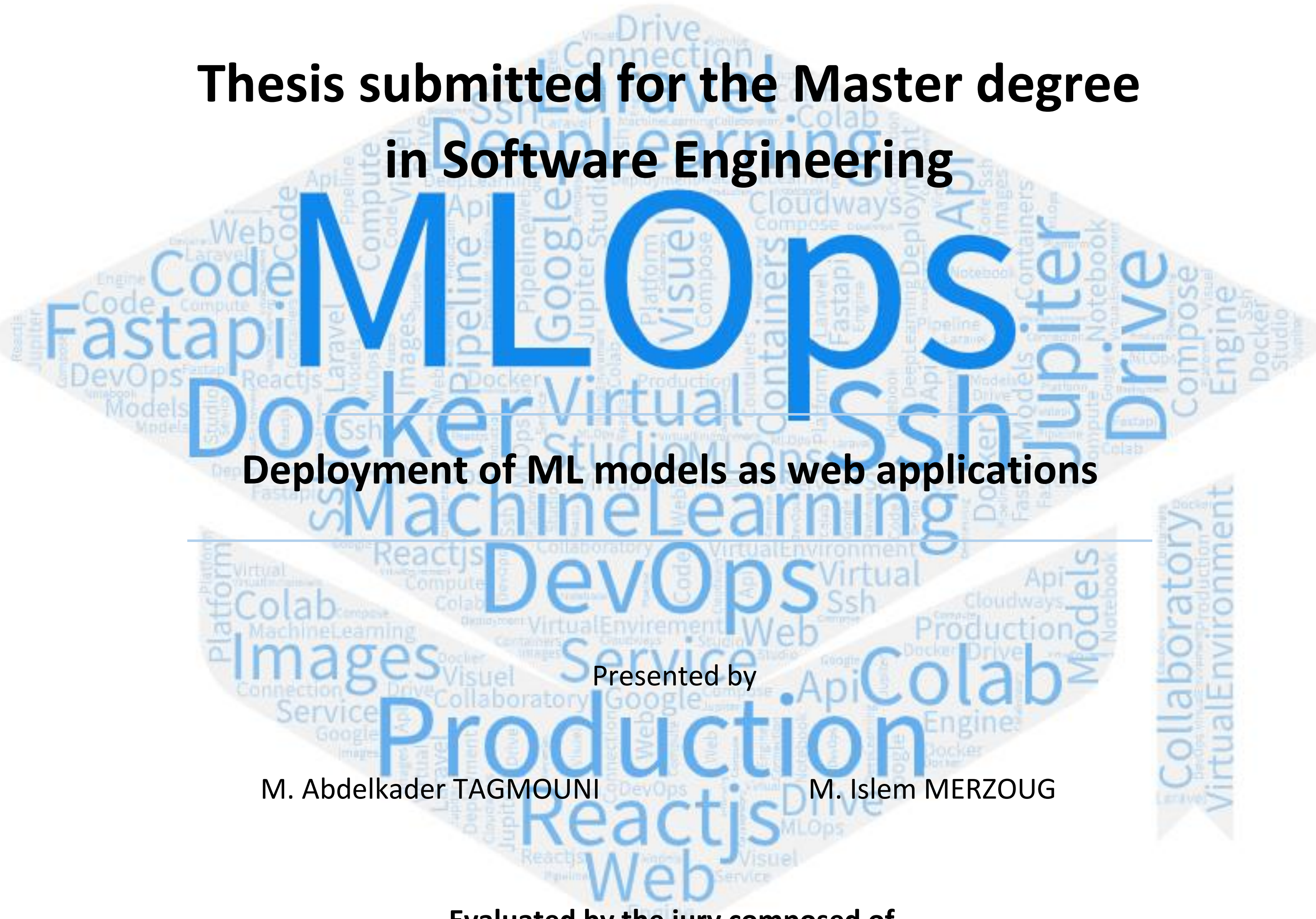
Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University A. Mira of Bejaia
Faculty of Exact Sciences
Computer science Department



Thesis submitted for the Master degree in Software Engineering

Deployment of ML models as web applications

Presented by

M. Abdelkader TAGMOUNI M. Islem MERZOUG

Evaluated by the jury composed of

Supervisor : M. A. BELAID

Member : M. A. AKILAL

Member : Ms. H. BRAHAMI EL BOUHISSI

2020 - 2021

Acknowledgements

First and foremost, we thank Allah for assisting us in completing this project and for being with us throughout our lives.

We would like to express our gratitude to our mentor, Dr. BELAID Ahror, for providing us with an amazing opportunity to carry out this research and gain valuable information and expertise in the field of MLOps. His patience, drive, and vast knowledge constantly lead us in the right direction.

We also appreciate the jury members, Dr. AKILAL Abdellah and Dr. BRAHAMI El BOUHISSI Houda, for taking the time to assess our work.

We'd also like to express our gratitude to all of our instructors and colleagues at the University of Bejaia.

Thank you to all of our friends who have consistently pushed us ahead.

Last but not least, we want to express our gratitude to our parents and families for always being there for us and for supporting in such difficult moment.

Table of contents

Table of figures

Introduction:

Nowadays, Machine learning (ML) has become essential in our daily lives due to the nearly limitless quantity of available data, affordable data storage, and the growth of less expensive and more powerful processing, which permitted its prosperity. As we can see, it has become one of the most influential and powerful technologies globally, but we are still far from reaching its ultimate consequences.

Most of us are unaware that we are using it every day and at every moment. Still, with a simple click on the smartphone to do research on google or by scrolling on YouTube suggested videos, we call machine learning services.

But before we can take full advantage of its benefits, a big job has been done, from the modelization to the coding and finally the deployment. This last one is the most crucial phase because everything remains just ink on a sheet without it, and these different steps are called Machine Learning operations (MLOps) process.

The MLOps is a new technology, appeared for the first time on 2015. It combines the long-established practice of DevOps with the emerging field of Machine Learning; therefore, the deployment of the ML models is still complex as it is still a new field.

The process of deployment into production is as complicated as it is crucial. It requires skills in both software development and machine learning skills. Because during the deployment, you may have to go back to the top levels of the MLOps life cycle to modify the data or train the model with a new data dictionary to adapt the project to the deployment environments.

This thesis defines the MLOps and its challenges and describes the MLOps lifecycle in the first chapter. The second chapter, consists of the modelling and development details then it will be followed by a third chapter that gives an idea about the used tools during the development and the deployment processes and explains their advantages. Finally, it ends with a fourth chapter which describes the different deployed models and their architectures.

Chapter 1

Generalities about Machine Learning Operations

1.1- Chapter introduction:

In this chapter, we present the field of machine learning and its actual use cases, then briefly explain the machine learning operations (MLOps) and their challenges. To finish, we describe the MLOps systems life cycle.

1.2- Machine learning:

According to Andrew Ng, it's the science of getting computers autonomous without being explicitly programmed. This science is so pervasive that it is used dozens of times in a day without knowing it. [1]

Arthur Samuel also defined ML as a field of study that gives the ability of computers to learn without being learned. During his career, he programmed thousands of games against himself, and by analyzing the game situations that lead to winning and those that lead to losing, the game program learned over time what to do in each case. From this, a new AI field named machine learning was born. [1]

Nowadays, many fields and industries use it: manufacturing, retail, healthcare and life sciences, travel and hospitality, financial services, energy, feedstock, and utilities.

It helps make the right decisions, detect diseases, risk analytics, regulate and optimize the businesses.

Therefore, this technology can change our way of life and make our future more accessible, and it becomes just a question of three clicks.

1.3- Different Types of machine learning:

There exist four types of machine learning algorithms: supervised, unsupervised, semi-supervised, and reinforcement. [2]

Supervised learning: in this type of learning, the operator provides a specific data set to the machine, including the desired inputs and outputs. In this type, algorithms find ways to fit inputs and outputs, and then operators correct the resulting predictions.

Unsupervised learning: the ML algorithm looks for patterns by studying the data. There is no operator or a human to provide instructions. The algorithm alone understands the relationship between the inputs and the outputs, and it can make predictions by interpreting this relation.

Semi-supervised learning: it's similar to the previous type. The only difference is that this one use labelled and unlabeled data, while labelled data are pieces of information that contain meaningful tags, and the unlabeled data lacks that information.

Reinforcement learning: focuses on structured learning processes. It introduces algorithms with initial actions, parameters, and values. Then, based on the rules, it explores new possibilities to get the results to determine the optimal one. Thus, it earns from past experiences and adapts its approach to the situation to get optimal results.

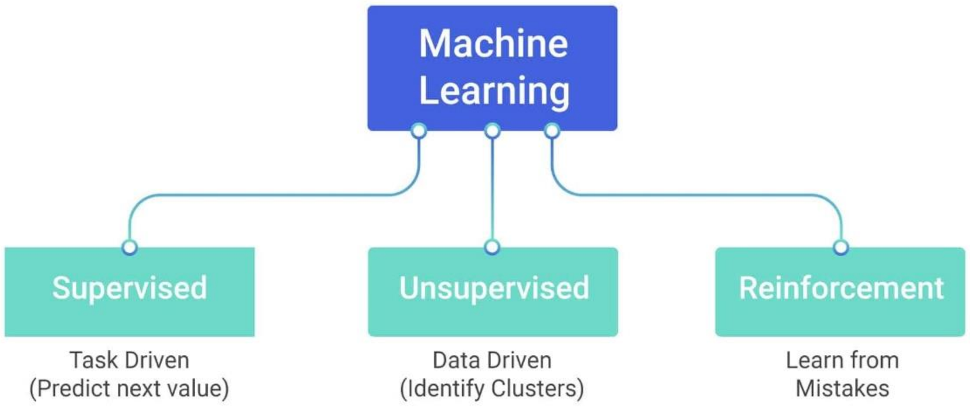


Figure 1 Different types of ML algorithms

1.4- What is MLOps?

MLOps is a set of practices that unifies machine learning systems' development (dev) and deployment (ops) to simplify the management process. MLOps also addresses the automation of Machine Learning deployment, including Deep Learning models. [3]

These practices aim to standardize and streamline the life cycle management in chine learning systems because it became a vital component to successful data science project implementation. [4]

It is a solution for assisting companies, and business executives generate long-term value and reduce risk connected with data science, machine learning, and artificial intelligence programs. Nevertheless, it is a relatively new concept that started in 2015 from a paper titled "Hidden Technical Debt in Machine Learning Systems”. From that, Its growth has exponentially increased, and the market of MLOps solutions expects to reach 4$ billion by 2025. [5]

The following figure describes the interest in MLOps over time.

