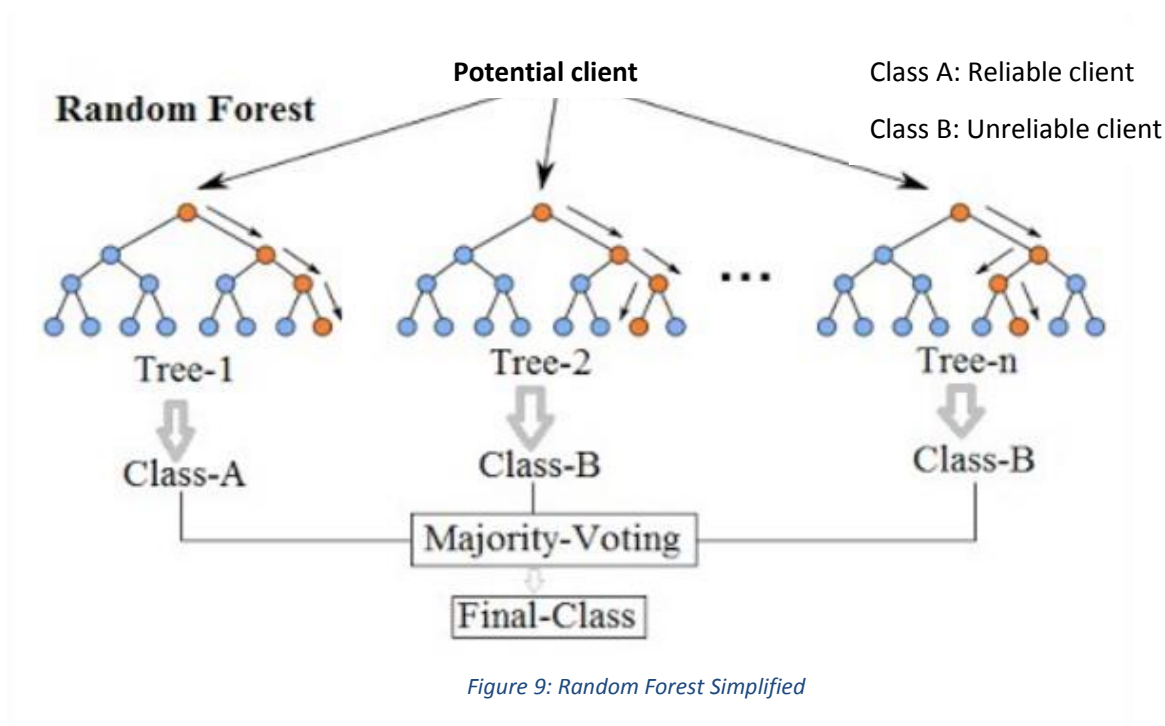


Figure 9: Random Forest Simplified

3.3.SVM algorithm

Support Vector Machines are classifiers based on two key ideas that allow to solve non-linear discrimination problems and to reformulate the classification problem as a quadratic optimization problem.

The first key idea is the concept of maximum margin (figure 10). The margin is the distance between the separation boundary and the closest samples. These are called Support Vector. In the SVM, the separation boundary is chosen as the one that maximizes the margin. The problem is to find this optimal dividing boundary, from a training/learning set. This is done by formulating the problem as a quadratic optimization problem, for which there are known algorithms [2].

