Ecole Supérieure Privée
d'Ingénierie et de Technologies

Report

Data Science Project

Reverse Vending Machine

Moez Abid

Sarah Abidli

Fedi Bayoudh

Wissal Bayoudh

Montassar Thabti

Mohamed Iheb Bousnina

College Year 2019 - 2020

Table of contents

Figure Table

General Introduction

Reverse vending machine is a concept or an idea which inculcate the habit of recycling the waste materials. It will be working by taking recyclable waste into the machine and gives a useful thing as a token of appreciation. The aim of this project is to design and fabricate a reverse vending machine which takes recyclable waste into the machine and displays a token of appreciation.

The machine can accept a plastic bottle of 90mm diameter without cap and tin cans can be accepted and crushed and stored. The machine has a capacity of storing 50 plastic bottles and 50 tin cans. There basically two parts, one is the mechanical part and the other is the electronics part. The mechanical part is used to crush the recyclable waste which is kept in the machine so that more plastic and cans can be recycled and stored. The electronics part which consist of sensor and microcontroller, is used to take the correct input and segregate the waste into its respective categories and give a token of appreciation as a LCD display.

The whole system is automated by the help of electronics. Combining both parts will give a reverse vending machine. Reverse vending machine will be working by taking recyclable waste into the machine and gives a use full thing as a token of appreciation. With limited resources in the world, we need to start preserving them and put an end to wastage. Being encouraged to recycle through a rewards system.To encourage recycling process we are designing and manufacturing reverse vending machine.

Chapitre 1

Business Understanding

Plan

Introduction

This first chapter is dedicated to the presentation of the preliminary study which amounts to the first stage of our project titled Reverse Vending Machine. First, we establish the business objectives that we aim to fulfill by capturing the project's goals. Next, we will introduce the project's context, data source identification and description and the system architecture.

1.1 Project context

In order to respond to the problem of waste management, an idea of creating an intelligent trash was made. This project offers a credible and hopeful solution to today's sustainable development goals. The main problem was the lack of sorting, which means that the waste is either burned or piled up in a landfill, so the main task of this machine is to capture the waste and the classifier according to their types : Plastic, glass, ...etc Sorting waste and recycling it are crucial issues of our century. It is therefore important that sorting becomes a daily gesture.

Within Our Smart Reverse Vending Machines, we use a smart camera which recognize material. When an object is out in the receiver unit of the machine, electronic recognition system evaluates the object and compares it with models in the system database if it's classified as recyclable the machine accepts the objects and stores it separately depending on material. By compressing plastic and aluminum waste, the machine provides more space for the waste material.

Figure 1.1: Process of the reverse vending machine

1.2 IBM Master Plan methodology

As the use of Data Science in the business context is constantly growing, the need of setting a business-oriented strategy is becoming a very important step in order to limit the huge number of existing algorithms and therefore, understand and answer the client's needs.

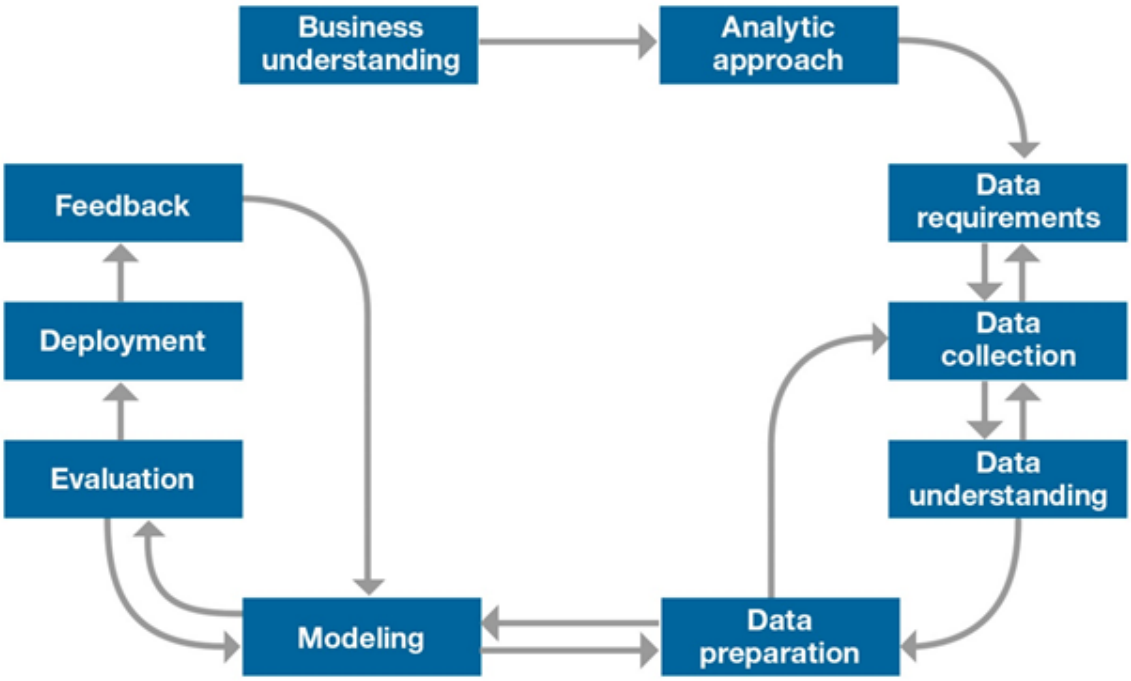

Figure 1.2: IBM Master Plan cycle

1.3 Business objectives

This proposed project is to the development of recycling machine for bottle with reward system. This project was developed by giving an instant reward to the user and provides receipt with user matrix number and quantity of recycled empty bottle. This machine also developed to boost recycling activities among the student. Instead, there are several objectives as follows :

- Leasing machines to organizations : it creates a great buzz monthly and would be an efficient way for the local companies not to invest directly. By offering renting in couple months machines can pay back easily. For long term, it's more profitable then resell model.
- Managing program model could be established by distributor within various business scenarios,running self owned machines has great potential of continuous profit generation.
- CO-OP model mainly aims to work with recycling companies and waste collection companies. Distributors can establish an operation model and partnership with them. CO-OP model assist distributor to buy machine with lower cost by receiving investment from these companies, and they can supply materials continuously.
- Geo-localization can be very useful to companies and to the project in this case, because this can provide additional information about the consumption of products for each region. This way, product advertisements can be aimed in a better manner, and we can get additional information about the habits of each type of client according to his position/location.

1.4 Conclusion

This first chapter gave us an overview of the general context of our project, we clarified our objectives and the needs of our project that was done by specifying the objectives of the business and the data source identification.

Chapitre 2

Building the RVM (IoT)

Plan

Introduction

This chapter is about the tools and technologies we used in this project.

2.1 Tools and technologies

2.1.1 Technologies

☛ **Raspberry Pi**

The Raspberry Pi is a credit-card sized computer designed and manufactured by the Raspberry Pi Foundation. It comes with a set of open source technologies, communication and multimedia web technologies. We used this card to make programs that detect bottles into a vending maching. It is the main component of our project.

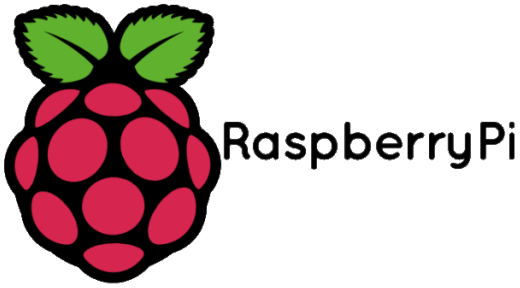

Figure 2.1: Raspberry logo

☛ **Jupyter**

Jupyter is a web application used to program in more than 40 programming languages, including Python, Julia, Ruby, R, or Scala2. Jupyter is an evolution of the IPython project. Jupyter allows you to make notebooks or notebooks, that is to say programs containing both text in markdown and code in Julia, Python, R ... These notebooks are used in data science to explore and analyze Data.

Figure 2.2: JupyterLab logo

☛ **Google Colab**

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by team members - just the way documents are edited in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in the notebook.

— Zero configuration required

— Free access to GPUs (TESLA K80) TPUs

— Easy sharing

Figure 2.3: Google Colab

☛ **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant 18 white space. It provides constructs that enable clear programming on both small and large scales.

Figure 2.4: Python logo

☛ **TensorFlow**

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

Figure 2.5: TenserFlow logo

☛ **Keras**

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

Figure 2.6: Keras logo

2.1.2 Tools

☛ **Raspberry Pi card**

The raspberry pi board comprises a program memory , processor and graphics chip, CPU, GPU, Ethernet port, GPIO pins, Xbee socket, UART, power source connector. And various interfaces for other external devices. It also requires mass storage, for that we use an SD flash memory card. So that raspberry pi board will boot from this SD card similarly as a PC boots up into windows from its hard disk.

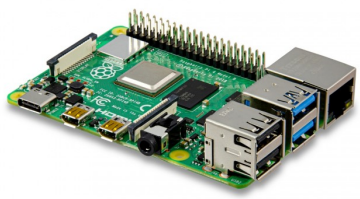

Figure 2.7: Raspberry card

☛ **Camera**

The camera module comprises of a sensor and lens and needs to get instructions from the Pi in order to act as a camera. It has a 5 megapixel sensor, is capable of taking 2592*1944 images and can record 1080p H.264 video at 30 frames per second, allowing us to detect the bottle when inserting.

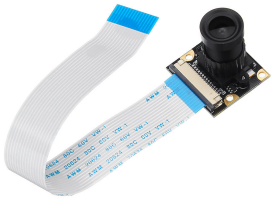

Figure 2.8: Raspberry Camera

☛ **LED**

The White LED is an add on board for the Raspberry Pi which adds visible bright white light for use with the Raspberry Pi camera module. It plugs directly into the I2C interface on the Raspberry Pi, and is completely controllable via the command line, using Python and lots of other programming languages as well.

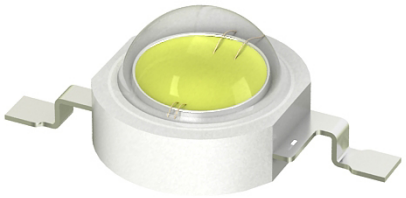

Figure 2.9: Raspberry LED

☛ **Relay module** The Relay is used to control modules with a higher voltage with the Raspberry Pi. The relay “switch” is utilized by means of a low-voltage pulse. Since the Pi only tolerates a maximum of 5V without relays, there is the risk that the Pi could burn out. However, if you have two separate circuits this can not happen.

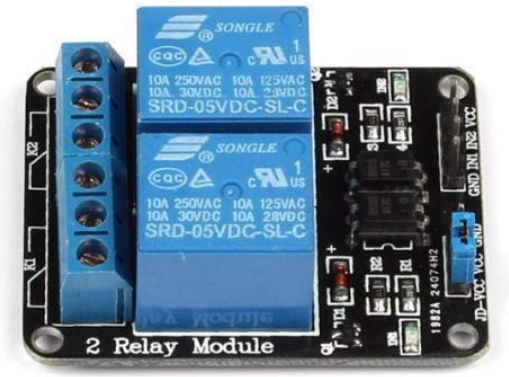

Figure 2.10: Raspberry Relay

2.2 System architecture diagram

This section discusses the system modelling, process and theoretical design of the reverse vending machine. It also describes the design flow which was used to implement the design.