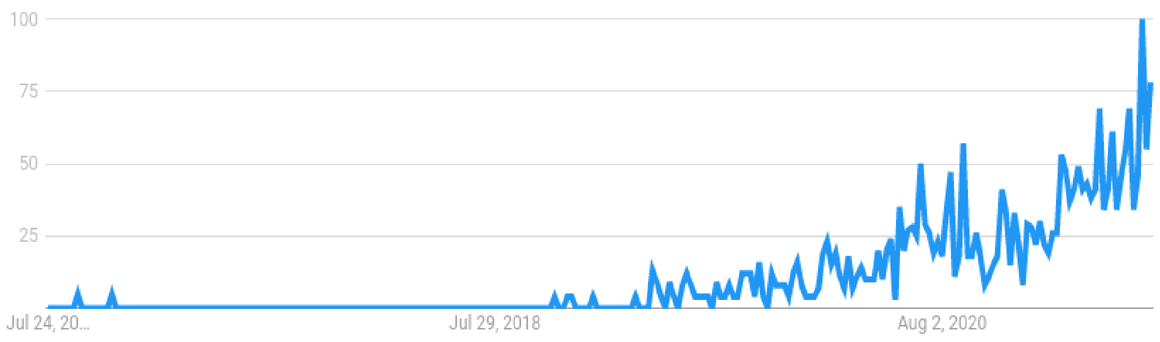


Figure 2 Interest in MLOps overtime

1.5- MLOps challenges:

Developing machine learning models and their deployment in a production environment is still relatively new for most traditional companies.

But recently, the number of the deployed ML models has increased mainly with the arrival of decision automation, where models have become more critical. On the other side, the management of its risks became more important at the top level. [6]

Setting a machine learning model in enterprises is much more complex in terms of needs and materials.

To understand the significant challenges that face MLOps, we can resume them in three main points:

- The data and the business needs are constantly changing, so the results need to be continually relayed back to the business to ensure that the actual results align with the expectations and, critically, meet the original goal of this model.
- The second major challenge is that MLOps involves many people from the business, data science, and IT teams. And none of these groups is using the same technical language and materials. In fact, in many cases, they use the same fundamental skills to communicate with each other.
- As a third major challenge, most data scientists are specialized in model building and assessment. But the problem is that they may find themselves playing the role of a software engineer and becoming specialists more on the deployment or operational side, which is not easy for them.

We can have a better idea about these challenges from the following figure.

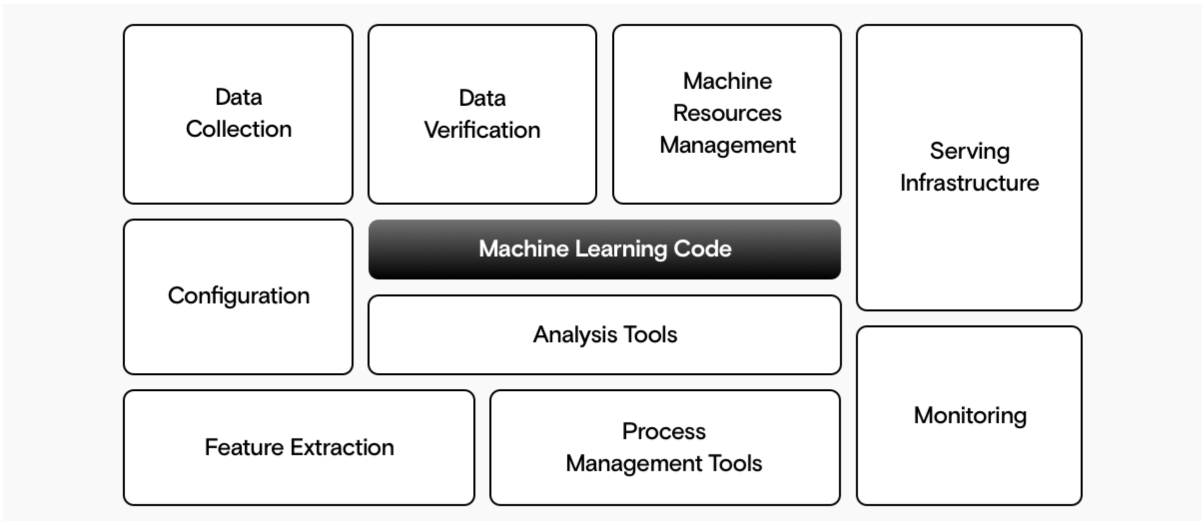


Figure 3 MLOps challenges

1.6- Who is concerned during the MLOps process?

Although data scientists build ML models, it's wrong to think that only data scientists can benefit from robust MLOps processes. MLOps involves many people in the life cycle of the ML system: [6]

Subject matter experts: provide the business questions and goals and ensure that the model aligns with the initial needs.

Data scientists: mainly work on the model and its delivery to be used in production environments and with production data.

Data engineers: their primary mission is to optimize the retrieval and use of data to power ML models.

Software engineers: as they are software specialists, they integrate the models in the company's systems and software and ensure the seamless working of these ML programs with non-ML-based applications.

DevOps: They Conduct and build operational systems and test them. They also ensure continuous Integration/Delivery (CI/CD) pipeline management.

Model risk managers/auditors: they reduce the company's overall risk due to ML models in production and make sure that they comply with internal and external standards.

Machine learning architects: they prepare the environment of the models from design to development and monitoring and introducing new technologies to improve the model's performances.

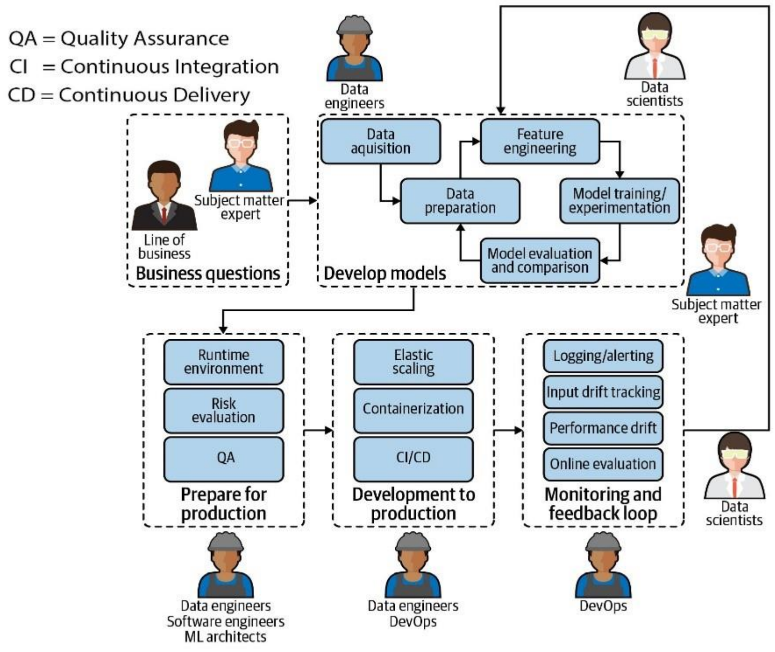


Figure 4 Involved people in the MLOps process

1.7- MLOps life cycle:

To better understand the MLOps process, we have to go ahead and analyze its life cycle that we can resume in 4 primary phases, according to Andrew Ng, the founder of DeepLearning.AI. [7]

1.7.1- Scoping:

The first step of any machine-learning project is the definition of the objectives. Then, it helps to determine feasible solutions to a problem. Often, this is a good starting point for most projects because we often find out that there is someone who might have tried to tackle the same problem statement, sins we would have to adapt the pre-developed open-source model to our data sets.

Principally, these pre-existing models give an idea about the solution to the main problem such that the other phases of the life cycle become easy.

1.7.2- Data collection:

After the project definition, it's essential to select the data which feeds the future model. It's necessary mainly in supervised learning because the data quality affects the model quality automatically.

While working with the data, it's necessary to ensure the clean labels and their uniformity; for this, taking enough time in the data collection is more than essential, and it's preferable to start working with the model simultaneously.

Andrew ng divided data collection into two different steps. The first one is the data definition, which consists of establishing a data baseline. The second step is data labelling which is crucial in the case of supervised learning. It aims to identify each element within an image by annotating the content manually.

1.7.3- Modelling:

Most researchers and scientists may think that it's the most crucial phase, but they are wrong. The real-world data that the model interacts with differs from the data introduced in the precedent phase. Indeed, it may force a come back to the last step to introduce new data to improve the model's performance in the future.

During the modelling, the engineers can repeat three main steps many times to optimize the results. For example, suppose a company wants to change the model environment or introduce a different data set. In that case, the data scientist must select new data, train the model, and analyze any error if it occurs.

1.7.4- Deployment:

If the process stops in the last phase, it is like nothing was done. The integration of ML models in production is vital to becoming useful for the end-user. We can classify the deployment into three types:

- **Shadow Deployment:** in this type of deployment, humans take the final decision, irrespective of what the model predicts. It's mainly used to determine if the model is processing well and which points it fails on.
- **Canary Deployment**: here, the model interacts with a small set of data, on which he is allowed to make decisions. So, depending on the model's performance, the traffic may be increased or pulling back this model to adjust it.
- **Blue-Green deployment**: blue-green deployment refers respectively to old and new versions of deployment. They are principally used to separate the development version from the deployed versions of the model. Thus, it allows detecting bugs, and the rollback to the old stable versions becomes accessible.

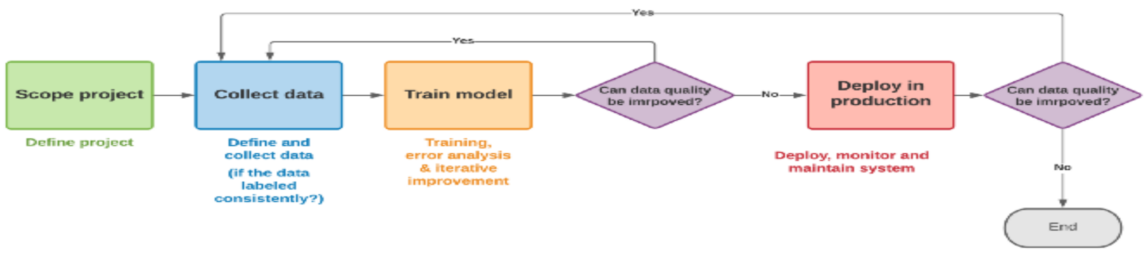

Scope project
Define project
Collect data
Define and collect data
(if the data labeled consistently?)
Train model
Training, error analysis & iterative improvement
Can data quality be imrpoved?
Yes
No
Deploy in production
Deploy, monitor and maintain system
Can data quality be imrpoved?
Yes
No
End

Figure 5 ML project lifecycle

1.8- Chapter conclusion:

In this chapter, we gave a general idea about the MLOps, as it is a new field that involves people from different computer science fields. Then we presented the significant challenges facing this new field, and we finished describing the MLOps lifecycle process.