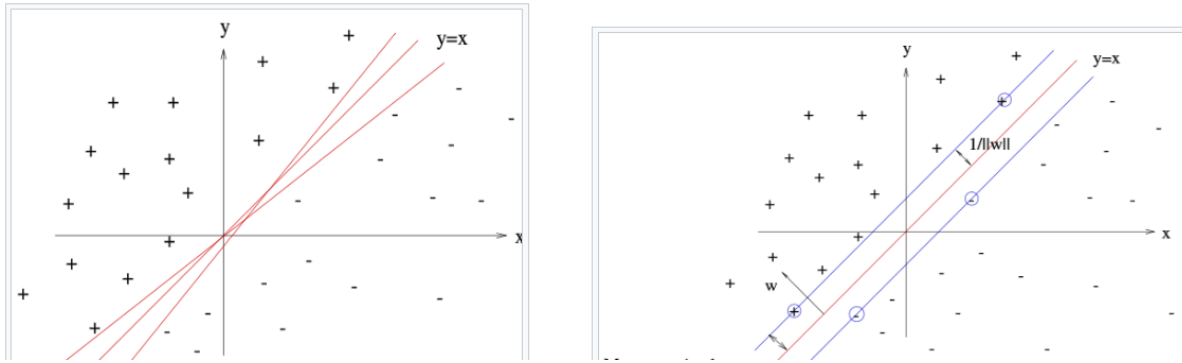


Figure 10: Maximum margin

In order to be able to deal with cases where the data are not linearly separable, the second key idea of the SVM is to transform the representation space of the input data into a larger dimension space (possibly of infinite dimension), in which it is likely that there is a linear separation (figure 11). This is achieved through a kernel function, which must respect the conditions of Mercer's theorem, and which has the advantage of not requiring the explicit knowledge of the transformation to be applied for the change of space. The kernel functions make it possible to transform a scalar product in a large space, which is expensive, into a simple evaluation of a function. This technique is known as the kernel trick [2].

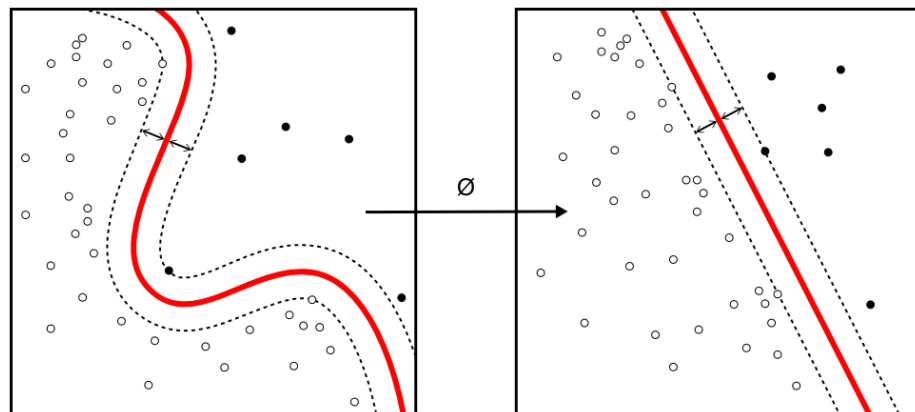


Figure 11: Kernel interest

3.4.Optical flow

The last theoretical point concern optical flow. Indeed, to classify the events, motion is the only thing taken into account. Shape and colours of objects are not used. In our case, motion is detected thanks to optical flow [5].

Optic flow is defined as the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene (figure 12). In this project, optical flow will be used to quantify motion and will be the main tool used for classification.

To improve the efficiency of the classification, the optical flow is normalised in space and time. For instance, if one frame is missed the optical flow (computed in a first time) is twice higher than in the reality, and the mistake is corrected thanks to the normalisation. Normalisation in space means that the difference of size between a person in the front and in the back is corrected.