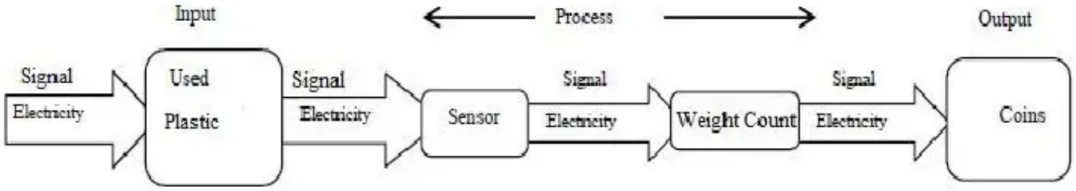

Figure 2.11: Block Diagram of the reverse vending machine

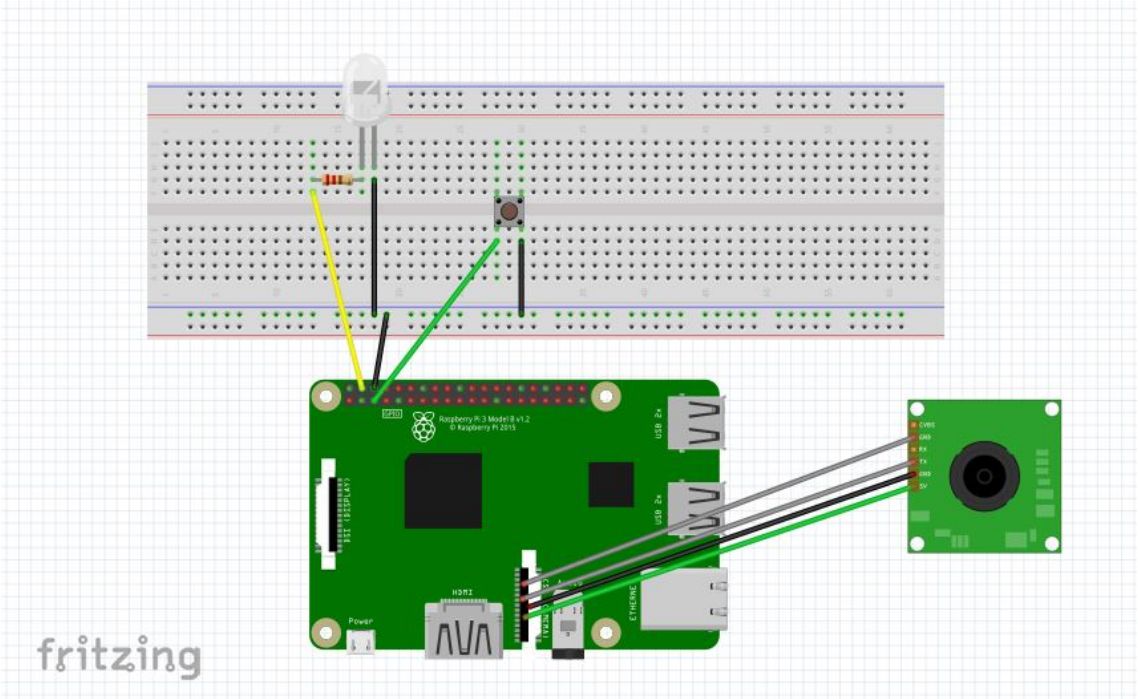

Figure 2.12: RVM architecture with Fritzing

Chapitre 3

Data Collection and preparation

Plan

Introduction

In this chapter, we will move to the Data Collection and preparation by shedding light on the various steps provided to ensure the accomplishment of this project.

3.1 Data requirements

According to our business objectives, there is a specific set of data that need to be collected in order to satisfy the needs of our project.

First of all starting with the ecological goals, we need to get as much insight as possible about the types of materials that are getting bought by consumers, and how their distribution is going to be during the recycling phase. This will give us an idea about the things we can do to prevent pollution, and overload of garbage.

We also want to try to identify the green profiles. These are the profiles that contribute the most to the environment. We can get an idea about their manners, their habits, their consumption profile, and try to encourage the people to have the same the behaviour as them.

Client segmentation has always been a goal of a lot of companies. So in this project we will try to identify the different types of consumers that are available on the market, and try to get a conclusion about the similarities and differences between them.

Position and information about regions is very important and is considered a part of Client Segmentation. Getting an idea about the specific consumption habits of each region is very important because it helps us know where we can put our machines, and what each region has to offer in terms of bottle and can consumption.

3.2 Data Collection

3.2.1 Internal data

For the collection of our internal training data, we tried to collect a sample that represents the types of containers will be used and thrown in the vending machine by tunisian consumers.

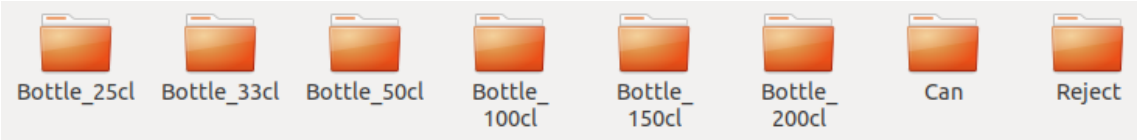

Figure 3.1: Training labels

These are the main classes :

- Plastic Bottle of 25cl
- Plastic Bottle of 50cl
- Plastic Bottle of 100cl
- Plastic Bottle of 150cl
- Plastic Bottle of 200cl
- Aluminium Cans
- Rejects : This class represents the items that are not allowed to be inserted in the RVM (Glass bottles,filled bottles...etc)

Here are a few examples of some of the glasses :

50cl plastic bottle :

Figure 3.2: 50 cl bottle example

150cl plastic bottle :

Figure 3.3: 150 cl bottle example

Aluminum cans :

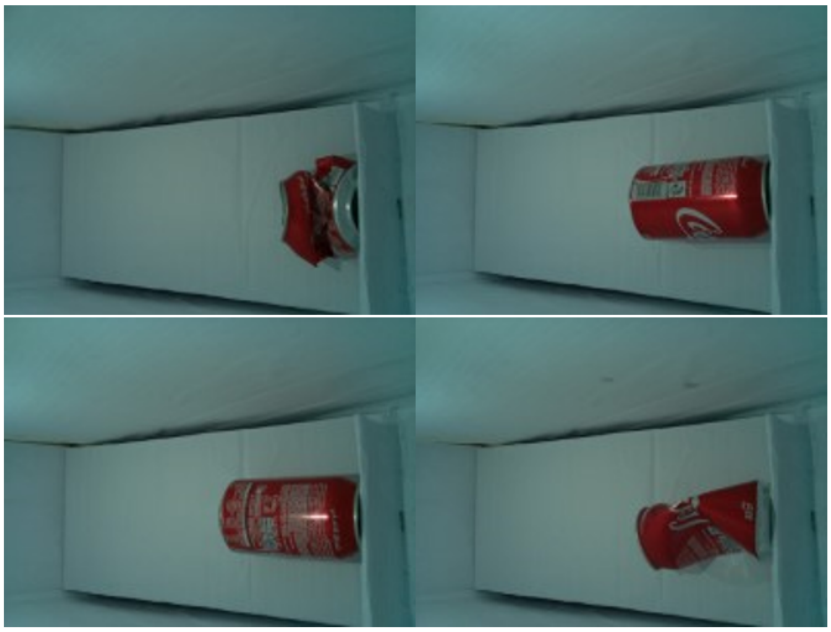

Figure 3.4: Cans example

3.2.2 External data

This is the most important part in our project since this process of gathering and measuring information on targeted variables is used to answer relevant questions and evaluate our outcomes.

We used google Form to collect information from clients by knowing their localisation,gender, profession and their favorite type and brand of soft drink and water.

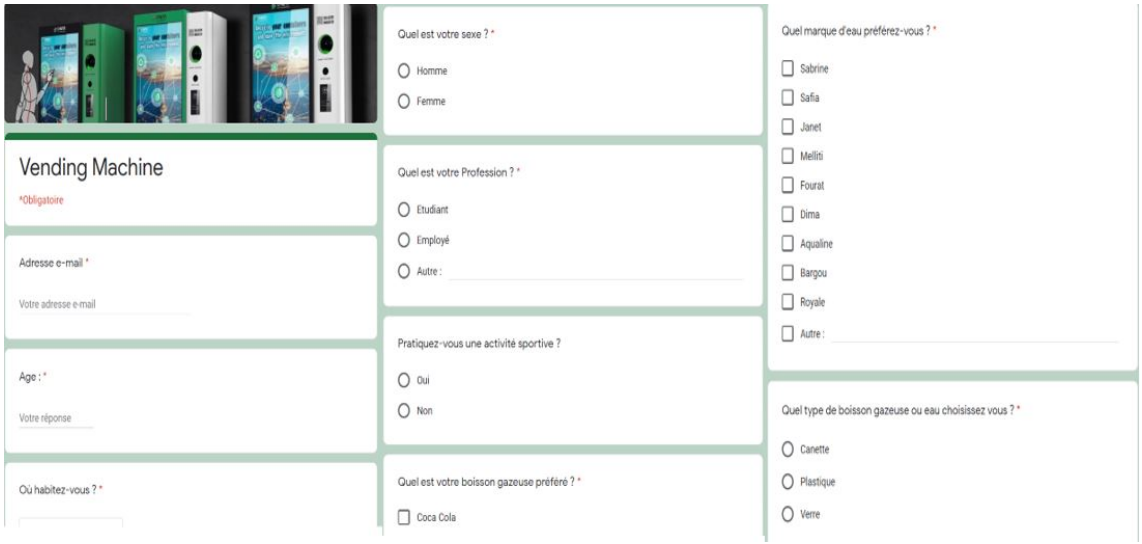

Figure 3.5: Google Form

Also, we collected some data from the National institute of statistics's web site that estimates the population of different region in Tunisia.

Socio économique

Unité; Gouvernorat de L'Ariana

		2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Estimation du nombre de la population dont la	Nombre	37.034,4	37.599,5	37.831,1	36.711,8	37.314,7	38.398,8	43.214,0	43.988,1	45.700,0	..	49,6	..
Estimation du nombre de la population dont la	Nombre	34.453,6	34.850,8	35.076,3	31.576,8	31.429,8	32.044,0	38.442,0	38.099,0	39.590,0	..	41,2	..
Estimation du nombre de la population dont la	Nombre	40.646,4	40.209,8	39.615,8	38.916,3	37.976,5	37.520,5	36.500,0	36.112,1	37.540,0	..	36,2	..
Estimation du nombre de la population dont la	Nombre	46.324,6	46.420,2	46.135,4	53.076,2	53.198,3	53.494,0	46.632,0	47.055,9	48.890,0	..	45,7	..
Estimation du nombre de la population dont la	Nombre	64.405,2	67.600,9	69.996,4	73.852,5	54.393,5	56.115,8	52.349,0	53.652,1	55.720,0	..	54,8	..

Figure 3.6: table of estimated population

3.3 Data Understanding

We collected many reply from different clients around the country and based on their answers we created statistical graphs that helped us in the data preparation.

