**IA: Movement Recognition**

Ansiaux M. Godfroid M.

Advisor: Dr Slama R. [rim.slama@henallux.be](mailto:rim.slama@henallux.be)

Industrial Engineering Departement, Henallux College

Rue d’Arlon, 112, B-6760 Virton

*Abstract*—This project consists of making a movement recognition made by a human body. In this project we used some sports movement but a project like this one can also be use in an industry.

# Introduction

For our intelligent systems course, we were asked to do a project on a topic of our choice, but it had to be feasible in a computer camera and include the human body. Our choice was to use a database with movements of different sports, but an application like the one we thought could be used in the medical field or in an industry to help the staff to have more precise movements in their work.

Our main goal is to develop an intelligent program that will recognize these movements and display them on a screen where there will be the video of the webcam and the name of the sport

# State of the art

Several projects have already been carried out on movement recognition. To make our own project we consulted some of them which are:

1. The UCF (University of Central Florida) made an Action Recognition with a 101 categories data set. [1]
2. Hueihan Jhuang made an Action Recognition with ²

There are also other realisations, like the Xbox “Kinect”, which use some movement recognition.

# Our Solution

In this section we are going to explain our model and our interface.

## Network arcitecture

For our network architecture, we decide to make a CNN (Convolution Neural Network) model. During our research, we found that we made a Transfer Learning by using the VGG16 model.

The architecture of the VGG16 is like this:

* 16 convolutions with 3 FC layers
* 5 MaxPooling

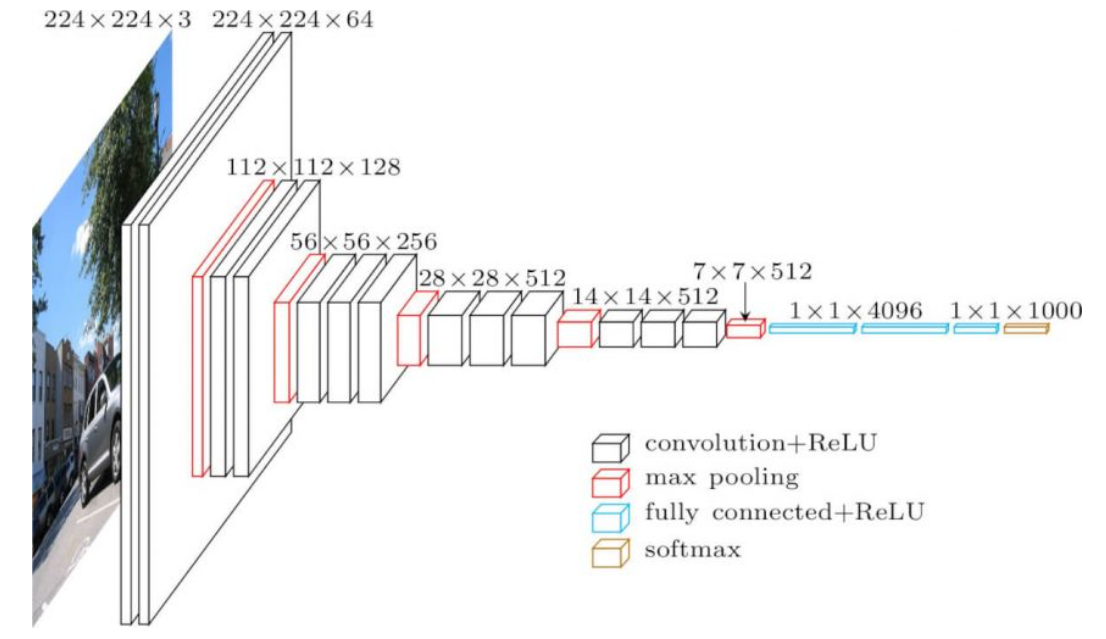


Figure 1: VGG16 Architecture

For the input we have a 224x224x3 picture and in the output we have a 7x7x512 picture.

After this pre-trained model, we defined an “output” architecture with 5 Dense functions, from 1024, which it is the size of the picture in the end of the pre-trained model, to 10, which is the number of classes that we have in our database.