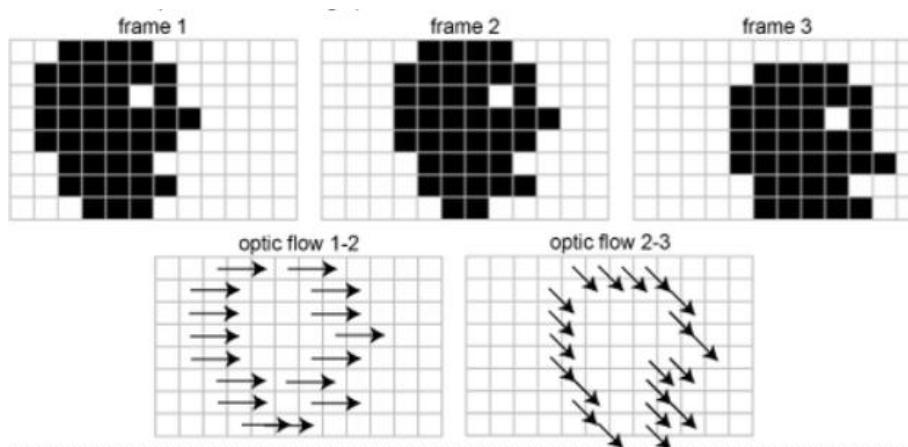


Figure 12: Optical flow

4. Results and limitation of the method

4.1.MAP

By following the first 5 stages of the method presented in part 2, a model is created. To know if this model is a good one or not, a score is computed based on tests on the training video. This score is named mAP (mean Average Precision) and gives an estimation of the model's efficiency.

To compute the mAP of a model, the training video is divided in n parts. One of them is left apart while the remaining is used to train the model. Then the model is tested on the remaining part of the training video, and a score is given (in white on figure 13).

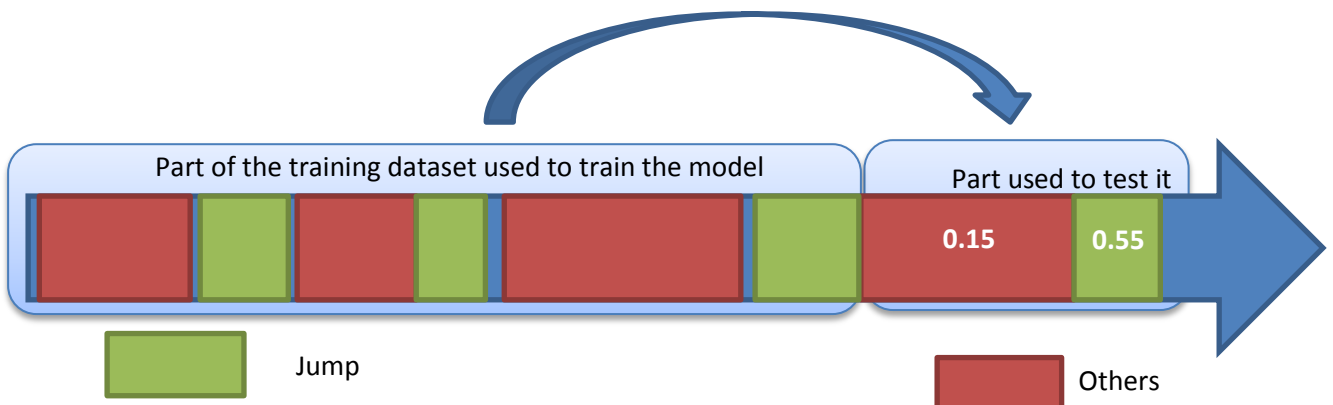


Figure 13: mAP, stage 1, to test a part of the training

This stage is repeated for the n-1 part not tested yet. And finally each interval of the training video is tested several times. The best score is kept. Finally the mAP is a score which show to what extend the scores given to each interval is relevant. If the scores given for the "jumps" are always higher than the score given for the "other", the mAP will be 100%. In the example of the figure 14, the 3rd "other" has a score higher than the last "jump", meaning that the model is not perfect. The mAP will be lower, around 96%. An example of compute is given on figure 13.

