Project report - AutoPylot

Alexandre Girold
Mickael Bobovitch
Maxime Ellerbach
Maxime Gay

Group: Autonomobile

March 2022

Contents

1 Introduction

1.1 Project presentation

Autonomous vehicles and more specifically self-driving cars have grasp the attention of many people for good or ill. In this spirit, we have decided with the Autonomobile team to create our first ever project, AutoPylot. The name of our team is of course full of meaning in that regard. Autonomobile is a two-word name, the first one a French word for autonomous : "Autonome", the second one a French word for car : "Automobile". These two-word combined literally mean Autonomous car.

What is AutoPylot's goal ? Drive itself on a track and win races. It may, at first glance seem very simple but not everything is at it seems. Yet we will try to make it as easy to understand as possible, without omitting crucial information. To achieve our goal, we need to solve many other problems. Those problems can be separate into two distinct groups.

The first one would be the software part. Indeed, in this project we will need to learn and acquire certain skills, from teamwork to coding in different languages. With those newly acquired skills we will be able to bring machine learning to our car to make it drive itself. This leads use directly to our second part, the more tangible one : hardware. Indeed, as we will progress in our work, we will need to see the results of our work in real life condition. This means implementing our code to a functioning car which will be able to race on a track.

This project will lead by a team of four young developers, Maxime Ellerbach, Mickael Bobovitch, Maxime Gay and Alexandre Girold. In this project work will be divided equally amongst all of us, sometimes we will have to work together to achieve our very tight time frame.

1.2 Team members

1.2.1 Maxime Ellerbach

I am a curious and learning hungry person, always happy to learn and collaborate with new people ! Programming, robotics and tinkering has always attracted me. Writing code and then seeing the results in real life is something that I find amazing ! I had multiple projects in this field : Lego Mindstorms, a robotic arm, more recently an autonomous car and even a simulator in unity to train even without a circuit at home ! Even if I know quite well the domain of autonomous cars, there is always something new to learn. I look forward working with this team full of hard-working people on such a fun project !

1.2.2 Mickael Bobovitch

Roses are red. Violets are blue. Unexpected "Mickael BOBOVITCH" on line 32. Hello I am a French Student with Russian parents. Lived half of my life in Moscow. Passionate in web dev, servers, and business. Started programming at 13 years old. Created many projects. I like to learn everything, from AI, to UI, from Hardware to Software. Actually I am like OCaml, you need to know me well to appreciate me.

1.2.3 Maxime Gay

I am 18 years old, and I am crazy about investment, finance and especially cryptocurrencies and blockchain. I already worked with a team on different Investment projects and during summer Jobs, but this is the first time that I am working on such a project. Furthermore, I am a beginner in computer Science and autonomous car. However, I am impatient to learn new skills with this incredible team.

1.2.4 Alexandre Girold

I am already getting old. I am 19 years of age, yet I am full of resources. I am delighted to be able to learn something new. There are many things which I enjoy from programming to geopolitics. I know this project will push me toward a better me and make great friends along the way.

1.3 State of the art

In this section, we will try to see what was previously made in this sector of industry. It would not be realistic to compare our 1:10 project to real sized cars such as Tesla's, simply because in a racing environment, we don't need to deal with such an amount of safety: pedestrian detection, emergency braking, speed limit detection and other. So we will only see miniature autonomous racing framework that we would likely race against.

The most known is called "DonkeyCar", created by Will Roscoe and Adam Conway in early of 2017. Most of the models trained with DonkeyCar are behavior cloning models, meaning models that tries to replicate the behavior of a driver. This method uses a big amount of images (input) associated to steering angles and throttle (output), it requires the user to drive the car (collect data) prior to training the model: no examples means no training. The lack of training data often leads to the car leaving the track.

One other framework worth looking at is one created by Nvidia called "JetRacer" released in 2019. It uses a different approach from DonkeyCar where the user annotates the images by hand by clicking on where the car should go. The model used is similar to what DonkeyCar uses: a Convolutional Neural Network with one input (the image) and two outputs, one for the steering angle and one for the throttle to apply.

Both of those frameworks are written in python and use packages such as Tensorflow and OpenCV, we will also use them in our project.

2 Realized tasks

2.1 Telemetry Server

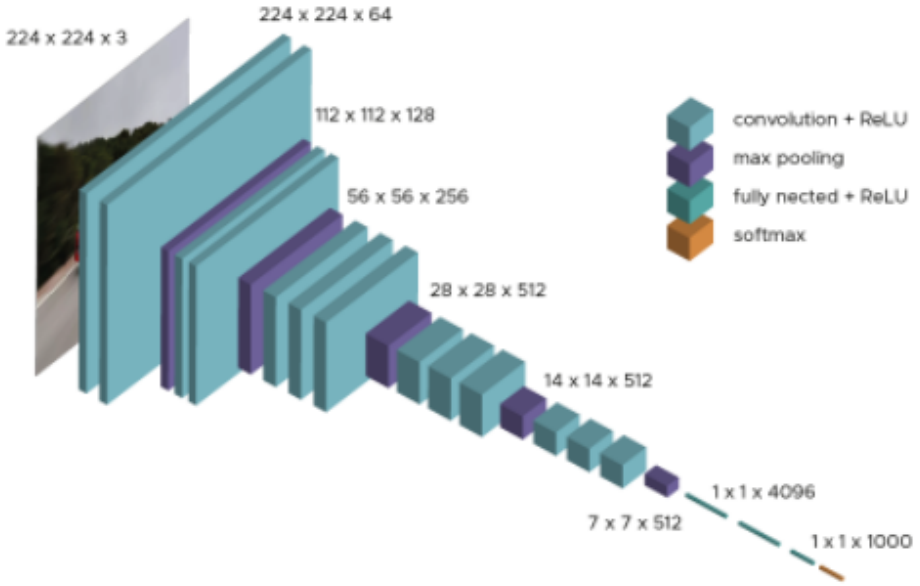
2.2 Theory behind Convolutionnal Neural Networks

2.3 Model architectures

In order to create my model, I was inspired by the VGG-16 model.