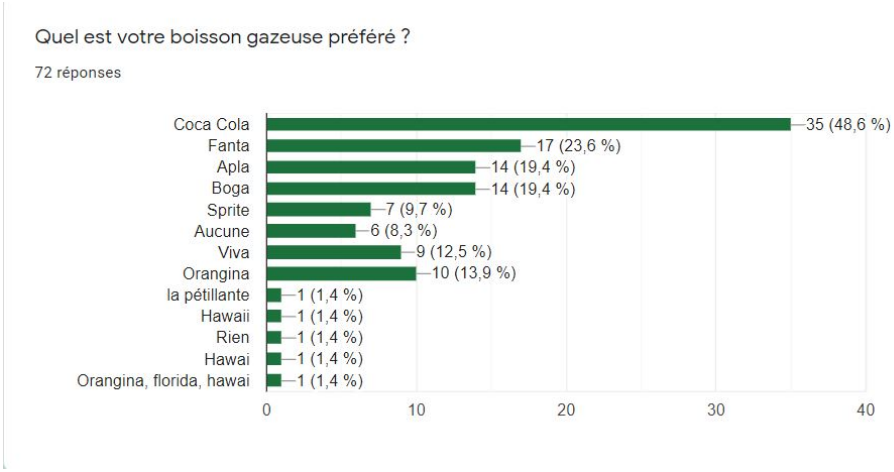

Figure 3.7: Best brand of Soft Drink

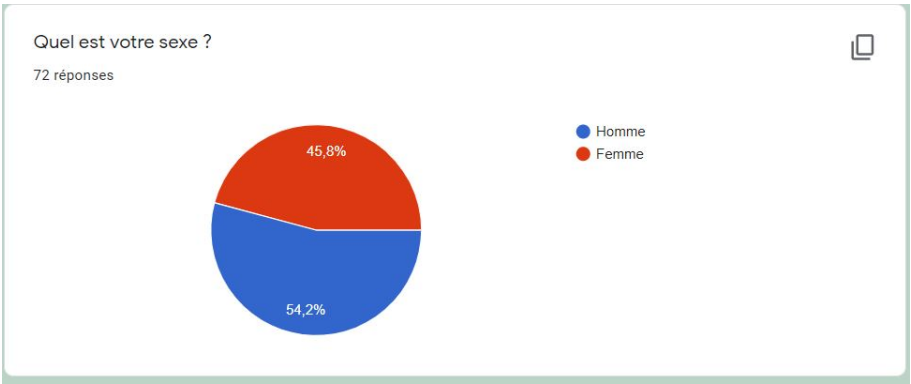

Figure 3.8: Type of Gender

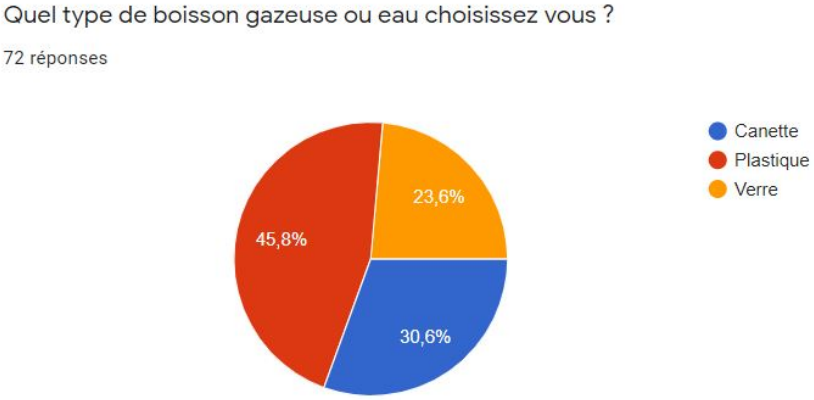

Figure 3.9: Best type of Soft Drink and water

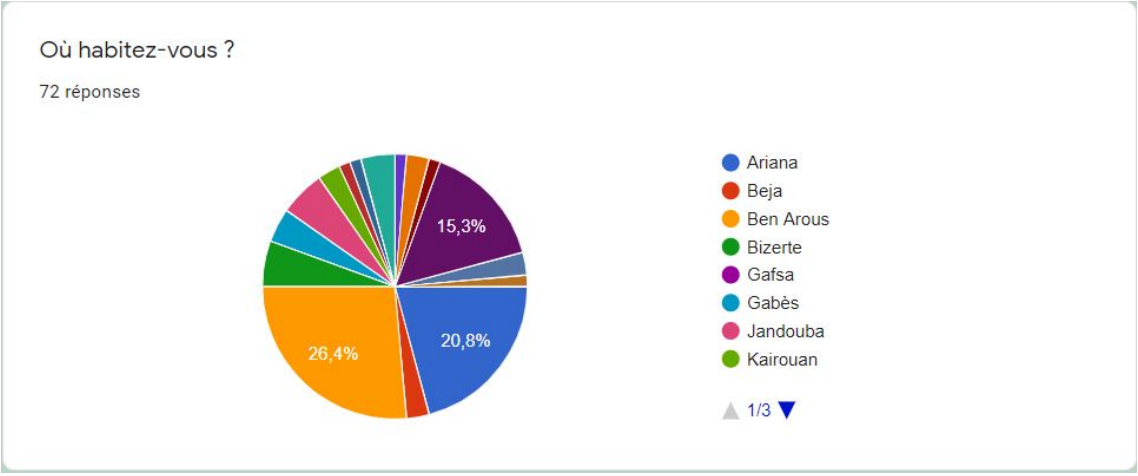

Figure 3.10: statistic of location

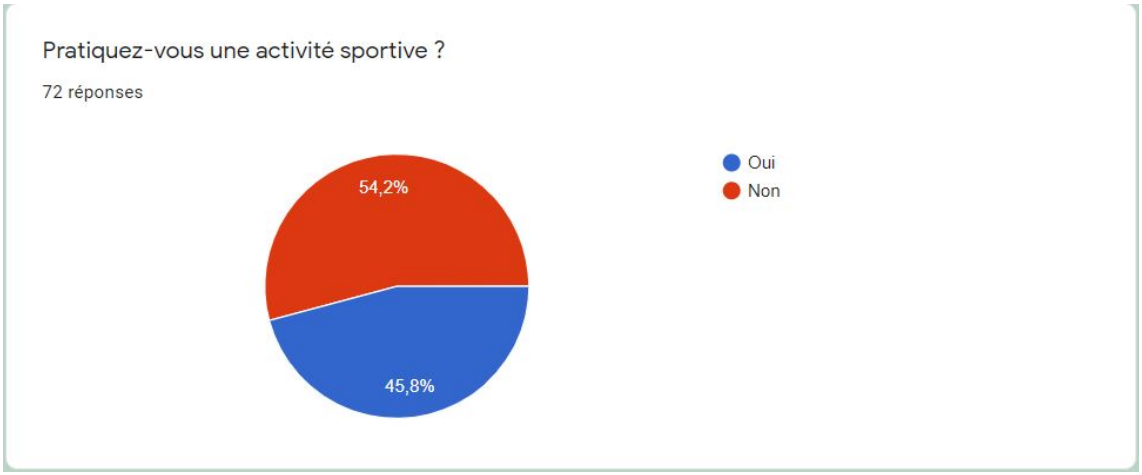

Figure 3.11: statistic of sport

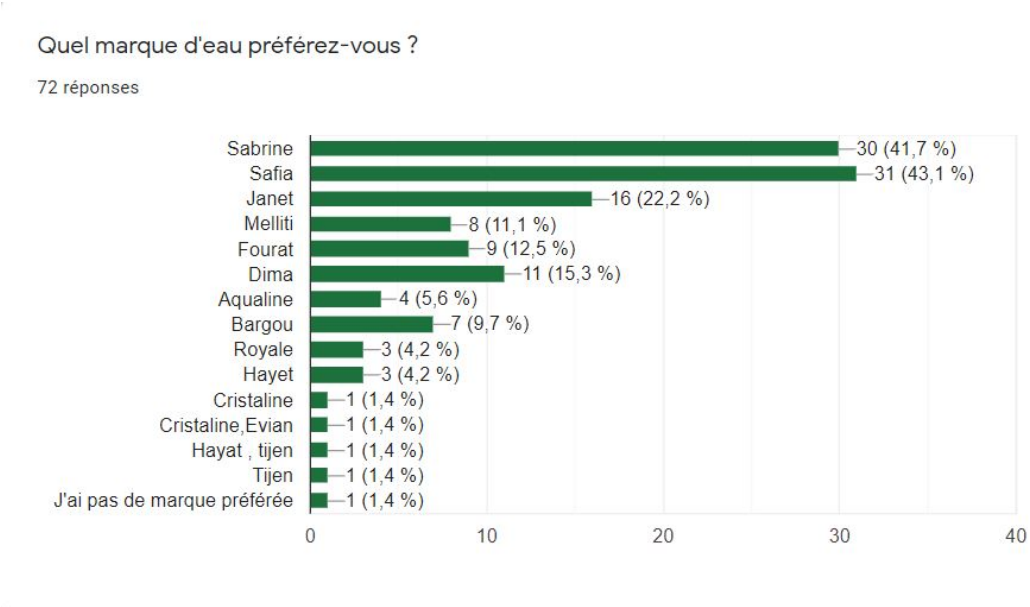

Figure 3.12: Best brand of water

From the survey's answers and the graphs, we can already see that there is a dominant profile. These profile describes the consumers that :

- Drink a lot of Coca-Cola
- Are male
- Prefer plastic containers
- Live in Ariana
- Don't practice any sports
- Drink Safia as mineral water

This will give us a good insight on what to do in the project's next phases.

3.4 Data Preparation

One of the primary purposes of data preparation is to ensure that the information readied for analysis is accurate and consistent, so the results of our project will be valid.

3.4.1 Internal Data

So far during the internal data collection phase, we managed to collect 1075 training images. This can be considered as a low number during the modeling/training phase. So now we are going to rely on some data augmentation techniques. This will allow us to add more variance to our training data in order to have a good generalized model. These are some of the techniques we used :

- Image rotation
- Image zooming
- Skew tilt
- Random distortion
- Random cropping
- Random flipping

This is an example of the changes applied on single image :

Figure 3.13: Data augmentation code

This is the code we used :

```
p.rotate(probability=0.7, max_left_rotation=20, max_right_rotation=20)
p.zoom(probability=0.5, min_factor=0.8, max_factor=1.1)
p.skew_tilt(probability=0.5)
p.random_distortion(probability=0.5, grid_width=4, grid_height=4, magnitude=8)
p.crop_random(probability=0.5, percentage_area=0.5)
p.flip_random(probability=1)
```

```
Initialised with 1075 image(s) found.
Output directory set to /media/moez/Local Disk/ESPRIT/4_Data_Science/Semestre 2/PIDS/Reverse_Vending_Machine/Final_Data/Data_Original/Data/output.
```

```
p.sample(7000)
```

```
Processing <PIL.Image.Image image mode=RGB size=512x384 at 0x7F8B4051E780>:  33%|███       | 2277/7000 [01:52<09:23,  8.38 Samples/s]
```

Figure 3.14: Data augmentation code

3.4.2 External Data

We ensured that the data collected from the form's answers are cleaned in order to have a better vision in the modeling phase.