In the next chapter, we talk about the project needs and the different required configurations in order to implement the expected solutions.

Chapter 2

Modeling and development

2.1- Introduction

We present in this chapter the working methodology of our project and the different configurations made during the development process, the functional needs necessary for the actors to interact with the system, and the non-functional needs to improve the software quality of the system.

2.2- The methodology and languages adopted

We are always looking for the most efficient and fastest methods for the realization of our project. We use the agile methodology for this purpose. Scrum is the most straightforward agile framework

Scrum guarantees us the best overview of the project, aims to reduce difficulties such as lack of planning, work is done through short cycles called Sprints. Within a Sprint, our team works from a list of items called the Backlog (see figure below).

We use the UML (Unified Modeling Language) modelling language to specify the needs and requirements of the actors, the system, and the overall architecture

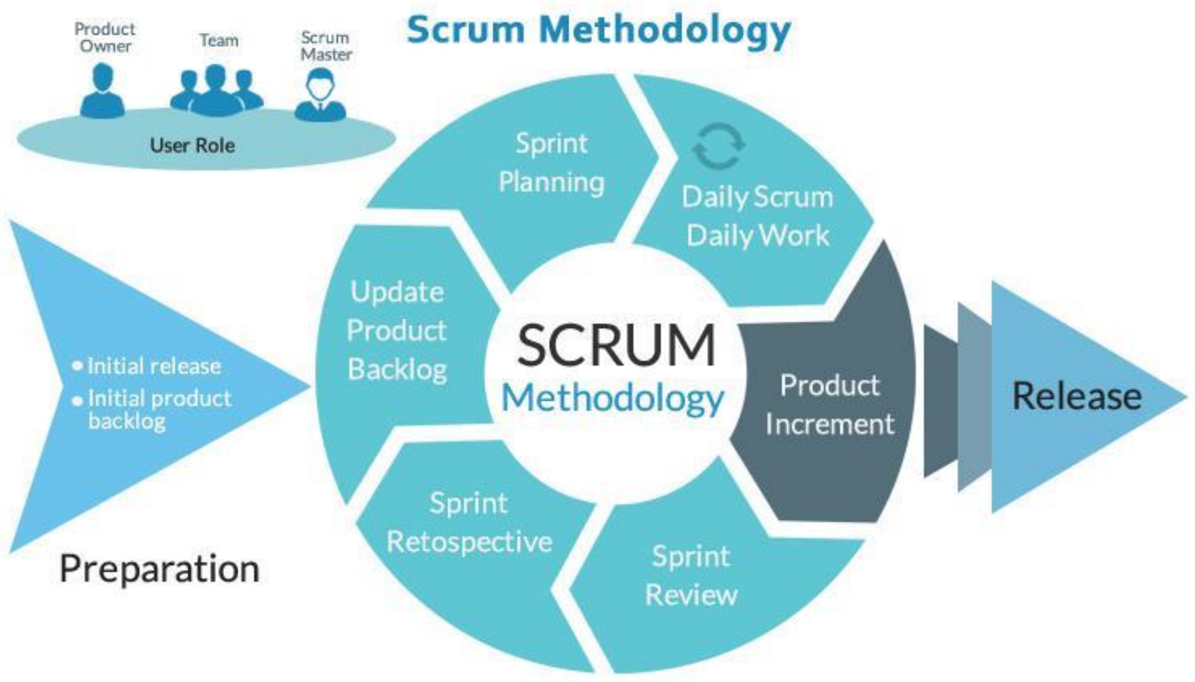


Figure 6 Scrum methodology

2.3- Specification of functional needs

The application to be produced must offer a set of functionalities related to a group of user needs. These define the services that users expect to see provided by this application.

This web application must meet the following functional needs:

2.3.1- Medical images processing:

One of the main functionalities of this application is to permit the user to analyze his medical images as skin images to get a segmented skin image.

2.3.2- Object detection processing:

The user can scan images and videos of his own choice, and in return, he gets a file of the same type as the one uploaded containing all the objects detected.

2.3.3- Face and gender detection processing:

This functionality allows the user to detect the faces and the gender of the people contained in the uploaded images.

2.3.4- Contact admin by mailing form:

The user of this application has the right to contact the administrator by email just using a contact form visible to all the website visitors.

2.3.5- Backoffice:

This part is the most important service of this application. It allows the administrator to have a general view of the application. He can view the results of the predictions made by the different services, getting the id of the user who did the action and mostly the date and time of the prediction.

2.4- Specification of non-functional needs:

A non-functional need is a restriction or constraint on a service in the system, such as environmental and implementation constraints and performance requirements, project dependencies, ease of maintenance, scalability, and reliability.

As part of this work, the application must meet these needs:

Security: The upload of files is secured with data encryption, and the sending of the emails via the website uses TLS encryption.

Efficiency: The application must be functional regardless of any circumstances that may surround the user (a practical solution)

Validity: (correction, correctness, conformity), carry out precisely the tasks defined in the specification

The performance: The response time is short.

Reliability: The data provided by the application must be reliable, and the solution must give correct results.

2.5- UML language:

UML was used in this project as a modelling language. UML (Unified Modeling Language) is a graphical modelling language. It appeared in the world of software engineering as part of "object-oriented modelization".

it represents several advantages as:

- The UML is based on the mechanisms of abstraction, hierarchy, and decomposition.
- It is a powerful communication medium: it allows, thanks to its graphic representation, to visually express an object solution, to facilitate the comparison and the evolution of the solution
- it is a formal and standardized language. It allows access to precise information and guarantees stability.
- It facilitates the understanding of complex abstract representations. It is based on the implementation of diagrams representing static and dynamic views in developing a computer application.

To better design our application, we need to use the use case diagram, which allows us to specify the functionalities of our system and their interactions with the user. Also, the sequence diagram aims to explain in detail how operations perform in chronological order.

2.6- Use case diagram:

Use case diagrams are a technique for capturing the functional needs of the system; They describe the interactions between users of the system and the system itself. In other words, they help to formalize the way that future users operate with the application.

2.6.1- Identification of the different actors:

An actor is a person, hardware, or software that interacts with the system to perform functions regarding the use cases, in the following, an identification of the main actors who interact directly with our application.

there are 2 types of actors:

The administrator: He is a user with power. He is responsible for the administration of the application.

The user: It is a user with restricted power. He can use the different services of the web application.

2.6.2- Identification of the use cases:

Actor	Use case
Admin	● Authentification ● Manage the web application
User	● Medical image processing ● Object detection processing ● Face and gender detection processing ● Contact the admin by email

2.6.3- Global use case diagram:

The figure below represents the different use cases of our system.

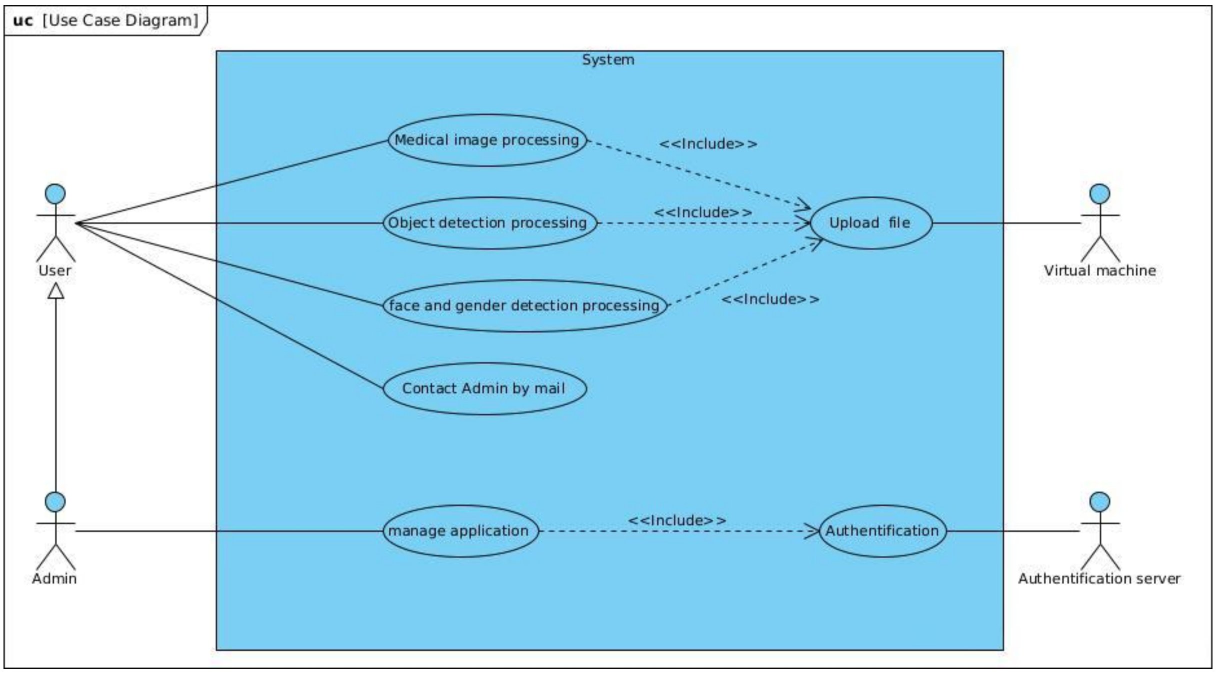


Figure 7 Use case diagram

2.6.4- Textual description of each use case:

These descriptions permit us to have a general view of the workflow of each use case, and then we can model the different sequence diagrams.