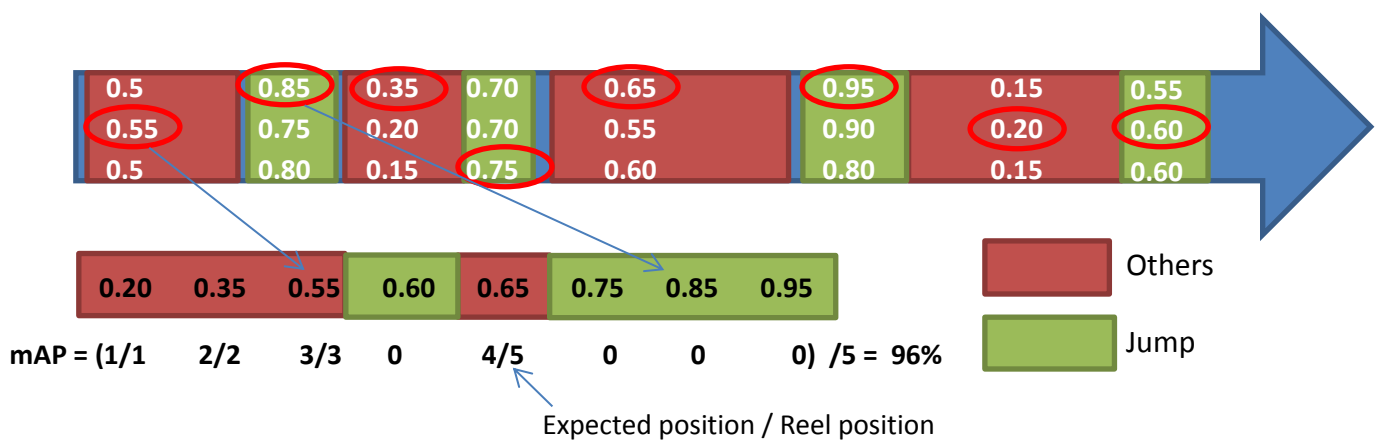


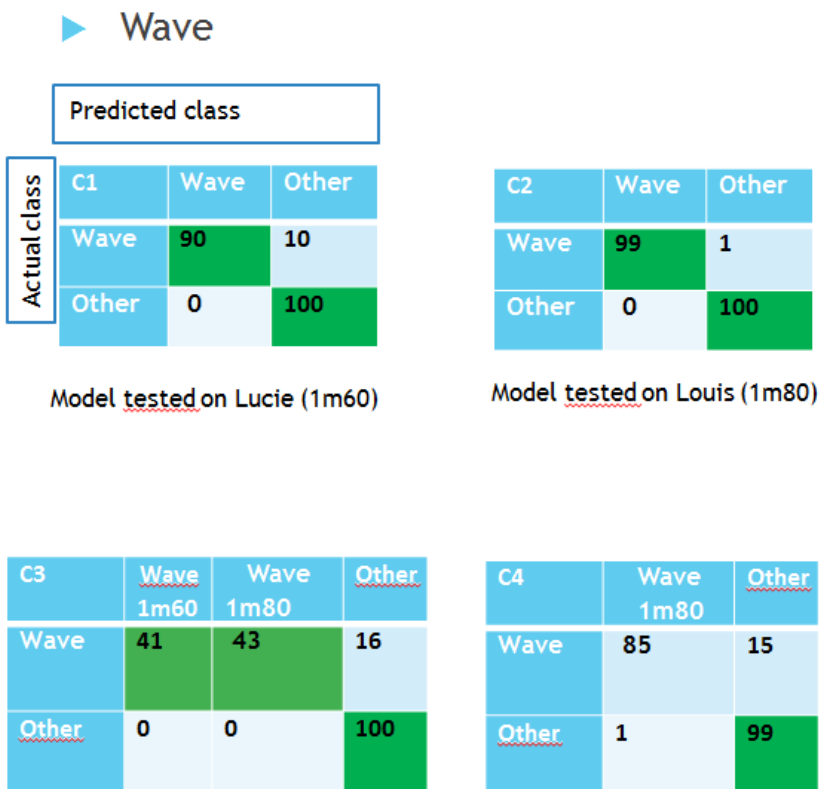
Figure 14: mAP, Stage 2, to give a general score to the model

4.2 Results

The mAP gives a good estimation of the model’s efficiency. It is really helpful to optimize the model, but at the end it is not enough to be sure that the model is good, mainly because the mAP is by testing the model on the same video than the video used to train it. Obviously the result is skewed.

To be determinate the efficiency of the model, it is tested on live video. The following figures are the results got for a model supposed to detect “waves”. As you can see it is really efficient, indeed, there are few false positives and nearly all the waves are detected.