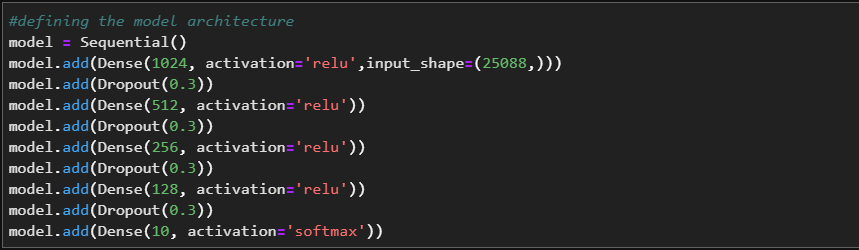


Figure 2: Dense Network

In all Dense function, expect the last one because it is our output, we use a dropout egal to 0.3, which that during the running time of the line, 30% of the neurons are disconnected.

## Interface conception and design

For the simulation of our interface, we have used the tkinter interface. We have also program this in python in our notebook. For the programming:

* To start with, we need to create the tkinter window on which we have the camera and the texts that indicate the sport
* After we have the programming of the type of window we want example the size, colour etc
* The last point is the programming of the different texts. (The text for the type of sport and the titles)

On the next image we can see our interface and the design it has

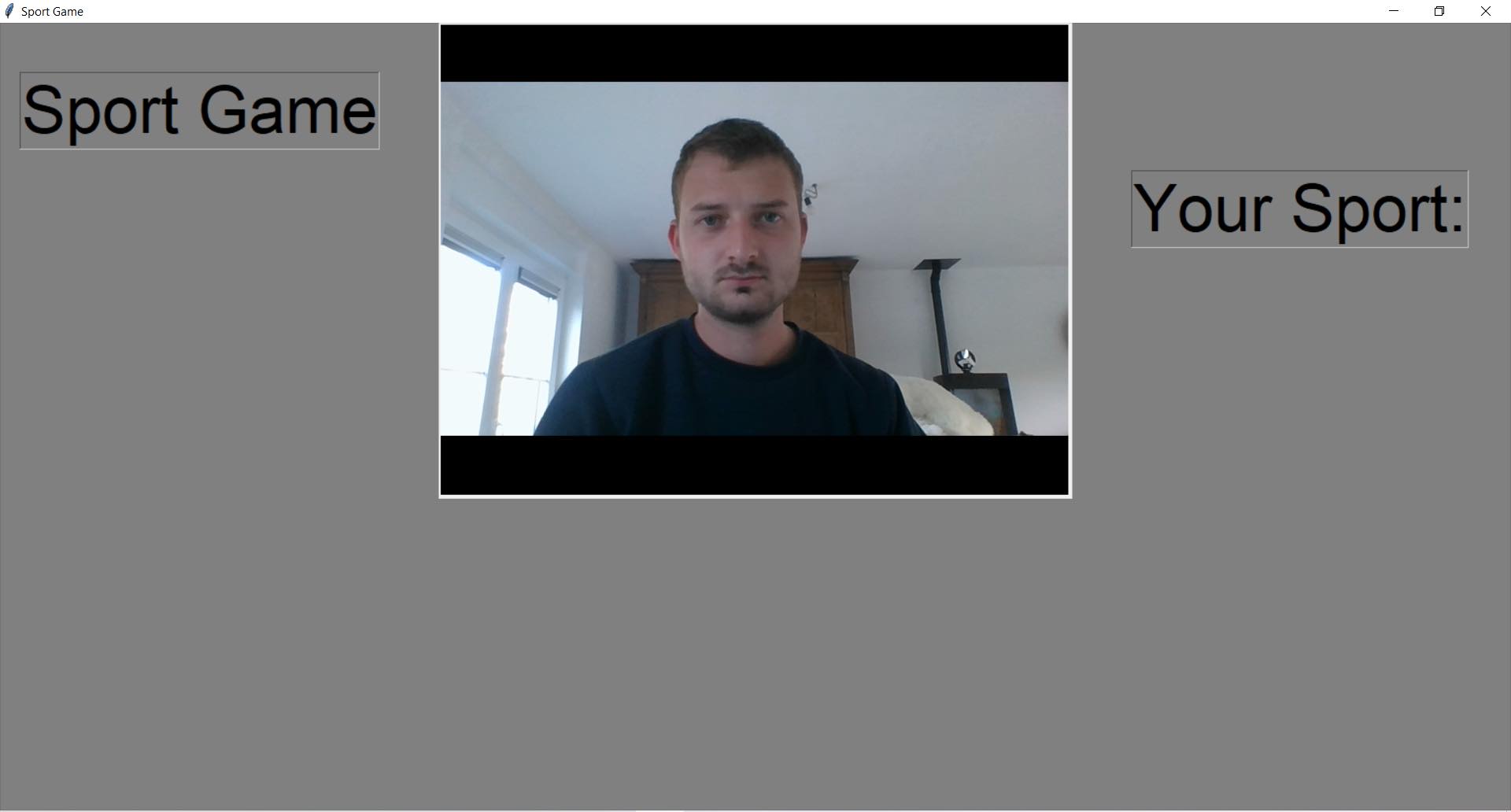


Figure 3 : Interface

To start the game, we need to press the button “A” on our keyboard.

When the game is launched, we can start the different gestures and can read the sport we do under the tittle “Your sport:”

To leave the game we only need to press the button “Q” on the keyboard and the screen turns off.

We have here an easy to use and understand interface. That is important everyone can use it.

# Experiments

In this section we are going to describe the database that we used and present different results.

## Dataset

To make our project, we need some data, so after some research we found 2 databases that we can use to realise this project.

We decide to mix the 2 databases because several movements were not possible to make in front of a computer, like kayaking movement our riding a horse, and if we did not mix the 2 databases, we would not have had enough class in the project.

So, in this project we have 10 different class of movement:

* Bend
* Body squats
* Boxing
* Juggling
* Jumping Jack
* Lunges
* Jumping
* Wave 1, which is a one arm movement
* Wave 2, which is a two arms movement.

In the different class there are many videos of the movement we want to recognise with our code.

To make our prediction, we decide to cut the videos into different picture with a size egal to 224x224x3, we also use 80% of the picture to make the training of our model and 20% to make the test after the training.

Une image contenant bâtiment, extérieur, béton, pierre

Description générée automatiquement

Figure 4 : Example of picture from video

## Results

During our test part, we obtain a 98% of accuracy. This value has been attempted with a batch size egal to 10 et 20 epoch.



Figure 5 : Test result

When we discovered this result, we were very happy because it is a really good percentage.

But, when we tried our model on new video, capture on our webcam, we saw that the result obtain was not the result we expected.