2.6.4.1- Textual description of the use case "medical image processing":

Title: medical image processing	
Actor	User
precondition	access to the web application and select the service medical image processing
post-condition	Upload a file
Scenario	After the uploading file, the system checks the validation of this file. In the backend, the skin segmentation script is running. It returns the prediction result.
exception	File uploaded is not a nifti file, or it's a corrupt file

2.6.4.2- Textual description of the use case "object detection processing":

Title: object detection processing	
Actor	User
precondition	access to the web application and select the service object detection processing
post-condition	Upload a file
Scenario	After the uploading file, the system checks the validation of this file. In the backend, the object detection script is running. It returns the prediction result.
exception	File uploaded is not an image nor a video, or it's a corrupt file

2.6.4.3- Textual description of the use case "face and gender detection processing":

Title: face and gender detection processing	
Actor	User
precondition	Access to the web application and select the service face and gender detection processing
post-condition	Upload a file
Scenario	After the uploading file, the system checks the validation of this file. In the backend, the face and gender detection script is running. It returns the prediction result.
exception	File uploaded is not an image, or it's a corrupt file

2.6.4.4- Textual description of the use case "contact admin by email":

Title: contact admin by email	
Actor	User
precondition	Access to the contact us web page
post-condition	Filling the contact form fields
Scenario	the system checks the validation of the inserted data. If everything is ok, the SMTP server sends an email to the admin containing the inserted data.
exception	Data not valid or SMTP server error

2.6.4.5- Textual description of the use case "authentification":

Title: authentification	
Actor	Admin
precondition	Access to the administration page
post-condition	The admin is logged-in
Scenario	The system shows the authentification page. The admin fills the login form. The system checks the validity of the inserted data, and if they are correct, the admin is redirected to the administration dashboard
exception	Mail/password credentials are not valid

2.6.4.6- Textual description of the use case "manage application":

Title: manage application	
Actor	Admin
precondition	authentification

post-condition	The files list is shown
Scenario	The system shows the list of the different files predicted with their date
exception	No file records in the database to show

2.7- The proposed solutions

To meet the needs mentioned above, we propose two solutions with different tools and architectures:

- Solution 1: Deploy Machine Learning Pipeline on the cloud using Docker Container and a ReactJS client interface
- Solution 2: Deploy Machine Learning Pipeline using Google Drive API and Google Colab virtual machine.

2.7.1- Solution 1: Deploy Machine Learning Pipeline on the cloud using Docker Container and a ReactJS client interface

2.7.1.1- Global view of the solution:

This part demonstrates deploying a machine learning pipeline as a web app using the Google cloud platform service for hosting.

In this technique, we created a frontend ReactJS application where the user can navigate through the various pages, select a service to use, upload a file to a particular model, and finally receive the result with the option to download it. Thanks to a FastAPI API, when the user uploads a file, it is stored on a cloud server directory. The model performs on the cloud virtual machine, and the results are returned to the main application (see figure below).

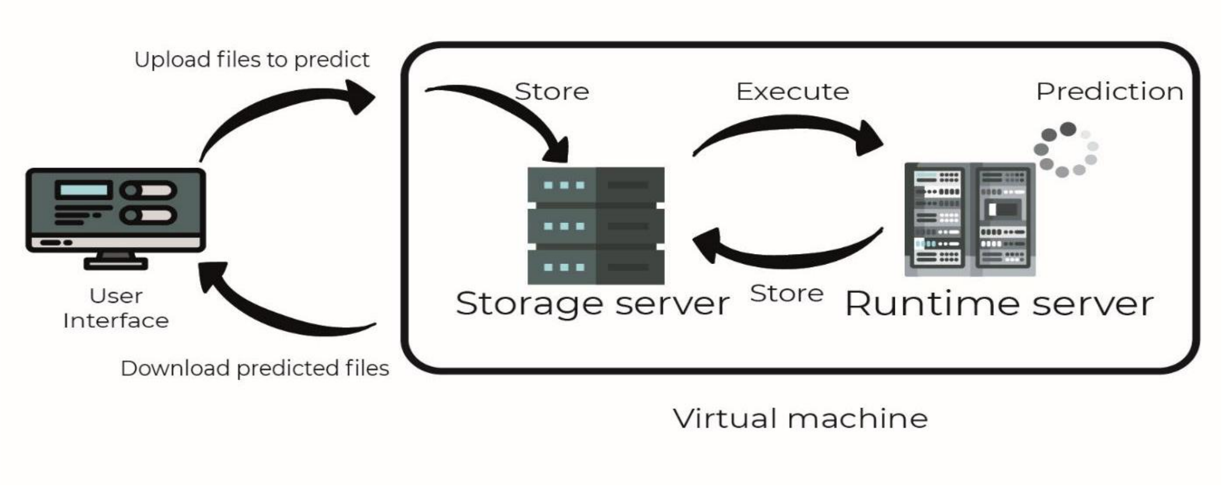


Figure 8 Deploy Machine Learning Pipeline on the cloud using Docker Container and a ReactJS client interface

2.7.1.2- Sequence diagrams for the proposed solution:

In this part, we introduce the sequence diagrams of the 4 prominent use cases, which are:

- medical image processing
- object detection processing
- face and gender detection processing
- management of the application

Sequence diagram of the case "medical image processing":

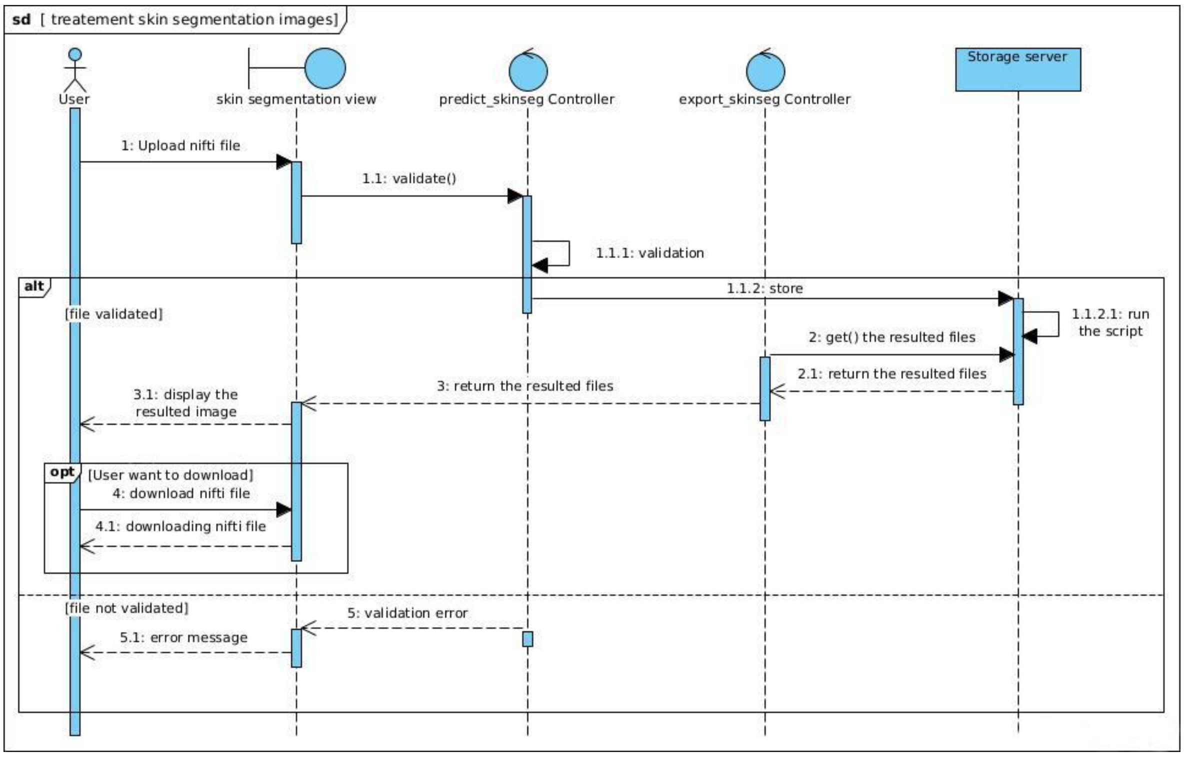


Figure 9 Sequence diagram of the case: medical image processing

Sequence diagram of the case "object detection processing":

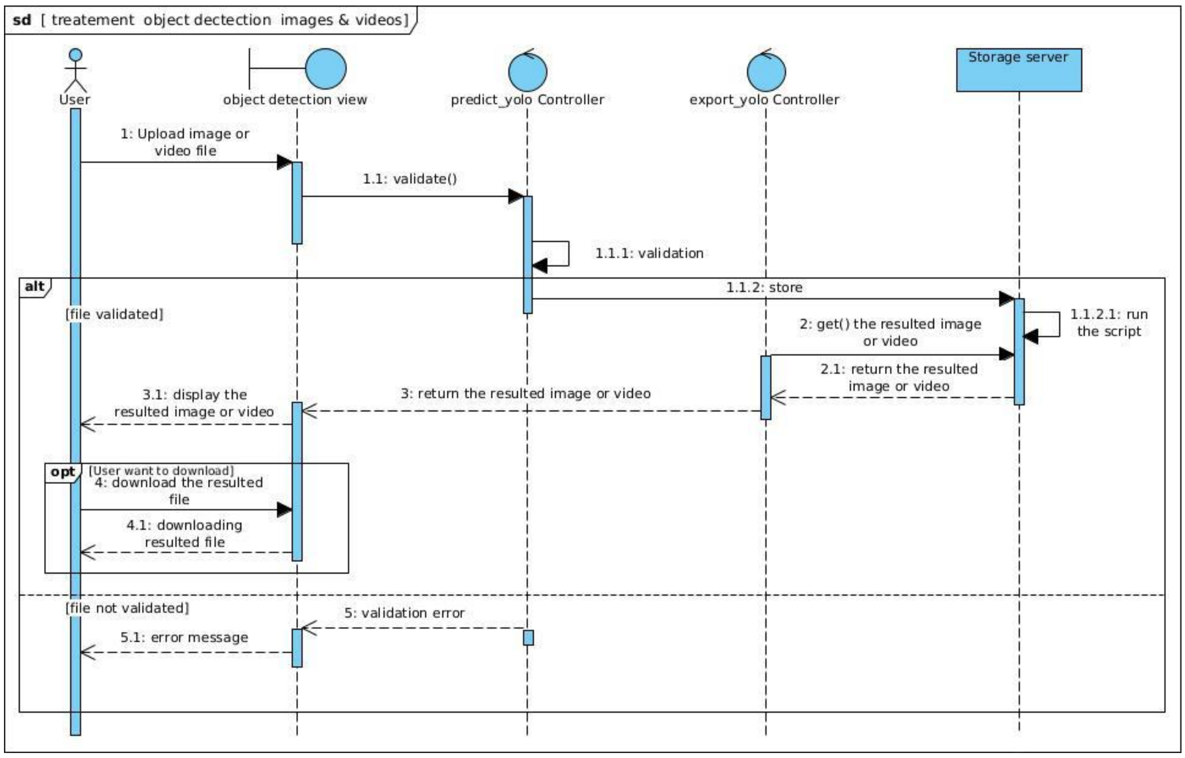

sd [treatement object dectection images & videos]
User
object detection view
predict_yolo Controller
export_yolo Controller
Storage server
1: Upload image or video file
1.1: validate()
1.1.1: validation
alt
[file validated]
1.1.2: store
1.1.2.1: run the script
2: get() the resulted image or video
2.1: return the resulted image or video
3: return the resulted image or video
3.1: display the resulted image or video
opt
[User want to download]
4: download the resulted file
4.1: downloading resulted file
[file not validated]
5: validation error
5.1: error message