	Timestamp	Age	Where do you live ?	What is your gender?	Do you practice a sporting activity?	What is your favorite soft drink?	Which brand of water do you prefer?	What type of soft drink or water do you choose?	Email	What is your profession ?
0	27/02/2020 14:14:20	45	Monastir	Male	Yes	Coca Cola	Sabrine	Plastic	not defined	Student
1	27/02/2020 14:14:36	77	Monastir	Male	No	Coca Cola, Apla	Sabrine, Safia	Plastic	not defined	Student
2	27/02/2020 14:31:34	22	Monastir	Female	No	Fanta	Sabrine	Plastic	not defined	Student
3	27/02/2020 14:58:58	23	Ariana	Male	No	Coca Cola	Hayet	Plastic	not defined	Student
4	27/02/2020 17:39:29	22	Gafsa	Female	No	Coca Cola, Fanta	Sabrine, Dima	Plastic	not defined	Student
5	27/02/2020 17:39:56	35	Ben Arous	Female	No	Fanta	Janet	Plastic	not defined	Student
6	27/02/2020 17:40:38	33	Ben Arous	Male	No	Apla, Boga	Sabrine, Melliti, Dima	Plastic	not defined	Student
7	27/02/2020 17:41:08	66	Kébili	Male	Yes	Aucune	Safia	Plastic	not defined	Student
8	28/02/2020 19:51:25	22	Ariana	Male	No	Coca Cola	Melliti, Dima	Plastic	not defined	Student
9	28/02/2020 20:02:37	24	Gabès	Male	Yes	Coca Cola	Sabrine	Can	thabti.montassar@gmail.com	Student
10	28/02/2020 20:04:07	33	Jandouba	Male	Yes	Coca Cola	Sabrine	Glass	mourad.mhiri@gmail.com	Employee

Figure 3.15: vending machine's dataset

```
#On va supprimer les lignes qui contiennent plus  de 6 valeurs non manquantes
external_data=external_data.dropna(thresh=6)
```

```
# Afficher les valeurs uniques pour les question qui peuvent avoir des autres réponses.
external_data["What is your favorite soft drink?"].unique()
external_data["Which brand of water do you prefer?"].unique()
external_data["What is your profession ?"].unique()
```

```
array([nan, 'Etudiant', 'Employé', 'Coach sportif', 'Chômage'],
      dtype=object)
```

```
# Remplacer les réponses nulles par la mediane (pour ne pas perdre l'information)
external_data.fillna(external_data.median(),inplace=True)
```

```
# Remplacer les réponses nulle par la veleur la plus fréquente.
external_data=external_data.apply(lambda x:x.fillna(x.value_counts().index[0]))
```

```
# Vérifier s'il y existe encore des valeurs nulles
external_data.isnull().sum()
```

```
Timestamp                                    0
Age                                          0
Where do you live ?                          0
What is your gender?                         0
Do you practice a sporting activity?         0
What is your favorite soft drink?            0
Which brand of water do you prefer?          0
What type of soft drink or water do you choose?    0
Email                                        0
What is your profession ?                    0
dtype: int64
```

Figure 3.16: cleaning phase

Conclusion

In this second chapter, we analyzed and refined the provided data and our external data in order to have a smart vending machine and choose better future course of actions to maximize customer's satisfaction and increase the turnover.

Chapitre 4

Modeling

Plan

Introduction

This stage is all about building a model that best solves your problem. A model can be a Machine Learning Algorithm that is trained and tested using the data. This stage always begins with a process called Data Splicing, where you split your entire data set into two proportions. One for training the model (training data set) and the other for testing the efficiency of the model (testing data set). This is followed by building the model by using the training data set and finally evaluating the model.

the algorithm builds an "internal representation" in order to be able to perform the task requested of it (prediction, identification, etc.). observation.

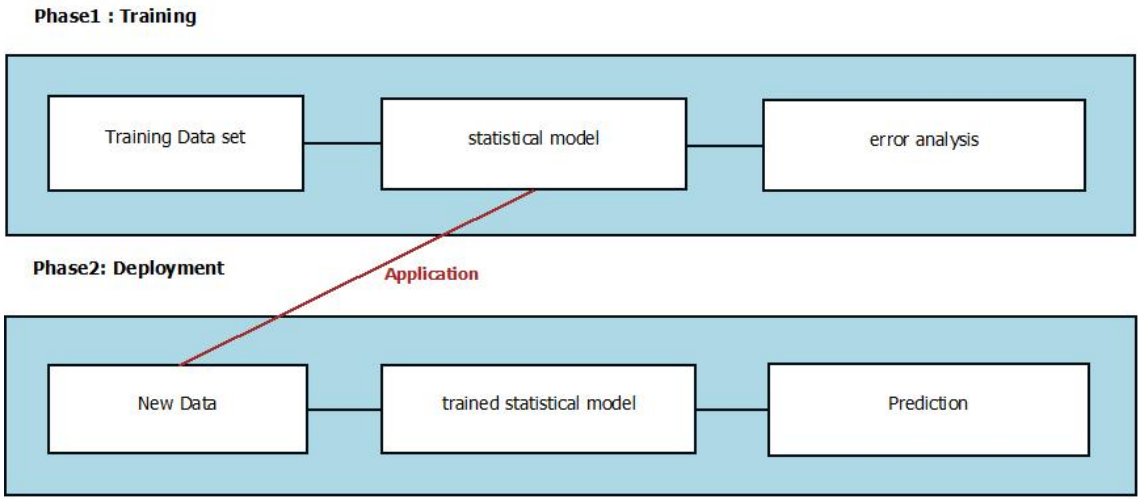

Figure 4.1: Phases of modeling process

In this part, we used the python library "fastai" for computer vision. This library simplifies training fast and accurate neural nets using modern best practices. It's based on research in to deep learning best practices undertaken at fast.ai, including "out of the box" support for vision, text, tabular, and collaborative filtering models.

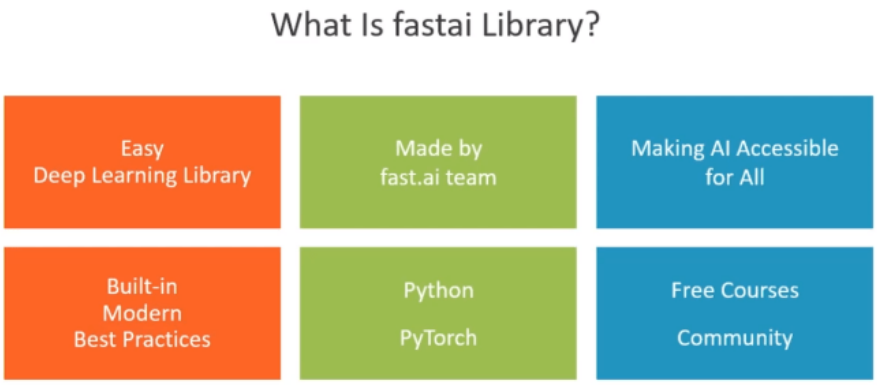

Figure 4.2: Advantages of Fastai library

✪ Use of pretrained Neural networks :

Transfer Learning is a machine learning technique where you use a pre-trained neural network to solve a problem that is similar to the problem the network was originally trained to solve. For example, you could re-purpose a deep learning model built to identify dog breeds to classify dogs and cats, instead of building your own. This could save you the pain of finding an effective neural network architecture, the time you spend on training, the trouble of building a large corpus of training data and guarantee good results. You could spend ages coming up with a fifty layered CNN for perfectly differentiating your cats from your dogs or you could simply re-purpose one of the many pre-trained image classification models available online.

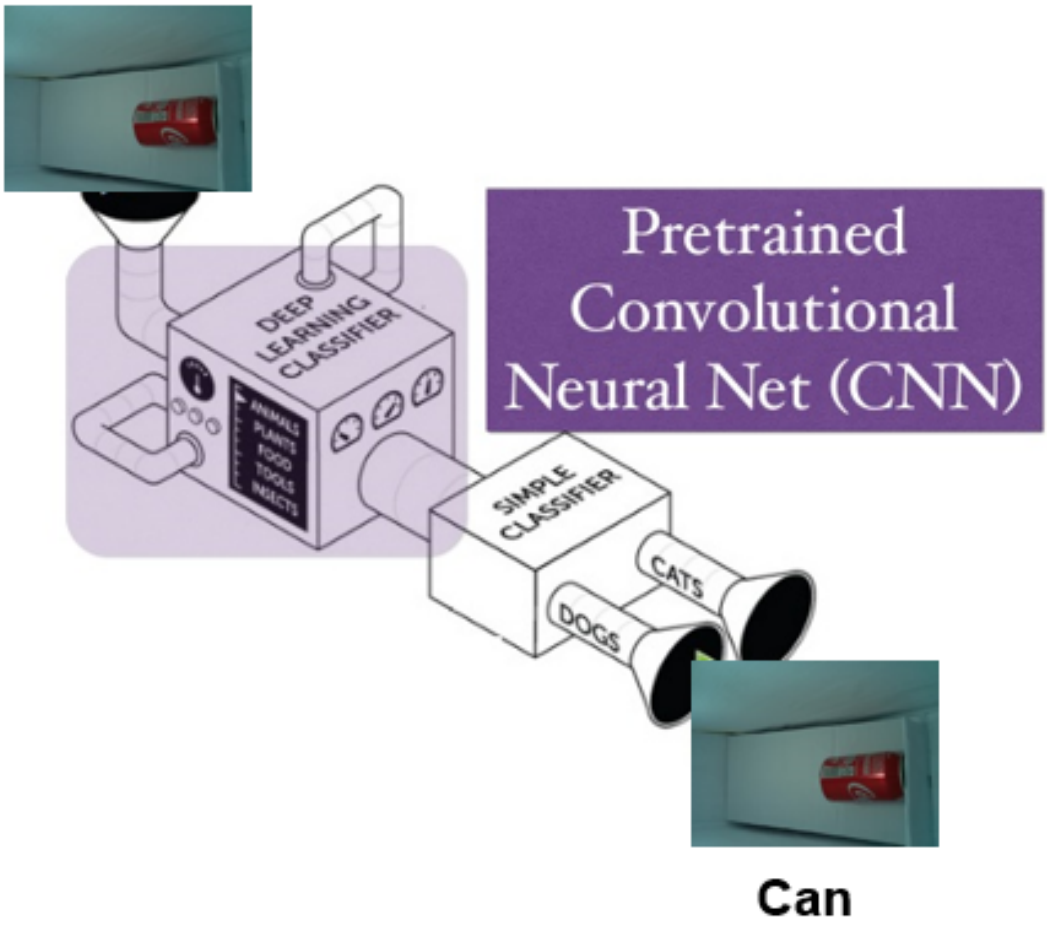

Figure 4.3: Pretrained Convolutional Neural Network

4.1 Initialization :

First, we started off with loading our dataset. Here are some examples of classes that we have :

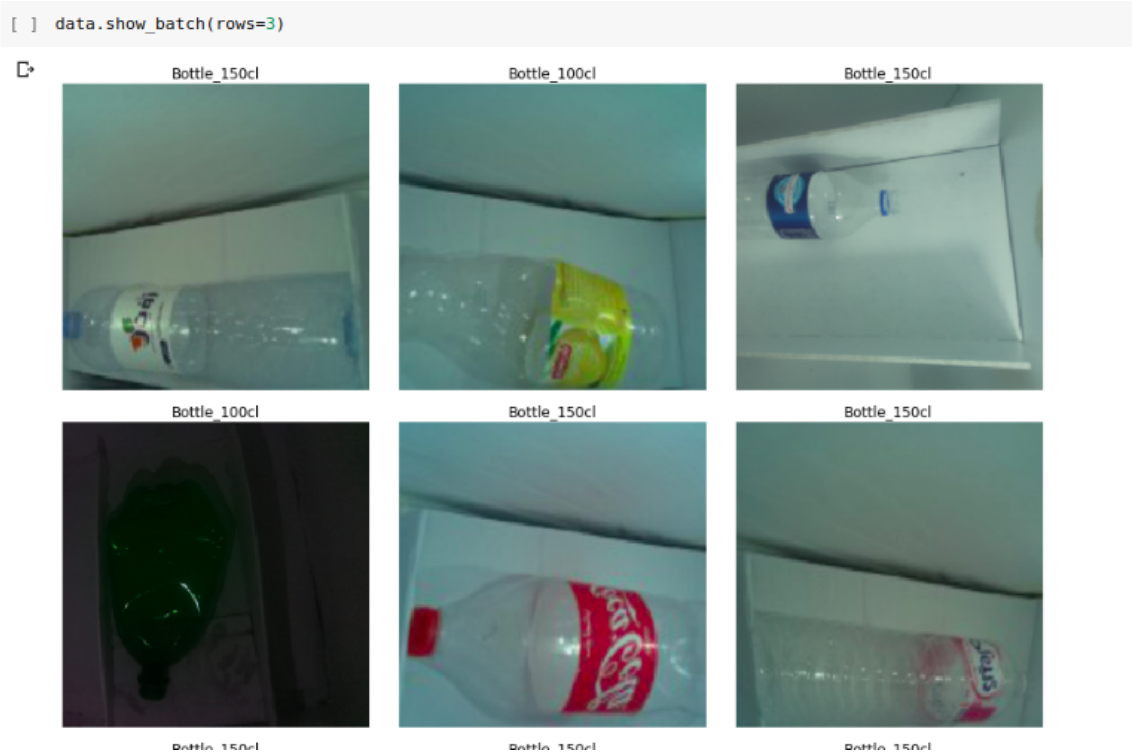

Figure 4.4: examples of classes

Second, we make transformations like rotations, changing lights to increase the number of our inputs. Then we normalized our images to get an image input size 224x224. We choose this size because the pretrained networks that we used in our models must have this shape.