These results have been got with a size of BBox of 1.4m * 1.4m * 2 seconds; 60 features per BBox, a depth of 5 for the random forest and a time-step at 0.



C1 : wave with a 1m60 person

C2 : wave with a 1m80 person

C3 : 2 people waving

C4 : 1 person waving, and another one present on the video

Figure 15: Results of the model for live video

The figure 15 presents the results of a model able to classify waves only, in several configurations. The 2 first cases are quite simple, with only one person. The case 3 is with 2 people waving at the same time, and the 4th with one person waving, another one walking or doing something else. They depend on the size of the person but remain globally really good when there is only one person, with an efficiency higher than 90%. With 2 people on the video, the model is a bit less accurate because it has not been trained for that, but the results remains quite good, with an efficiency higher than 84%.

► Wave and jump

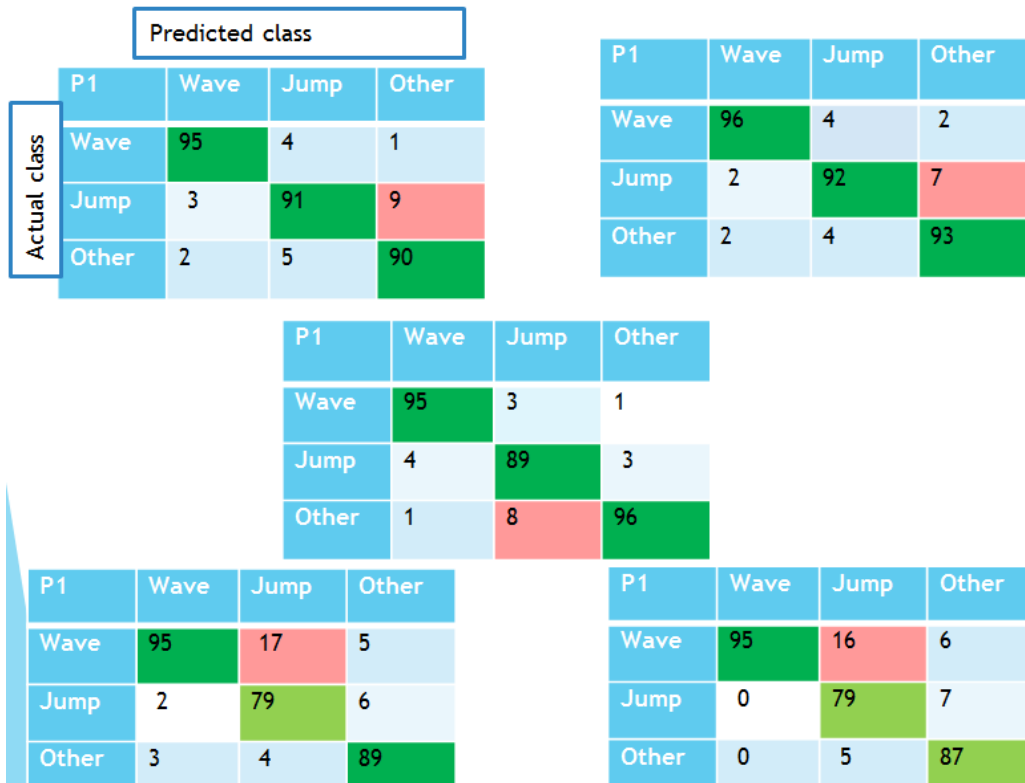


Figure 16: Results of the model for live video

The figure 16 presents the results for a model able to classify both waves and jumps. The results are given for several positions in the room because the efficiency of the model depends on the position of the person testing it. The main problem is that the model tends to classify the action as a wave instead of as a jump when the person is in the front of the scene. But the results remain acceptable insofar as the efficiency is always higher than 79%.