Figure 10 Sequence diagram of the case: object detection processing

Sequence diagram of the case "face and gender detection processing":

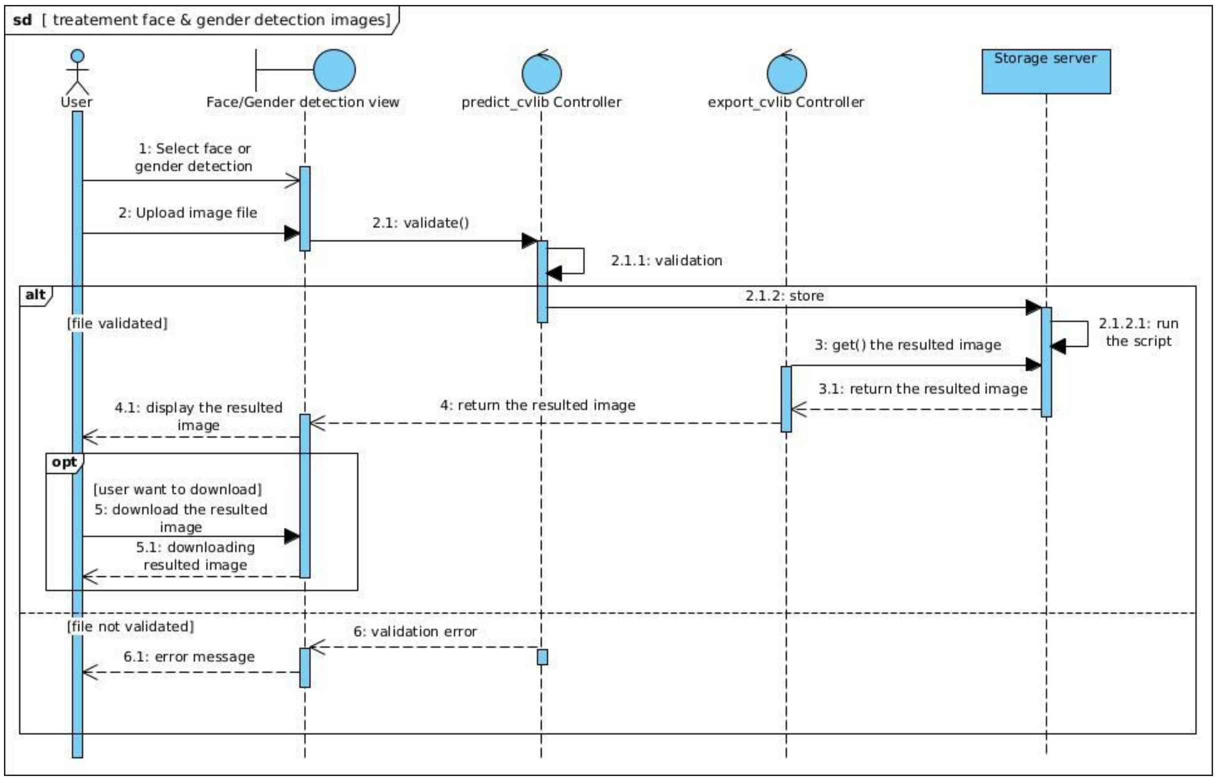

sd [treatement face & gender detection images]
User
Face/Gender detection view
predict_cvlib Controller
export_cvlib Controller
Storage server
1: Select face or gender detection
2: Upload image file
2.1: validate()
2.1.1: validation
alt
[file validated]
2.1.2: store
2.1.2.1: run the script
3: get() the resulted image
3.1: return the resulted image
4: return the resulted image
4.1: display the resulted image
opt
[user want to download]
5: download the resulted image
5.1: downloading resulted image
[file not validated]
6: validation error
6.1: error message

Figure 11 Sequence diagram of the case: face and gender detection processing

Sequence diagram of the case "manage application":

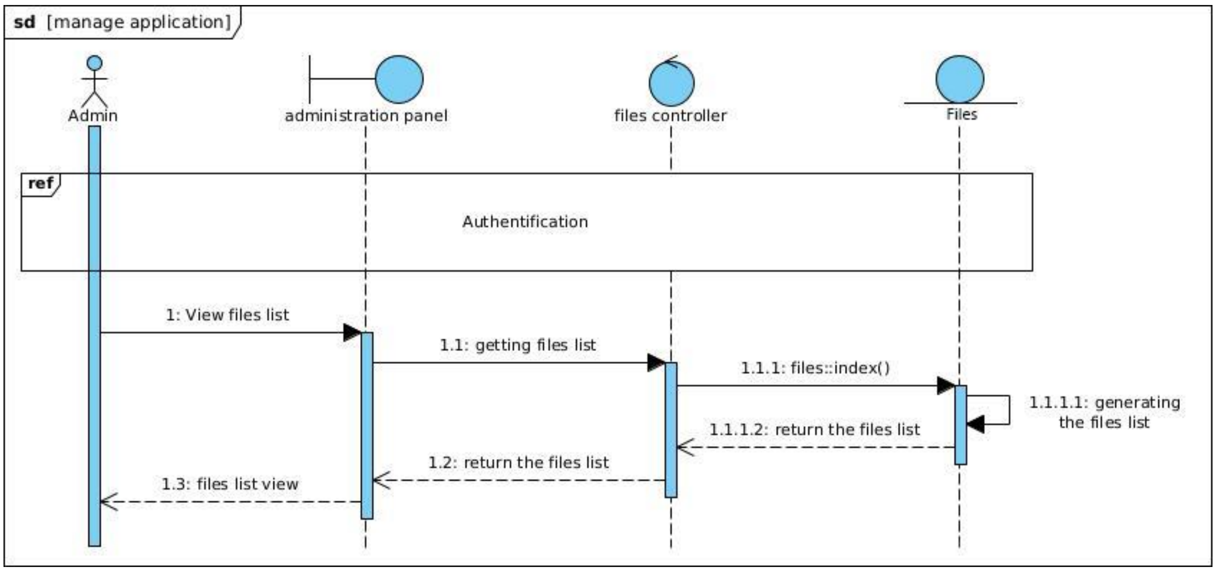


Figure 12 Sequence diagram of the case: manage application

2.7.1.3- Deployment diagram

This diagram describes the way that the application is deployed on the server.

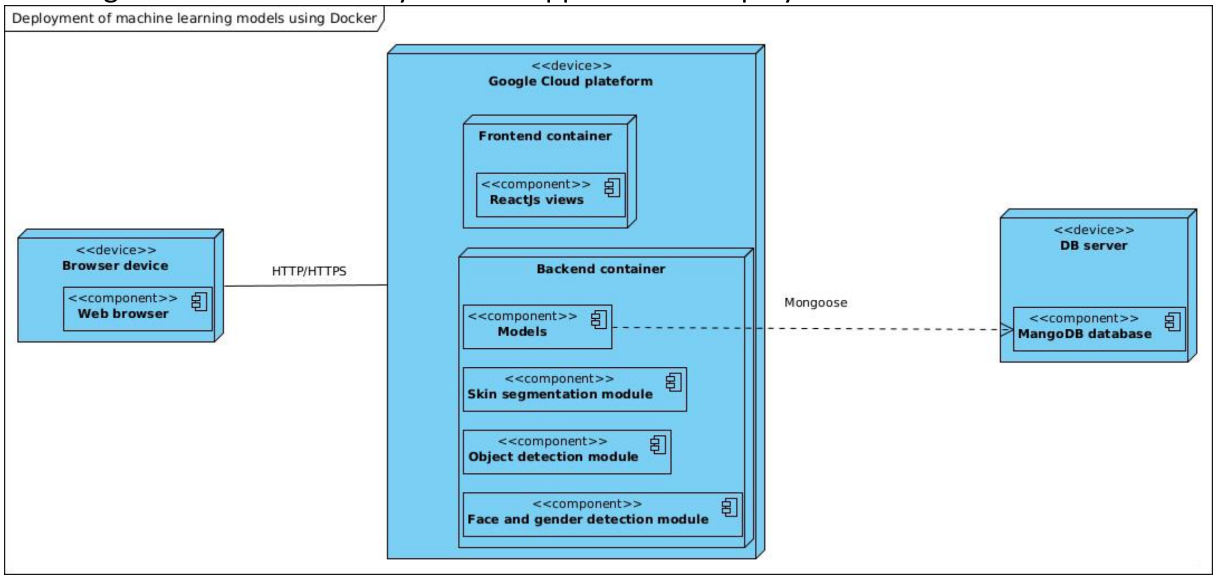

Deployment of machine learning models using Docker
<<device>> Browser device
<<component>> Web browser
HTTP/HTTPS
<<device>> Google Cloud plateform
Frontend container
<<component>> ReactJs views
Backend container
<<component>> Models
<<component>> Skin segmentation module
<<component>> Object detection module
<<component>> Face and gender detection module
Mongoose
<<device>> DB server
<<component>> MangoDB database

Figure 13 deployment diagram of the first solution

2.7.1.4- Required configurations for the deployment:

For the deployment of this solution, we have to make many configurations that we can categorize as follows:

- Frontend configurations
- Backend configurations
- Hosting server configurations
- Programming tools configurations.

Frontend configurations :

Virtualizing a solution based on ReactJS.