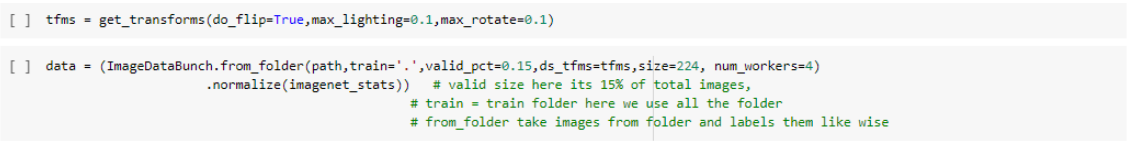

Figure 4.5: Normalized data

Then we tried out different model architectures, more precisely more pre-trained models, which are neural networks in which the hidden layers have already been trained on a huge image dataset. So these networks can use the learned features to make predictions on the bottle images.

✪ Parameters and hyperparameters

- Learning rate :

  We first started off by finding the best learning rate for our dataset :

Figure 4.6: Learning Rate

  According to this graph , the optimal learning rate that gave us the weakest loss is le-1.

- Epochs :

  An epoch is a hyper-parameter which is defined before training a model. One epoch is when an entire data set is passed both forward and backward through the neural network only once.

- Optimizers :

  Optimizers update the weight parameters to minimize the loss function. Loss function acts as guides to the terrain telling optimizer if it is moving in the right direction to reach the bottom of the valley, the global minimum.

  - SGD : Stochastic Gradient Descent is one of the simplest optimization algorithms. It uses just one static learning rate for all parameters during the entire training phase.

  - AdaGrad : It is very similar to SGD. The key difference in design is that AdaGrad uses Adaptive gradients — it has a different learning rate for every single parameter in the neural network.

  - Adam : Adaptive moment estimation uses past learning rates like AdaGrad do. However, Adam doesn't stop there, it also uses past gradients to speed up learning.

✪ Metrics

- Accuracy :

  It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{numberofcorrectpredictions}{totalnumberofpredictionsmade}$$

- FBeta score :

  The F-beta score is the weighted harmonic mean of precision and recall, reaching its optimal value at 1 and its worst value at 0. We are using fbeta macro average in case some class of birds have less train images.

- Error rate :

  The inaccuracy of predicted output values is termed the error of the method. This is the proportion of cases where the prediction is wrong.

Then we tried out some different models on the data. Below we can see the different accuracies for each model during different training epochs.

4.2 Models :

4.2.1 ResNet18 (Residual neural network) :

ResNet18 is a convolutional neural network that is 18 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as bottle, keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

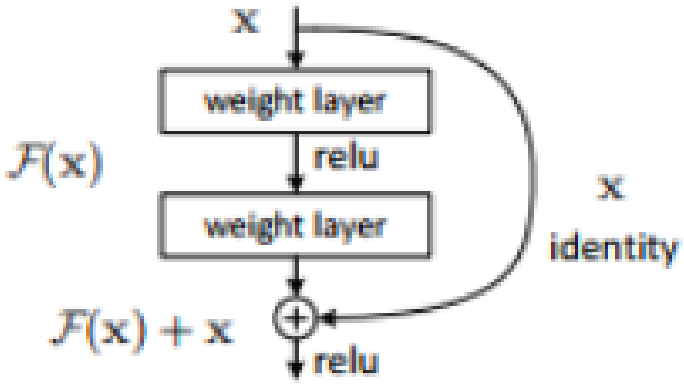

Figure 4.7: Logical scheme of base building block

- We start with ResNet18 with 10 epochs :

```
[ ]  lr = 1e-2 # learning rate
     learn.fit_one_cycle(10,lr,moms=(0.8,0.7))  # moms
```

epoch	train_loss	valid_loss	error_rate	f_beta	accuracy	time
0	2.459060	1.017971	0.304348	0.484071	0.695652	00:19
1	1.635923	1.194261	0.267081	0.492495	0.732919	00:20
2	1.208461	0.921845	0.229814	0.576269	0.770186	00:19
3	0.965122	0.802075	0.192547	0.539985	0.807453	00:19
4	0.784886	0.321595	0.118012	0.680907	0.881988	00:19
5	0.652753	0.301590	0.074534	0.769714	0.925466	00:19
6	0.536414	0.224569	0.080745	0.777957	0.919255	00:19
7	0.426982	0.182912	0.043478	0.791464	0.956522	00:19
8	0.354107	0.173844	0.049689	0.813603	0.950311	00:20
9	0.295119	0.170915	0.049689	0.813603	0.950311	00:19

Figure 4.8: Resnet18

- For the predictions :

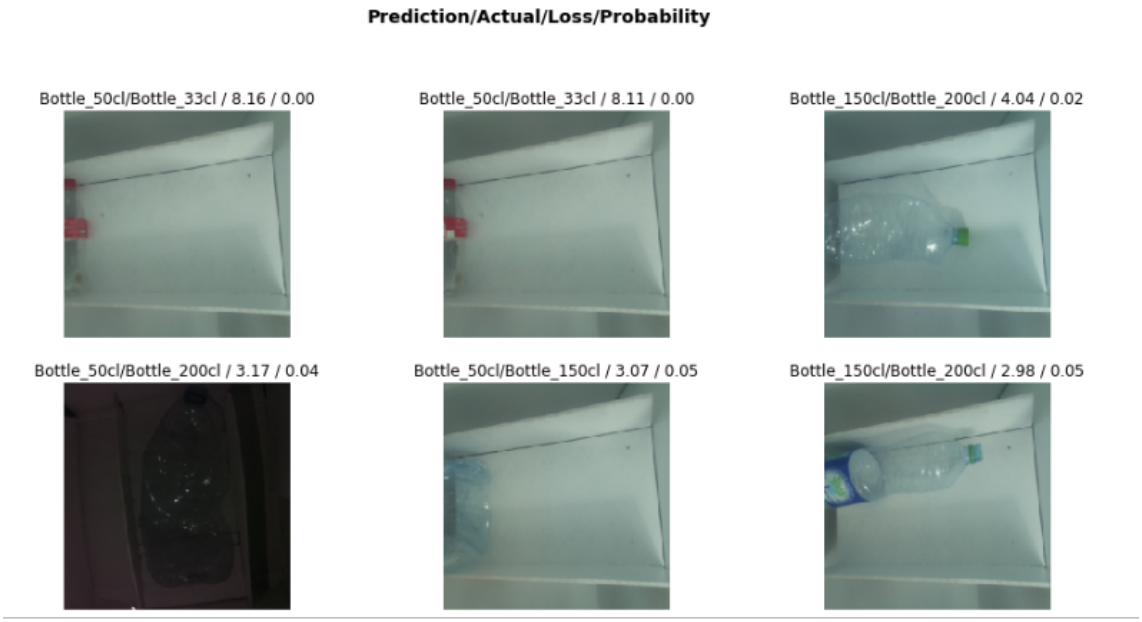

Figure 4.9: Predictions of ResNet18 with 10 epochs

- Like all models, it has errors. Here is the most confused images.

```
[ ]  interp.most_confused(min_val=2)
```

```
[('Bottle_200cl', 'Bottle_150cl', 4),
 ('Bottle_150cl', 'Bottle_50cl', 3),
 ('Bottle_33cl', 'Bottle_50cl', 2),
 ('Can', 'Bottle_50cl', 2)]
```

Figure 4.10: Most confused categories

For example, our model confuses between Bottle33cl and Bottle50 cl twice, also it assigns the category Bottle150cl to Bottle200cl four times.

4.2.2 ResNet101 :

ResNet101 is a convolutional neural network that is 101 layers deep. It's like ResNet18 but with different hidden layers. Its main use is in Image Recognition. The more hidden layers we have, the more chance we have to optimize our model.

```
[ ] learn = cnn_learner(data, models.resnet101, metrics=[error_rate,fb, accuracy],model_dir='./working')
    lr = 1e-2 # learning rate
    learn.fit_one_cycle(10,lr,moms=(0.8,0.7))  # moms
```

```
Downloading: "https://download.pytorch.org/models/resnet101-5d3b4d8f.pth" to /root/.cache/torch/checkpoints/resnet101-5d3b4d8f.pth
100% 170M/170M [00:29<00:00, 6.08MB/s]
```

epoch	train_loss	valid_loss	error_rate	f_beta	accuracy	time
0	1.768965	1.206343	0.229814	0.620358	0.770186	00:25
1	1.150995	1.322172	0.229814	0.685332	0.770186	00:23
2	0.845674	1.366473	0.223602	0.613312	0.776398	00:23
3	0.720732	0.776251	0.260870	0.662004	0.739130	00:23
4	0.619815	0.473714	0.130435	0.861599	0.869565	00:23
5	0.499802	0.429531	0.086957	0.855649	0.913043	00:23
6	0.391335	0.340279	0.080745	0.885405	0.919255	00:23
7	0.313563	0.283181	0.062112	0.908908	0.937888	00:23
8	0.250848	0.335731	0.062112	0.897340	0.937888	00:23
9	0.193131	0.321232	0.062112	0.890108	0.937888	00:23

Figure 4.11: Resnet101

4.2.3 AlexNet :

AlexNet is a convolutional neural network (CNN), designed by Alex Krizhevsky.
It contained eight layers ; the first five were convolutional layers, some of them followed by max-pooling layers, and the last three were fully connected layers. It used the non-saturating ReLU activation function, which showed improved training performance over tanh and sigmoid.