As you can see on the picture below, our model predicts that the man in front of the camera is making squat, but he is doing nothing and just watch the camera.

Une image contenant texte, capture d’écran, télévision, écran

Description générée automatiquement

Figure 6 : Prediction error

# IMPROVEMENTS

During the first programming we have done some errors. The first one is an error by the acquisition of the data from the webcam. In fact, we have worked with frame by frame and so we do not have good accuracy because it's more about position recognition than motion recognition. To solve this problem, we could work with the LSTM system. The long short-term memory allows for in-depth learning. It has feedback neurons. It works on sequences of data which makes it possible to work on voice recognition or in videos for example. Therefore, we need to work with this to improve our AI.

Our second problem is the webcam. When we did various tests, we realized that we problem to recognize the whole body. Indeed, we have used the haarcascade to frame the whole body and we could not do it when we were further away from the camera. We should surely have a more accurate long-range camera.

The last main improvement is the database. The problem with our database. It is not big enough and too old so we could use videos we make and do some data augmentation to have a more complete one which would increase the accuracy of the prediction.

VI. CONCLUSION

In conclusion, we can say we have worked on a super project. We have learned a lot of different things which is very interesting for us. Thanks to this course we learned to work in a different way but also to think differently in order to achieve our goals. During this project but also throughout the year in this course, we understood how the basics of artificial intelligence worked. Now, we know how to link things that could be useful at home or in everyday life.

We are disappointed that we were not able to get our project up and running. We know that the problem we have comes from the basis of our programme on data acquisition. We also know what mistakes we must not make if we start this type of project again.

We are happy to have worked on such an interesting project, even if we have not reached the end of the project.

REFERENCES

1. UCF, “UCF101 – Action Recognition Data Set”  
   URL: <https://www.crcv.ucf.edu/data/UCF101.php>
2. Hueihan, “ Action\_Recognition”

URL: <https://github.com/hueihan/Action_Recognition>

LIST OF FIGURES

Figure 1: VGG16 Architecture 1

Figure 2: Dense Network 1

Figure 3 : Interface 2

Figure 4 : Example of picture from video 2

Figure 5 : Test result 2

Figure 6 : Prediction error 2