Prerequisites:

- Install Docker locally
- Install NPM

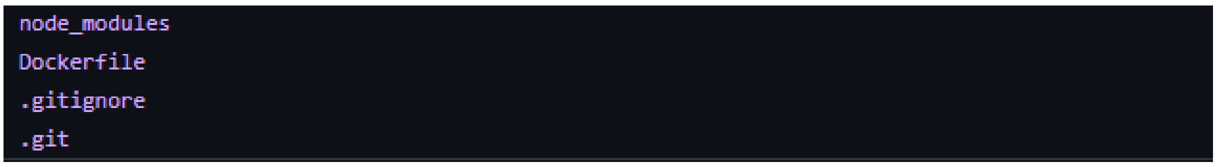
Writing the Dockerfile:

In step 1, we transfer our application code to the **"*app*"** folder, install the application dependencies from the package.json file, and make a production build.
In step 2, we establish an Nginx server using the Nginx server image and deploy our app to it by transferring the build components from the "***/app/build***" folder to the Nginx server at the "***/usr/share/nginx/html***" directory

Creating a "***.dockerignore***" file:

Although this file isn't required, it's a good idea to include it since it may speed up the image build process and keep the image slim by removing redundant code from the Docker build context, so it doesn't end up in the picture.

Here is the content of our ***.dockerignore*** :

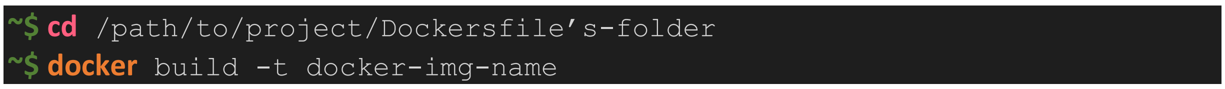


```
node_modules
Dockerfile
.gitignore
.git
```

Figure 14 .dockerignore source code

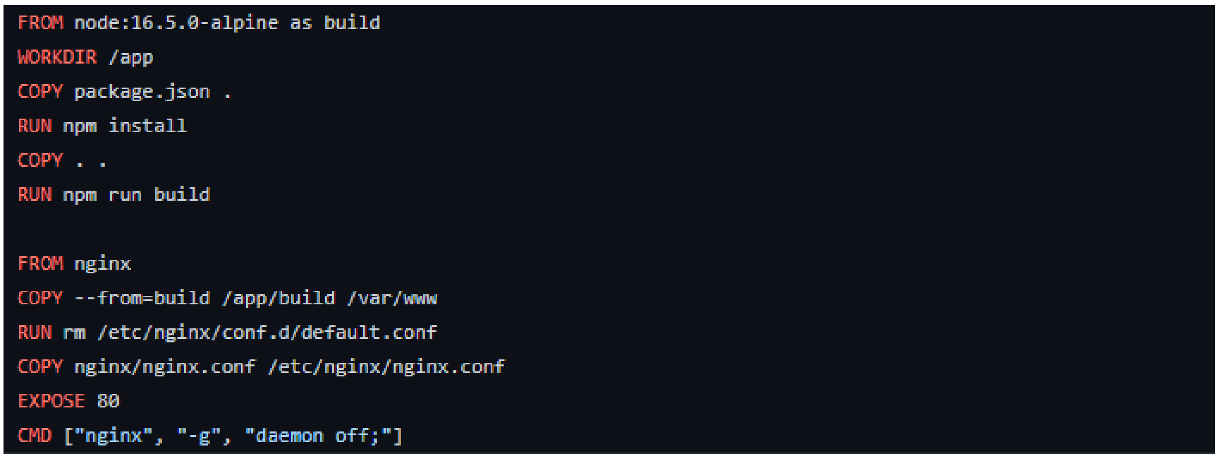
Creating a ***Docker*** image:

Run the Docker build command to generate a Docker image of our app based on the Dockerfile we just created:



```
~$ cd /path/to/project/Dockersfile's-folder
~$ docker build -t docker-img-name
```

Based on the **Dockerfile** that contain:



```
FROM node:16.5.0-alpine as build
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
RUN npm run build

FROM nginx
COPY --from=build /app/build /var/www
RUN rm /etc/nginx/conf.d/default.conf
COPY nginx/nginx.conf /etc/nginx/nginx.conf
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

Figure 15 Dockerfile source code

Let's break down this command and its flags:

- **–t:** builds a ***Docker*** image called **docker-img-name**

To get a list of the images produced with our ***ReactJS*** app image, we have the ***Docker*** command:

```
~$ docker images
```

We also used node and Nginx images and an intermediary image <none> to construct our app image. These pictures, however, aren't necessary and may be removed.

Running the ***Docker*** container:

To build a Docker container based on the image produced in the previous step, we used the following Docker command:

```
~$ docker run --rm -p 80:80 -v ~:/home/the_user_name/ docker-img-name
```

The –p option is used to map the container port to the external host port using 80:80/TCP because the app in the container runs on port 80.

Backend configurations:

Virtualizing a solution based on FastAPI and Jupiter.

Prerequisites:

- Install ***Docker*** locally.

Writing the ***Dockerfile:***

- In step 1: transfer the application code to the "***app***" folder.
- In step 2: Install the application dependencies from the "***requirements.txt***" file.
- In step 3: we start "***Supervisord***" to run two different processes in the same container with different execution ports.

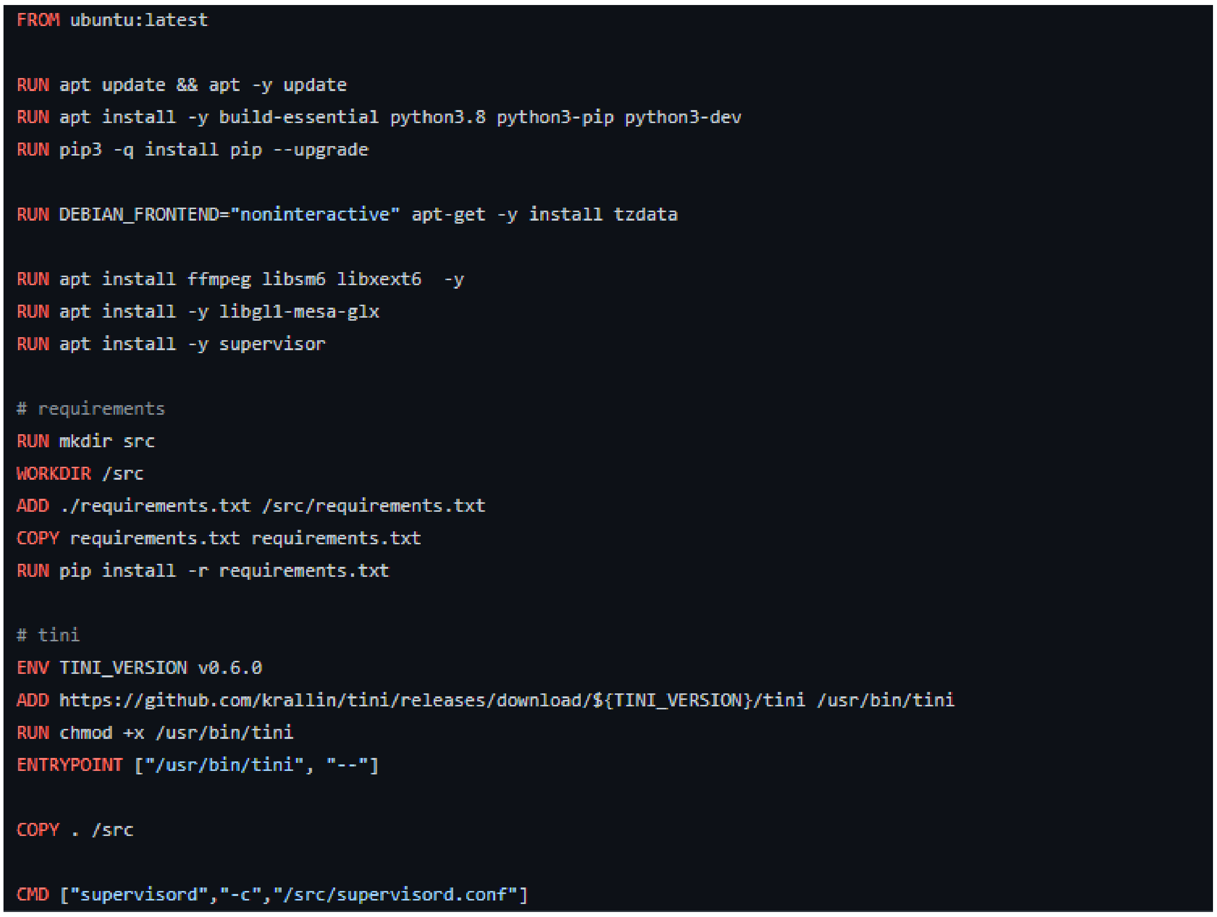
Creating a ***Docker*** image:

We built a Dockerfile with all of the necessary parts to launch our solution using Docker.

The image needed to perform this ungraded lab locally may be run using the following command:

```
~$ cd  /path/to/project/dockerfile_folder
~$ docker  build -t  docker-image-name  .
```

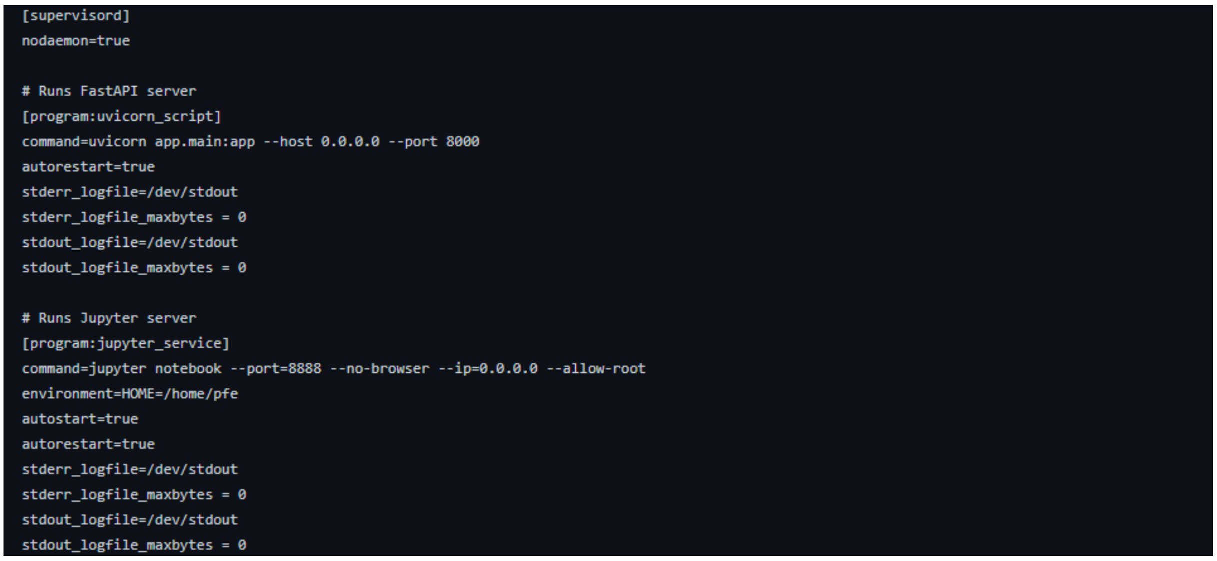
Based on the **Dockerfile** that contain:



```
FROM ubuntu:latest

RUN apt update && apt -y update
RUN apt install -y build-essential python3.8 python3-pip python3-dev
RUN pip3 -q install pip --upgrade

RUN DEBIAN_FRONTEND="noninteractive" apt-get -y install tzdata

RUN apt install ffmpeg libsm6 libxext6  -y
RUN apt install -y libgl1-mesa-glx
RUN apt install -y supervisor

# requirements
RUN mkdir src
WORKDIR /src
ADD ./requirements.txt /src/requirements.txt
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt

# tini
ENV TINI_VERSION v0.6.0
ADD https://github.com/krallin/tini/releases/download/${TINI_VERSION}/tini /usr/bin/tini
RUN chmod +x /usr/bin/tini
ENTRYPOINT ["/usr/bin/tini", "--"]

COPY . /src

CMD ["supervisord","-c","/src/supervisord.conf"]
```

Figure 16 Dockerfile source code

The last line of the Dockerfile is launching the servers in the same container (FastAPI and Jupyter) using Supervisor which is a Process Control System, here is its configuration file content:



```
[supervisord]
nodaemon=true

# Runs FastAPI server
[program:uvicorn_script]
command=uvicorn app.main:app --host 0.0.0.0 --port 8000
autorestart=true
stderr_logfile=/dev/stdout
stderr_logfile_maxbytes = 0
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes = 0

# Runs Jupyter server
[program:jupyter_service]
command=jupyter notebook --port=8888 --no-browser --ip=0.0.0.0 --allow-root
environment=HOME=/home/pfe
autostart=true
autorestart=true
stderr_logfile=/dev/stdout
stderr_logfile_maxbytes = 0
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes = 0
```

Figure 17 Supervisor source code

Running the ***Docker*** container:

We used the following Docker command to create a Docker container based on the image created in the previous step:

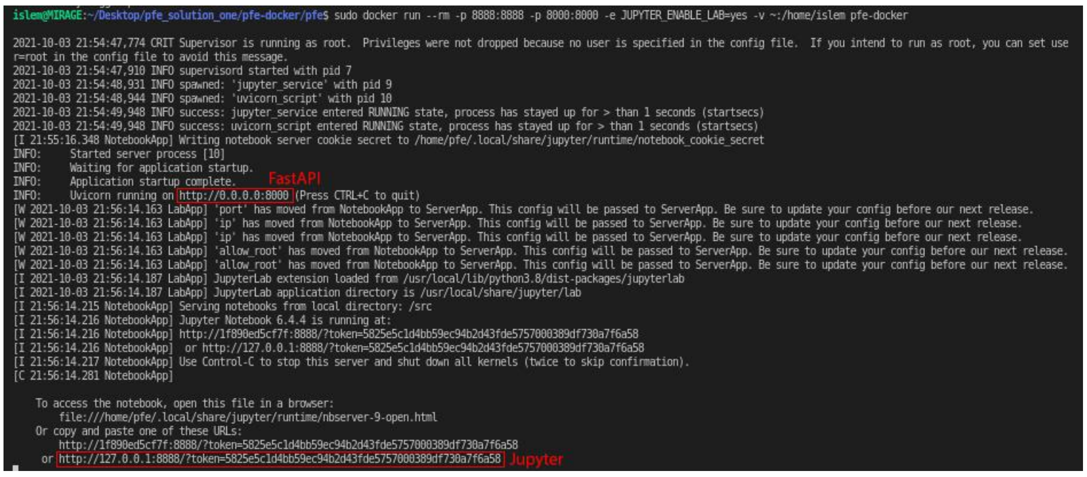
```
~$ docker  run --rm -p 8888:8888 -p 8000:8000 -e JUPYTER_ENABLE_LAB=yes -v
~:/home/user/ docker-image-name
```

Let's break down this command and its flags:

- **-it**: Runs the container in interactive mode and connects a pseudo-terminal to it so we can see what's printing in the container's standard streams. It is critical since we need to copy and paste the Jupyter lab access token.
- **--rm:** After stopping the container, this option deletes it.
- **-p:** This option lets us map a port on our PC to a port on the container. In this scenario, we need two ports: one for the Jupyter server and one for the ungraded lab server.
- **-v:** this option tells Docker which folders should be mounted inside the container so they may be accessed. Docker is in charge of managing the placement of anonymous volumes. It's worth noting that referring to the same volume while it's anonymous might be challenging. Run docker run -v /path/in/container -e to establish an anonymous volume. JUPYTER ENABLE LAB=yes - Tells the startup script to use the Jupyter lab command instead of the standard Jupyter notebook command. Setting environment variables is more accessible than altering command-line arguments in container orchestration settings.

When the container begins to run, some information is printed in the terminal. To utilize the Jupyter lab, we'll usually need to authenticate. We need to copy the token that appears on the console and paste it at http://ip_address:8888/.

The output from our terminal should look something like this, with the token highlighted for reference:



```
islem@MIRAGE:~/Desktop/pfe_solution_one/pfe-docker/pfe$ sudo docker run --rm -p 8888:8888 -p 8000:8000 -e JUPYTER_ENABLE_LAB=yes -v ~:/home/islem pfe-docker

2021-10-03 21:54:47,774 CRIT Supervisor is running as root.  Privileges were not dropped because no user is specified in the config file.  If you intend to run as root, you can set use
r=root in the config file to avoid this message.
2021-10-03 21:54:47,910 INFO supervisord started with pid 7
2021-10-03 21:54:48,931 INFO spawned: 'jupyter_service' with pid 9
2021-10-03 21:54:48,944 INFO spawned: 'uvicorn_script' with pid 10
2021-10-03 21:54:49,948 INFO success: jupyter_service entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2021-10-03 21:54:49,948 INFO success: uvicorn_script entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
[I 21:55:16.348 NotebookApp] Writing notebook server cookie secret to /home/pfe/.local/share/jupyter/runtime/notebook_cookie_secret
INFO:     Started server process [10]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
[W 2021-10-03 21:56:14.163 LabApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-10-03 21:56:14.163 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-10-03 21:56:14.163 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-10-03 21:56:14.163 LabApp] 'allow_root' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-10-03 21:56:14.163 LabApp] 'allow_root' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2021-10-03 21:56:14.187 LabApp] JupyterLab extension loaded from /usr/local/lib/python3.8/dist-packages/jupyterlab
[I 2021-10-03 21:56:14.187 LabApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 21:56:14.215 NotebookApp] Serving notebooks from local directory: /src
[I 21:56:14.216 NotebookApp] Jupyter Notebook 6.4.4 is running at:
[I 21:56:14.216 NotebookApp] http://1f890ed5cf7f:8888/?token=5825e5c1d4bb59ec94b2d43fde5757000389df730a7f6a58
[I 21:56:14.216 NotebookApp]  or http://127.0.0.1:8888/?token=5825e5c1d4bb59ec94b2d43fde5757000389df730a7f6a58
[I 21:56:14.217 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:56:14.281 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/pfe/.local/share/jupyter/runtime/nbserver-9-open.html
    Or copy and paste one of these URLs:
        http://1f890ed5cf7f:8888/?token=5825e5c1d4bb59ec94b2d43fde5757000389df730a7f6a58
     or http://127.0.0.1:8888/?token=5825e5c1d4bb59ec94b2d43fde5757000389df730a7f6a58
```

Figure 18 launch of the docker instance

After we've authenticated, we navigate into the "***/home/the_user_name/ directory***", where we shall see all of our current local directory's files. To begin the ungraded lab, look for the "***/notebooks/. file***" and open it.

We hit "Ctrl + C" twice to close the container when we're through with the lab. The container is deleted as well.

Serving the models using ***FastAPI***:

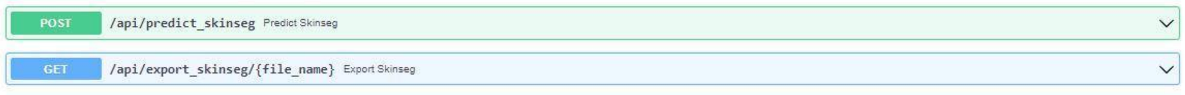
The installation is created as part of the docker image building process. Before we start deployment, we go over some key ideas and how they apply to ***FastAPI***. Let's make a directory to store the images sent to the server via a post request.

The following step concerns the use of this instance to construct endpoints that handles the prediction logic.

Endpoints:

To manage our project using routes, we need to develop our Endpoints. Here is the skin Segmentation endpoint example (see figure below):

- http://ip_address:8000/predict_skinseg
  - Import the inputs into the Virtual Machine
  - Run the model
  - Store the outputs
- http://ip_address:8000/api/export_skinseg/{file_name}
  - To export the model's output according to the file name


POST /api/predict_skinseg Predict Skinseg

GET /api/export_skinseg/{file_name} Export Skinseg

Figure 19 Skin segmentation routes

FastApi provides an integrated interface that we may use to execute our API for development purposes instead of installing other applications (such as ***Postman***).

Once all of the code is in place, all we have to do is launch the server using the command:

```
~$ uvicorn.run(app)
```

The API is written in ***FastAPI***, while the serving is handled by ***Uvicorn***, a lightning-fast implementation of the Asynchronous Server Gateway Interface (***ASGI***). Both technologies are intertwined, so we won't have to worry about the implementation specifics. For this lab, knowing that ***Uvicorn*** is in charge of serving is adequate.

We're attempting to run many services in the same container, as we can see:

- [program:uvicorn_script]: to start the FastApi application on port 8000
- [program:jupyter_service]: to start the Jupyter notebook in port 8888

This manipulation is not possible. As a result, we'll use a process manager such as ***Supervisord***. It is a somewhat heavy-weight method that needs us to package ***Supervisord*** and its settings and the many programs it controls in our image (or base our image on one that contains supervisord).

And finally we created the last container using the command :