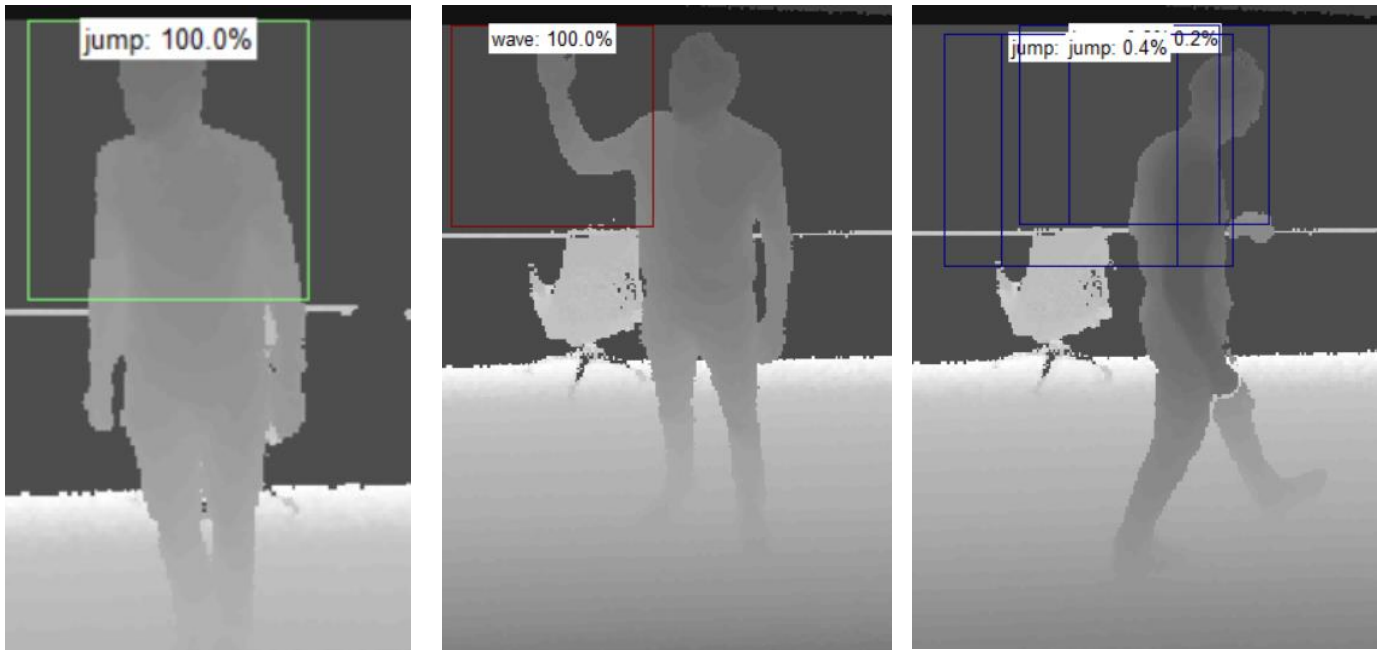


Figure 17: Results of the model for live video

4.3 How to detect different behaviors with the same model

We have now models able to detect one or two different behaviours. It could be interesting to try to detect three or more behavior with the same model. But is it possible?

If we want to detect and classify 10 behaviors, we could simply build a model for each behavior and use all the models at the same time, but it would require using 10 times the detector, quantifier and classifier stages on the same video, and would result in a huge computation time. So the idea is to classify 10 different behaviors with the same method. How to do that? What are the limits of the method?

To detect only one behavior, waves for instance, the choice of parameters is easy and to create a good model do not require a lot of time.

To detect 2 behaviors, wave and jump for instance, it becomes already much more difficult. Indeed the size of the Bboxes, chosen by the user before computation, is to be efficient to detect to different behaviors which have neither the same duration nor the same size. While a jump last only 1 second, a wave could last 10 (figure 18). It seems also more relevant to consider the whole body when detecting jump, while the arm should be enough to detect wave.