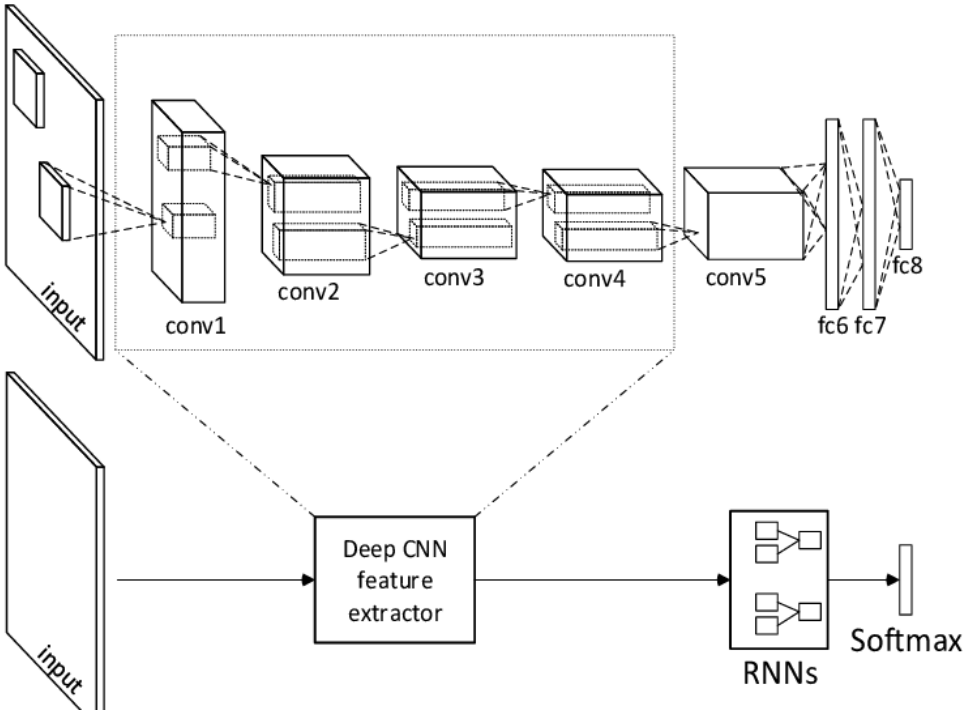

Figure 4.12: AlexNet Structure

▾ Alexnet

```
[ ] learn = cnn_learner(data, models.alexnet, metrics=[error_rate,fb, accuracy],model_dir='./working')
    lr = 1e-2 # learning rate
    learn.fit_one_cycle(10,lr,moms=(0.8,0.7))  # moms
```

```
Downloading: "https://download.pytorch.org/models/alexnet-owt-4df8aa71.pth" to /root/.cache/torch/ch
100% 233M/233M [00:10<00:00, 22.3MB/s]
```

epoch	train_loss	valid_loss	error_rate	f_beta	accuracy	time
0	2.472421	1.252826	0.335404	0.659168	0.664596	00:22
1	1.624842	0.770771	0.211180	0.581648	0.788820	00:21
2	1.214424	0.696330	0.161491	0.781141	0.838509	00:21
3	1.034228	0.570306	0.118012	0.666673	0.881988	00:21
4	0.830325	0.507688	0.161491	0.773901	0.838509	00:21
5	0.722491	0.362726	0.099379	0.804354	0.900621	00:21
6	0.621653	0.407559	0.111801	0.863525	0.888199	00:21
7	0.541612	0.195200	0.055901	0.904928	0.944099	00:21
8	0.479365	0.206866	0.055901	0.904503	0.944099	00:21
9	0.424868	0.194456	0.049689	0.909520	0.950311	00:21

Code Tex

Figure 4.13: Alexnet

Conclusion

The best Score we have is 95% , which was evaluated on the test set, which is the most accurate representation of the performance of our neural network.

But for speed, performance, size reasons, and sinde our model will be loaded on a RasperryPi with limited resources, we will be using the Resnet18 architecture with a top accuracy of 95.56522

Chapitre 5

Deployment

Plan

Introduction

In this chapter we will start the deployment phase. We will create a simple and ergonomic web and mobile application to allow our administrator,first to see some statistics about our Reverse Vending machine and secondly to allow our customers to quickly access their personal spaces.

5.1 Features and users

The system must necessarily be scalable, operational and offer all the necessary information in real time. The functionalities are classified in two categories according to the type of user :

- **the administrator** who has the possibility to consult
  - the total number of tips placed in the RVM (according to a given period, by location of the machine)
  - a detailed dashboard containing the statistics
- **the customer** who can consult
  - the type of bottle used
  - his member space
  - the number of points earned
  - the location of the nearest RVM
  - a mini-shop to order some products online with its points

5.2 Tools and technologies

5.2.1 Python

Figure 5.1: Logo Python

5.2.2 PyCharm

Figure 5.2: Logo PyCharm

5.2.3 Django

Figure 5.3: Logo Django

5.3 Screenshot

5.3.1 Admin

In order to connect to the admin dashboard, we created an interface that will allow us to handle different tasks and visualize different dashboards. First, the admin put his credentials (mail and password) in the specific boxes so he can access to the interface. If he enters a wrong credential , an error message is displayed “bad credentials.” The admin can also click on “forgot the password ? ” and an automatic mail will be sent to him in order to generate a new password.

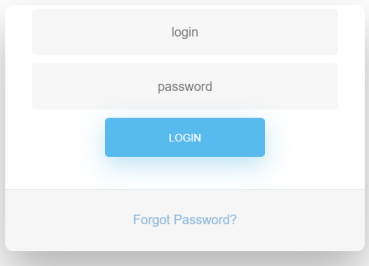

Figure 5.4: Login Admin

After entering the interface, many actions are allowed to be done : The sidebar “Statistics” contain

★ the total of bottles putted inside the vending machine classified by type

★ A curve of the total amount of bottle per month

★ A pie chart of the most putted type of bottle in the vending machine

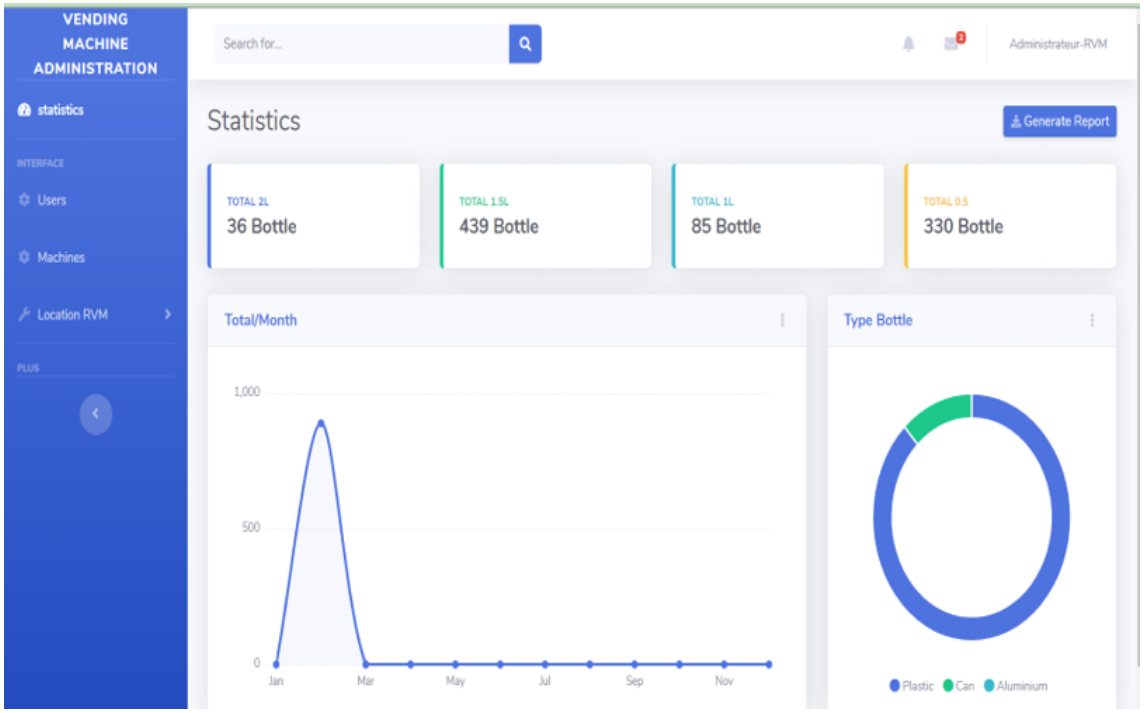

Figure 5.5: Stats Admin

In this area, the admin can consult the client’s suggestions or problems and respond to their claims.

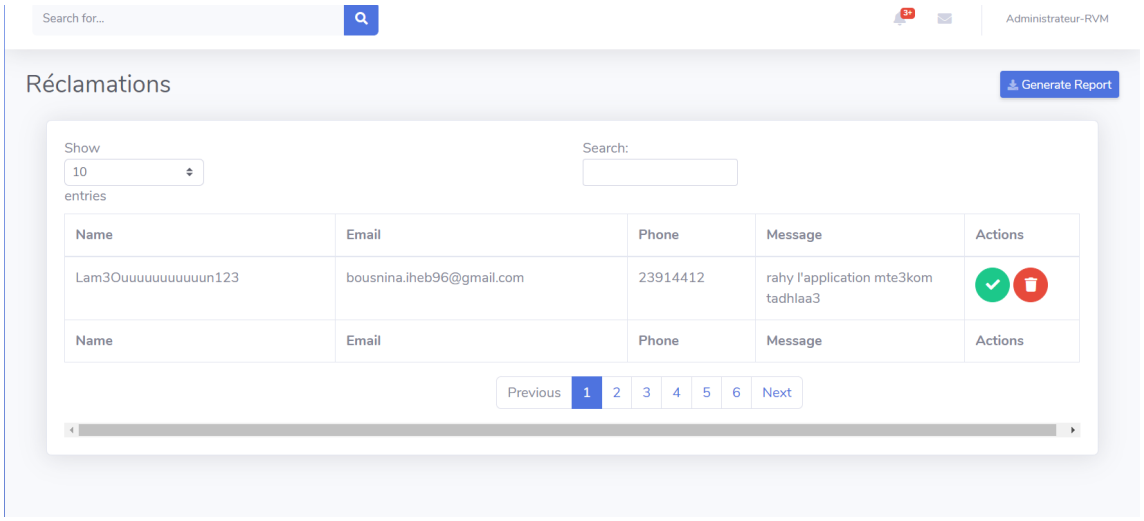

Figure 5.6: Problems interface

This section shows the different clients subscribed to the vending machine program based on

their information (e.g. : location, office, age, points. . .)

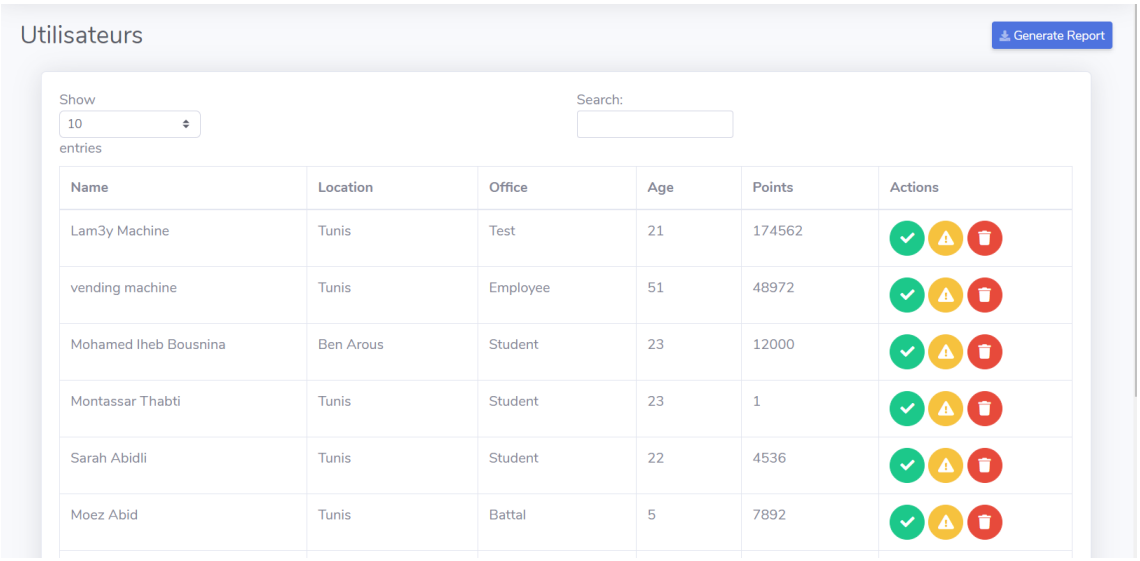

Figure 5.7: Users interface

5.3.2 User

We created an easy-simple interface for clients in order to attract them.