This model was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University in 2014. It won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) the same year. The model achieves 92.7

Architecture of VGG-16 model :



The VGG-16 architecture is composed of 13 convolutional layers with a 3x3 Kernel separated in five groups.

Between each group there is a pooling layer with a 2x2 size to reduce the size of the filter during the learning phase. At the end of those convolution and pooling layers, there are three Fully-Connected layers and an activation function softmax to determine the class of the image.

The problem of this model is that it is really slow and it is even slower in our case because it has to work with our car which does not have a lot of resources. Therefore, I decided to change the VGG-16 Model a little bit. I starte by removing some convolutional layers.

My first convolutional layer has a kernel of 5x5 and not 3x3 to have a filter with a bigger size at the beginning. Furthermore, I start with only four kernels on this first layer to reduce the cost of energy.

Then with the others layers, I increased progressively the number of kernels and I reduced the kernerl size to 3x3. Moreover, I apply a stride of 2 for the first four layers and a stride of 1 for the last two. At the end, I use two Dense layers with an activation function "relu" like the others layers.

Furthermore, I tested many different optimizer like SGC, Adadelta, Nadam and Ftrl but I decided to use the Adam Optimizer because it gave me the best result. The Adam optimizer involves a combination of two gradient descent methodologies

```python
def Max_model():
    inputs = []
    outputs = []

    inp = Input(shape=(120, 160, 3), name="image")
    inputs.append(inp)

    x = Cropping2D(cropping=((40, 0), (0, 0)))(inp)
    x = BatchNormalization()(x)

    x = Conv2D(4, kernel_size=5, strides=2, use_bias=False, activation="relu")(x)
    x = Conv2D(8, kernel_size=5, strides=2, use_bias=False, activation="relu")(x)
    x = Conv2D(16, kernel_size=3, strides=2, use_bias=False, activation="relu")(x)

    x = Conv2D(32, kernel_size=3, strides=1, use_bias=False, activation="relu")(x)
    x = Conv2D(48, kernel_size=3, strides=1, use_bias=False, activation="relu")(x)
    x = Conv2D(64, kernel_size=3, strides=1, use_bias=False, activation="relu")(x)

    x = Flatten()(x)  # construct vector from the matrix
    x = Dropout(0.4)(x)
    x = Dense(100, use_bias=False, activation="relu")(x)
    x = Dense(50, use_bias=False, activation="relu")(x)

    y = Dense(1, use_bias=False, activation="tanh", name="steering")(x)
    outputs.append(y)

    # Create the model
    model = Model(inputs=inputs, outputs=outputs)

    # Compile it
    model.compile(
        optimizer="adam", loss="mse"
    )  # can change the optimizer adam by stg else

    return model
```

2.4 Model wrappers

2.5 Load data during training

2.6 Training Process

2.7 Data Augmentation

3 Planning

3.0.1 What is next ?

The objectives we set ourselves for this presentation were acheived, for the next intermediate presentation we plan to finish what we are currently working on meaning The telemetry server and the logging. Moreover, we also plan to have a working prototype of the whole car including the AI part with the development of a basic convolutional neural network in a first time. This means we will have to create a model, then have a script to train it using collected data and finally a script to drive our car using this trained model.

3.0.2 Races

Tasks	Race 1	Race 2	Race 3	Race 4	Race 5	Race 6
Code controlled motors and servo	75%	100%				
Drive the car with a controller	25%	100%				
Data collection		50%	100%			
Telemetry server		25%	100%			
Logging		25%	100%			
Data processing and augmentation			50%	75%	100%	
Basic Convolutional neural network			25%	50%	100%	
Advanced models and optional objectives						50%

3.1 Presentations

Tasks	1st presentation	2nd Presentation	Final presentation
Code controlled motors and servo	100%		
Drive the car with a controller	100%		
Data collection	75%	100%	
Telemetry server	25%	100%	
Logging	25%	100%	
Presentation website	100%	Update	Update
Data processing and augmentation		75%	100%
Basic Convolutional neural network		50%	100%
Advanced models and optional objectives			50%

4 Task allocation

Tasks	Mickael B.	Maxime G.	Alexandre G.	Maxime E.
Low level car control				x
Driving with a controller		x	x	x
Dataset handling		x	x	
Data processing	x	x	x	x
Data visualization				x
Telemetry server	x			x
Logging	x			x
Presentation website	x			
Convolutional neural network	x	x	x	x
Main control loop	x			

5 Conclusion

To sum up, Autonomobile team improved the control of the car with controller, furthermore the data processing is working flawlessly allowing us to load and save images and metadata. Moreover, our presentation website is ready, it includes the presentation of the team, some links to download our project and even a road map. Nevertheless, the hardest part is yet to come, indeed we have to work on the AI part of the car and on the telemetry website to have it working by the next project defense.
The telemetry website is important in order to visualize data to know what is happening inside the car at any moment. We will have to work and learn a lot on this topic which is fascinating.

To make a long story short, we spent a lot of time on this project, which comport many sections that are important for the realization of this project, and we will do every thing to succeed.