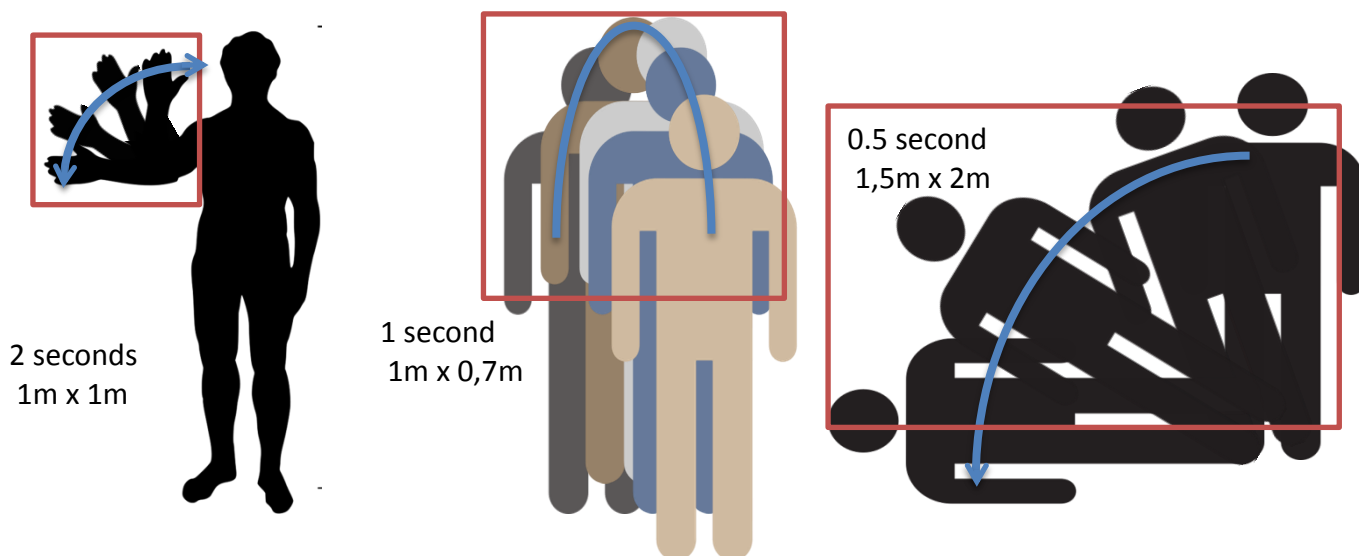


Figure 18: Problem linked to the choice of the BBox

Then we tried to detect wave, jump and fall. A good size for Bboxes is still more difficult to find and a new problem appeared: sometimes two behaviors can be quite similar. For example, it could be difficult to distinguish the patterns from the two behaviors of figure 19. Because of that, we often get false positive of fall while detecting waves or of jump while walking.