In the welcome interface of our reverse vending machine's website when the client click on the button "Scan now", the webcam will open and detect his face :

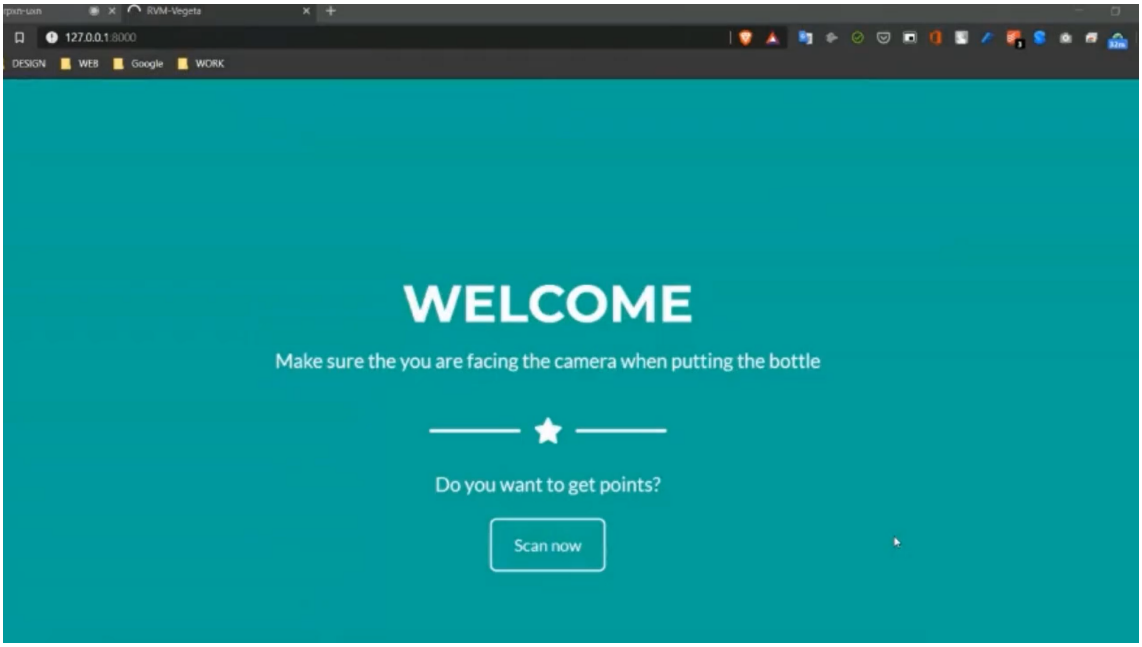

Figure 5.8: The welcome interface of the face recognition

- If the costumer have an existing account :

  The system will recognize him and detect the number and the type of the bottle he putted in the vending machine. The client can also see the number of points he gained plus many other informations by acceding to his personal space .

- If the client don't have an account :

  The system will propose him to create a new account.

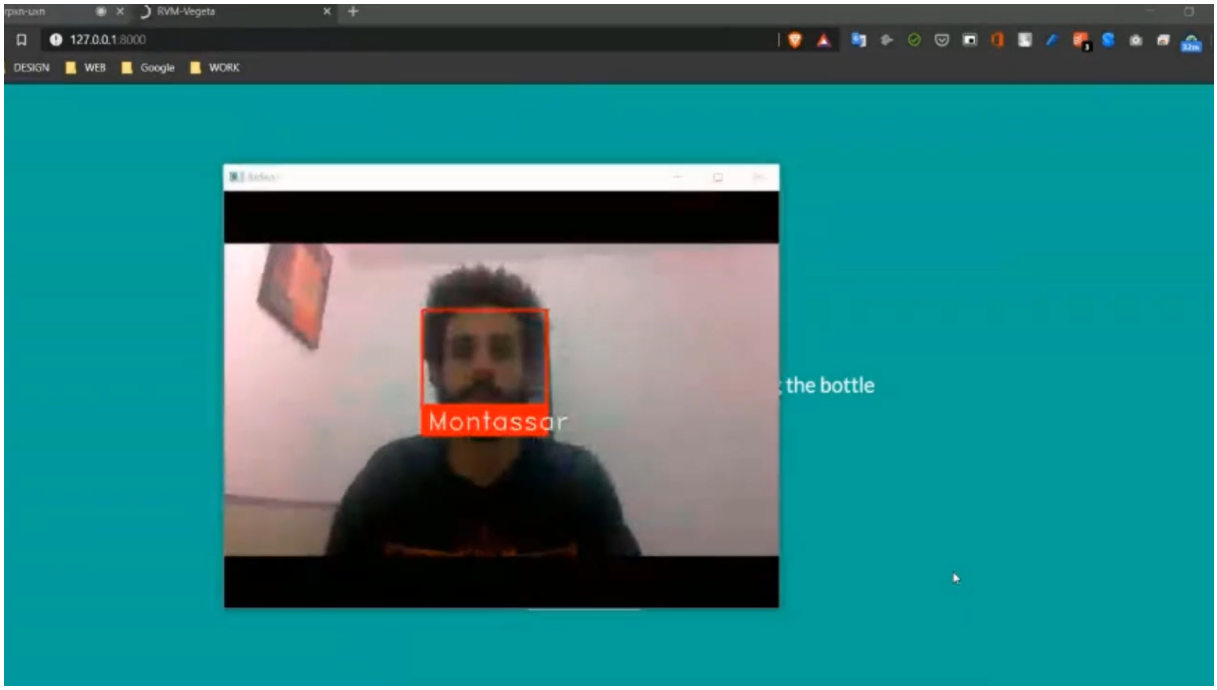

Figure 5.9: Face recognition program

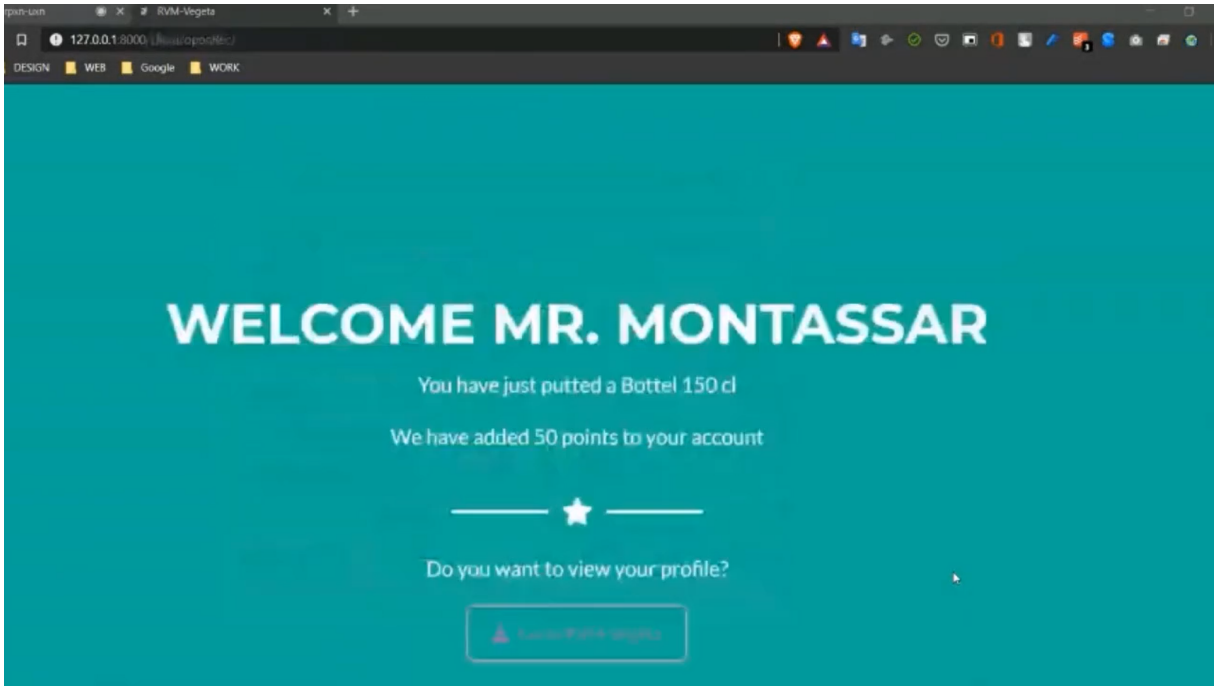

Figure 5.10: Face recognition program

First , the consultor of the Web site can create his account by entering the section "Registration" and fill the form. Then he put his credential in order to access to the interface.