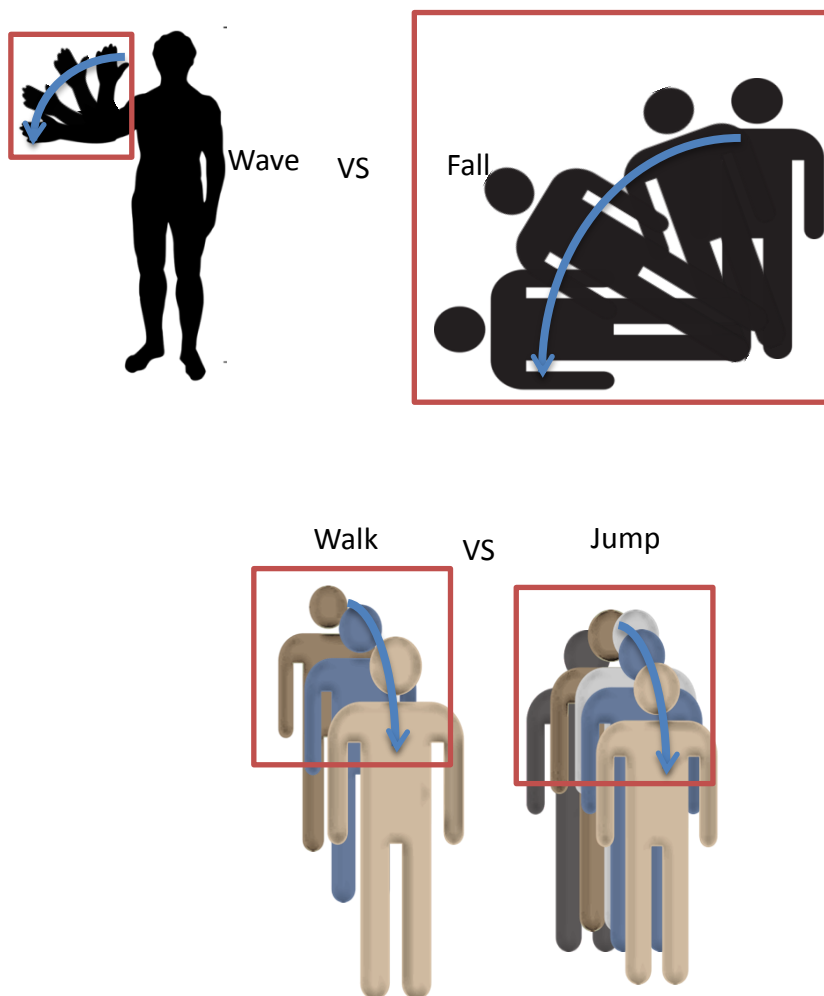


Figure 19 : Potential classification problem

5 Personnel interest

One of the main objectives of my internship was to learn about machine learning, and I learned a lot. Indeed machine-learning is increasingly used in the daily life and there are really interesting applications in robotics, which I am studying. For instance a quadrotor with a camera could easily detect and classify specific objects on the ground while mapping an area. The pictures from the AUV used to detect and destroyed submarine mines could be processed by a model able to classify the mines. To learn about machine learning is a good asset which could interest many companies.

Another goal of my internship was to discover the labour world and to work in a real company. Indeed I did my first year internship in a research center, and even if it is a professional environment, the way to work and the objectives are different from companies. Thus I worked on a project with real clients, with strong expectations. I learned a lot from the other employees, they explained me there jobs, how they work as a team and I realized the challenges and the economic issues of their projects. For instance my supervisor worked for the police administration of a local city, where there is a high criminal rate. The aim of his project was to use machine learning to detect the aggression in the streets. He worked with a team of 4 people and I followed their work for 3 months.

The last main objective of my internship was to improve my English. I worked and communicated in English and learned professional and specific vocabulary. After 3 months I feel a real improvement.

Conclusion

Finally the aim of my internship was to assess the efficiency of the Kinect2. Is a depth sensor better than a RGB camera? The results show that the Kinect 2 is efficient enough to detect motion and the data can be computed efficiently. One problem appends when the person on the video is next to a wall. It has the same colour than the wall and can't be detected that is why the depth sensor should be used with an RGB camera. Thus the assets of the two sensors could be used without there disadvantages, and the behaviours could be classified everywhere on the scene.

The different models show also that machine learning is efficient to detect one specific behaviour, but is less accurate when we want to classify two different behaviours. When we want to detect 3 or more, it becomes irrelevant. So to detect several behaviours it should be interesting to try to use several models at the same time, able to detect one (or two) behaviour. The computation time would increase but the results would be more accurate.

Références Bibliographiques

[1] Yu Kong and Yun Fu, (2018) Human Action Recognition and Prediction: A Survey, JOURNAL OF LATEX CLASS FILES

[2] Wikipedia, Support Vector Machine [en ligne], (page consultée le 15/08/2018)

https://en.wikipedia.org/wiki/Support_vector_machine

[3] Toward data sciences, Niklas Donges (2018) The random forest algorithm (page consultée le 08/07/2018)

<https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>

[4] Wikipedia, Decision Trees [en ligne], (page consultée le 15/08/2018)

https://en.wikipedia.org/wiki/Decision_tree

[5] Wikipedia, Optical flow [en ligne], (page consultée le 15/08/2018)

https://en.wikipedia.org/wiki/Optical_flow