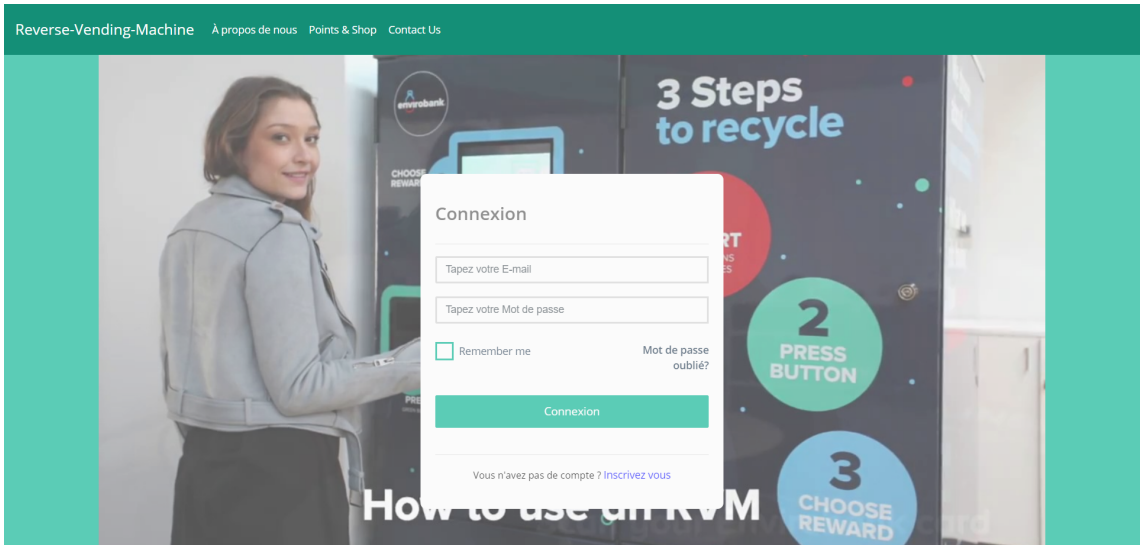

Figure 5.11: Login User

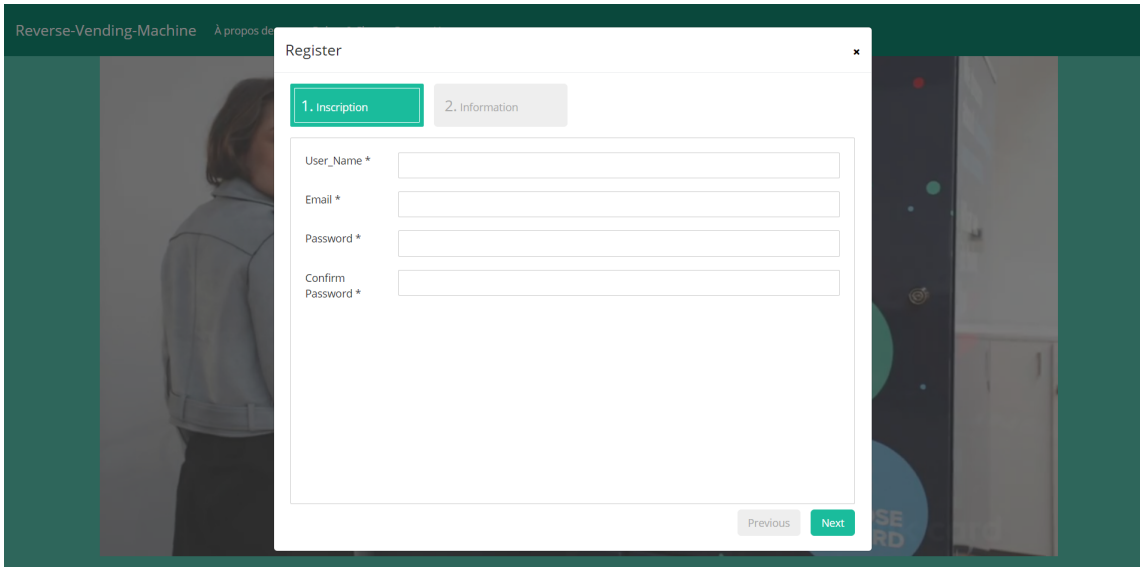

Figure 5.12: Register user step 1

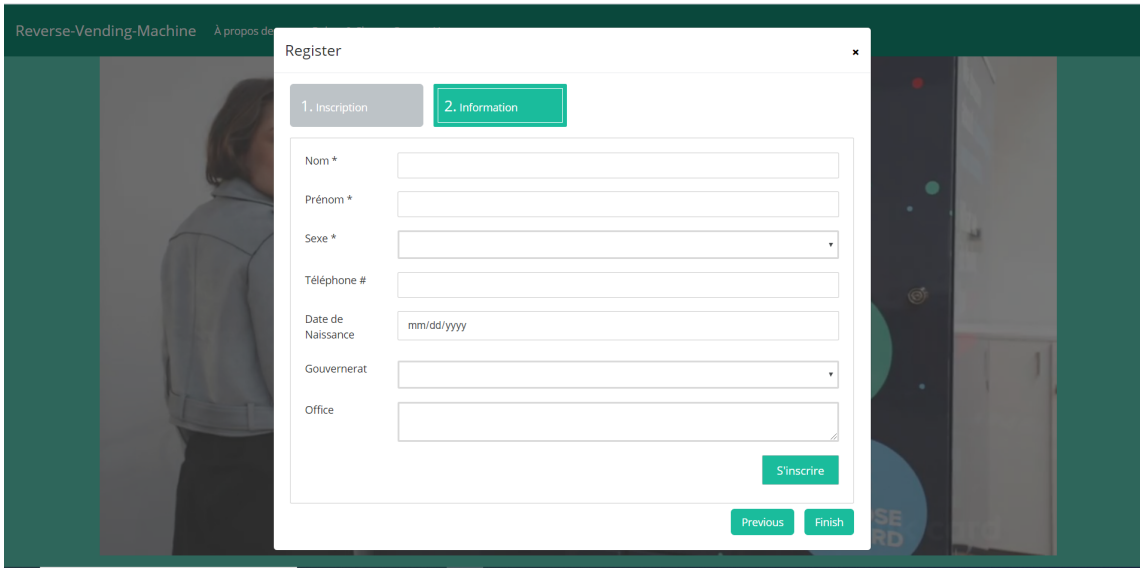

Figure 5.13: Register user step 2

This is the first page that will appear after login in. The name and picture of the client are shown at the first place. The button “Message” allowed him to interact with the admin and other sections such as “Contact us,” “About us” and “Points and shop.”

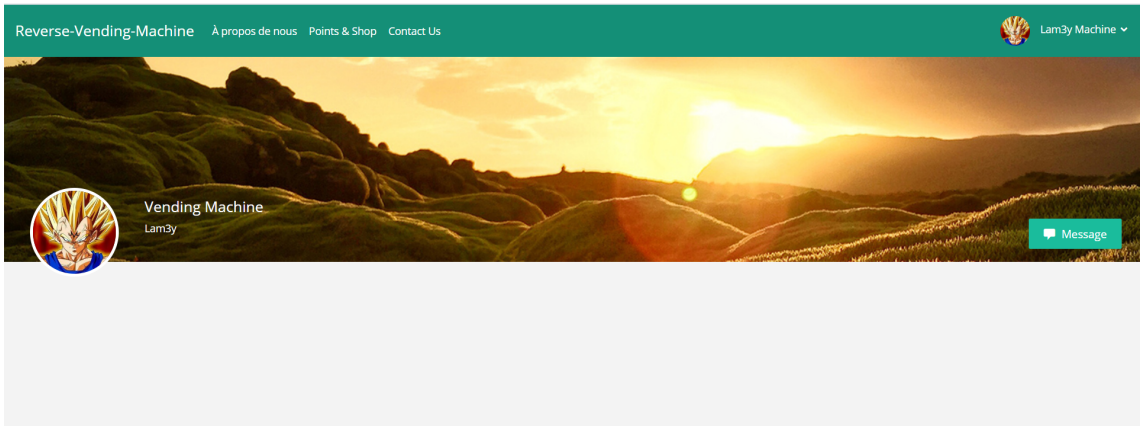

Figure 5.14: User Profile

In this page, the client can consult his points (gained by the number of times he putted bottles in the vending machine) and transform these points into phone recharges cards, charger pass-navigo or cinema tickets.

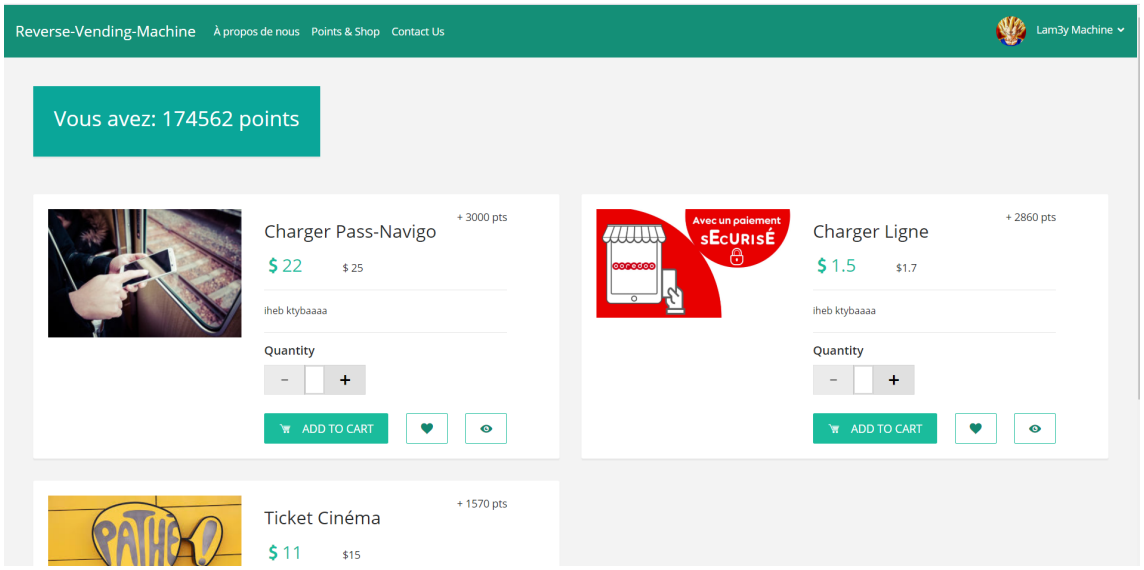

Figure 5.15: Shop

This section is where the client can send his complaint or suggestion to the admin.

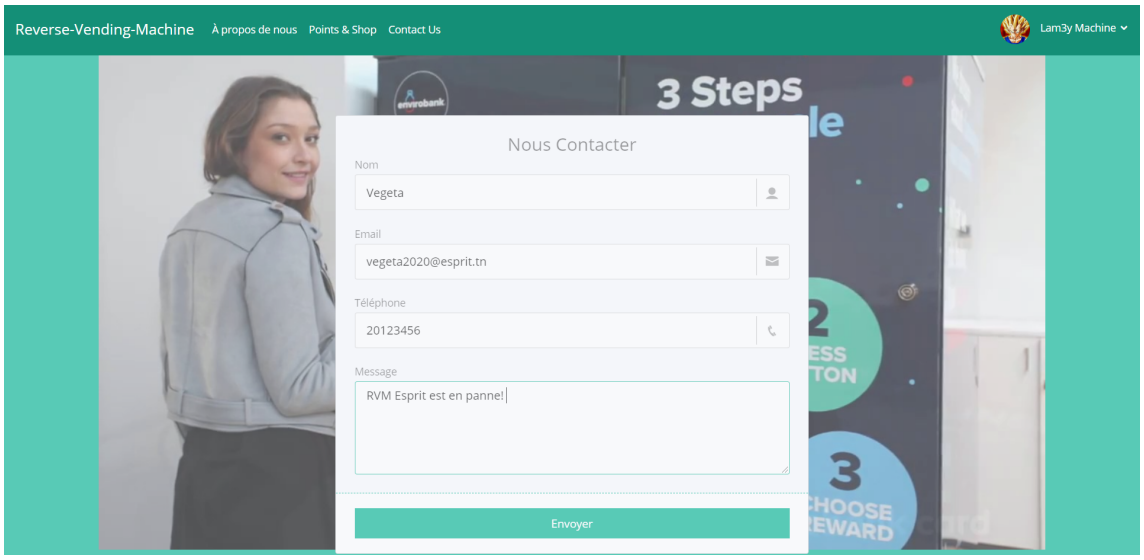

Figure 5.16: Suggestion

Conslusion

In this last phase we tried to create a fluid application containing all the information possible to our admins and facilitate access to users of our machines.

General Conclusion

Usage of Reverse Vending Machines in cities is a perfect way to solve various problems and gain substantial benefits. The common point is that improving the ecology is a primary goal for the society today. The recycling of waste contributes greatly to the cause, and, therefore, implementing effective garbage collection systems can make an important positive impact on the ecological situation in the cities. Apart from the overall cleanness of the city, it is also important to point out the reward-based system for the citizens and the public image benefits from implementing the systems. Last, but not least, the economical viability of these activities. The income created by selling the collected empty beverage containers for further recycling is capable of covering all the costs of implementation and creating of a sustainable profit flow.

References

Bibliography

https ://www.acorecycling.com/b-1-smart-reverse-vending-machine

https ://www.snapmunk.com/reverse-vending-machine-recycling

https ://www.raspberrypi.org/documentation/configuration/raspi-config.md

https ://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

https ://docs.fast.ai/

https ://ch.mathworks.com/help/deeplearning/ref/resnet18.html

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