```
~$ docker run --rm -p 27018:27017 -v ~:/home/user mongo
```

Which pulls a predefined docker hub image for mongodb deployment and runs it, as it is this database chosen for the monitoring and the authentication options.

and in case of wanting to access the database using the shell, here is the command to use:

```
~$ mongo  --port 27018
```

Hosting server configurations:

Google Cloud Platform is a Google cloud computing platform that provides hosting on the same infrastructure that Google uses internally for products like its search engine.

We may manage our Google Cloud projects and resources using the Google Cloud Console, a web-based graphical user interface. When we use the Cloud Console, we either start a new project or choose an existing one and then use the resources we generate in that project.

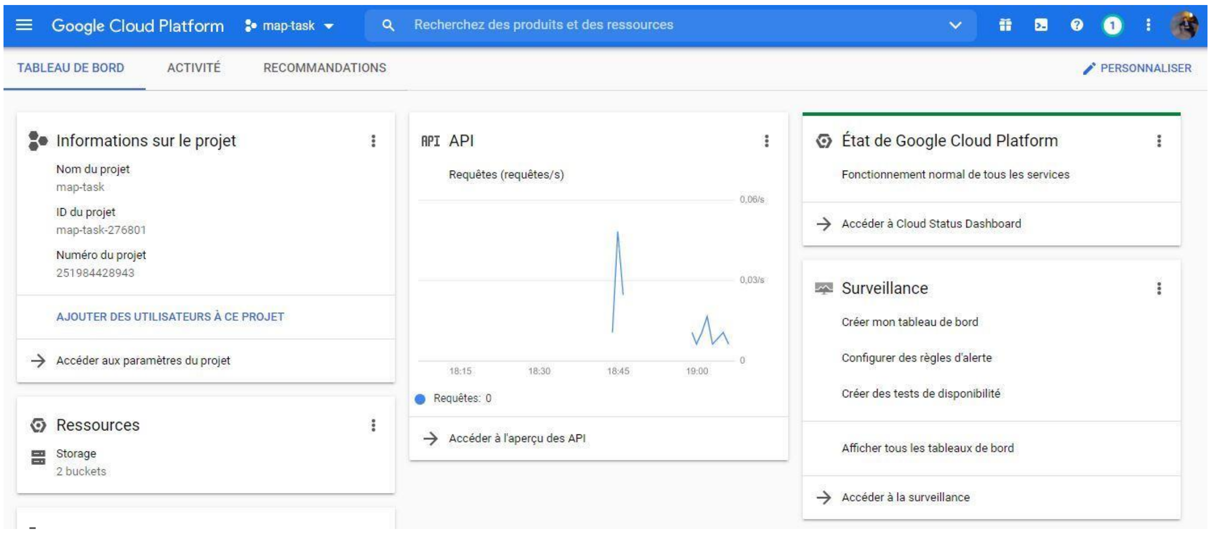
Technology selection: new clients receive $300 in free credits to explore and assess Google Cloud Platform for 90 days thoroughly. We won't be charged unless and until we want to upgrade. To be used for:

- Run workloads with no cost.
- There are over 20 free items available.
- Get free hands-on experience with popular products like Compute Engine and Cloud Storage up to a monthly limit. These free services do not have an expiration date.

To complete our free trial registration, we'll need to set up a cloud services billing account and verify our identity using a credit card or other payment method. No worries, creating a cloud billing account prevents them from charging us. We are not invoiced until we upgrade our Cloud Invoicing account to a paid account and expressly activate billing. During the trial term, we can upgrade to a paid account at any moment. We can still utilize any remaining credits after upgrading (within the 90-day limit).

Google cloud platform reporting:

Dashboards are one way to see and analyze metric data that is important to us. Custom dashboards are those that we create or install on our own. The Dashboards tab of the Google



Google Cloud Platform map-task Recherchez des produits et des ressources

TABLEAU DE BORD ACTIVITÉ RECOMMANDATIONS PERSONNALISER

Informations sur le projet
Nom du projet
map-task
ID du projet
map-task-276801
Numéro du projet
251984428943
AJOUTER DES UTILISATEURS À CE PROJET
Accéder aux paramètres du projet

Ressources
Storage
2 buckets

API API
Requêtes (requêtes/s)
Requêtes: 0
Accéder à l'aperçu des API

État de Google Cloud Platform
Fonctionnement normal de tous les services
Accéder à Cloud Status Dashboard

Surveillance
Créer mon tableau de bord
Configurer des règles d'alerte
Créer des tests de disponibilité
Afficher tous les tableaux de bord
Accéder à la surveillance

Figure 20 Google cloud platform dashboard

Cloud Console (see figure below) provides a selected set of dashboards that we may preview and subsequently install.

Suppose we have access to a dashboard's JSON representation, such as if it is stored on GitHub or a local server, we may install it using the Cloud Console or the Cloud monitoring API. Dashboard definitions for several Google Cloud services may be found in the monitoring-dashboard-samples project on GitHub.

This Dashboard contains presentations using widgets such as:

- The graph in a line
- Graph of stacked areas
- stacked bar graph
- Heatmap diagram
- Chart of warnings

Google Cloud Platform provides analysis reports that give a high-level view of the latency to our application for all or a subset of requests. Google Cloud Platform provides a daily report using a "Trace" that compares the previous day's performance to the same day the previous week. The daily report is shown on the Trace Overview page and listed on the Analysis report page.

Access the GCP remotely:

To access the cloud virtual machine remotely and manage our source code, we had to generate an SSH key by running these commands on our local shell:

```
~$ docker  run --rm -p 27018:27017 -v ~:/home/user mongo
~$ ssh-keygen -t rsa -f ~/.ssh/keyfile -C "comment for a keyfile "
~$ cd ~/.ssh
~$ nano keyfile.pub
```

We are then copying the content of the "remote_server_public_ssh_key.pub" file. Finally, navigate to "Compute Engine" and paste our ssh key into our virtual machine instance information.

To access the server remotely using SSH run from our local shell:

```
~$ ssh -i keyfile user_name@ip_adress
```

Configuration of the programming tools:

To modify our code hosted on the server using VScode, we shall set these configurations:

- Install ***Remote-SSH*** extension on VS Code.

To use this extension, we follow these steps:

- we need to press on F1 and then: ***Open SSH Host***
- Select: ***~/ssh/config***
- In the input box that opens, we should type the user's name and host/IP address in the following format and click enter: user@host-or-ip or user@domain@host-or-ip
- Enter our password if prompted (but we suggest setting up key-based authentication).

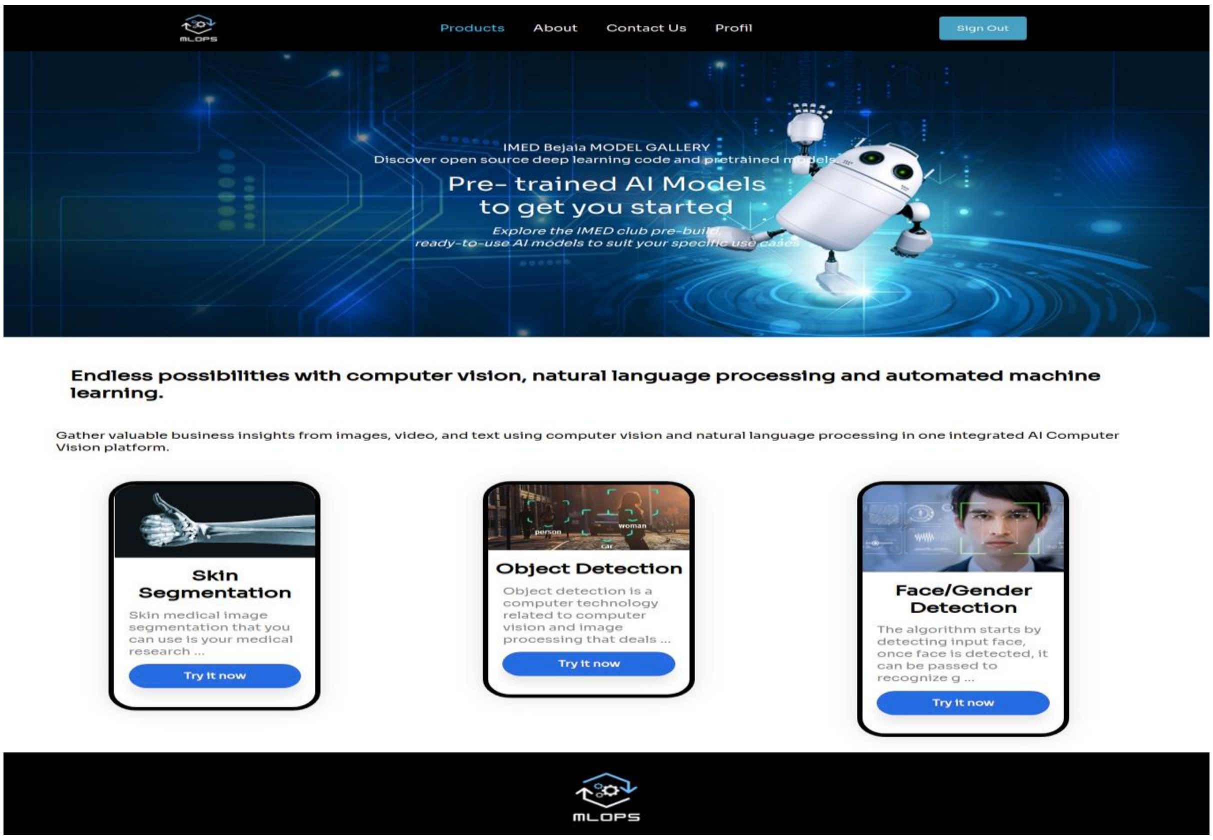
Example:

```
Host hostname
        HostName x.x.x.x
        User keyfile
        IdentityFile ~/.ssh/ keyfile
```

We may also receive a list of the most popular commands by clicking on the "Quick Access" status bar item in the lower-left corner.

2.7.1.5- Application screenshots:

Here are some screenshots of the application proposed for this solution (see figures below).



IMED Bejaia MODEL GALLERY
Discover open source deep learning code and pretrained models
Pre- trained AI Models to get you started
Explore the IMED club pre-build ready-to-use AI models to suit your specific use cases

Endless possibilities with computer vision, natural language processing and automated machine learning.

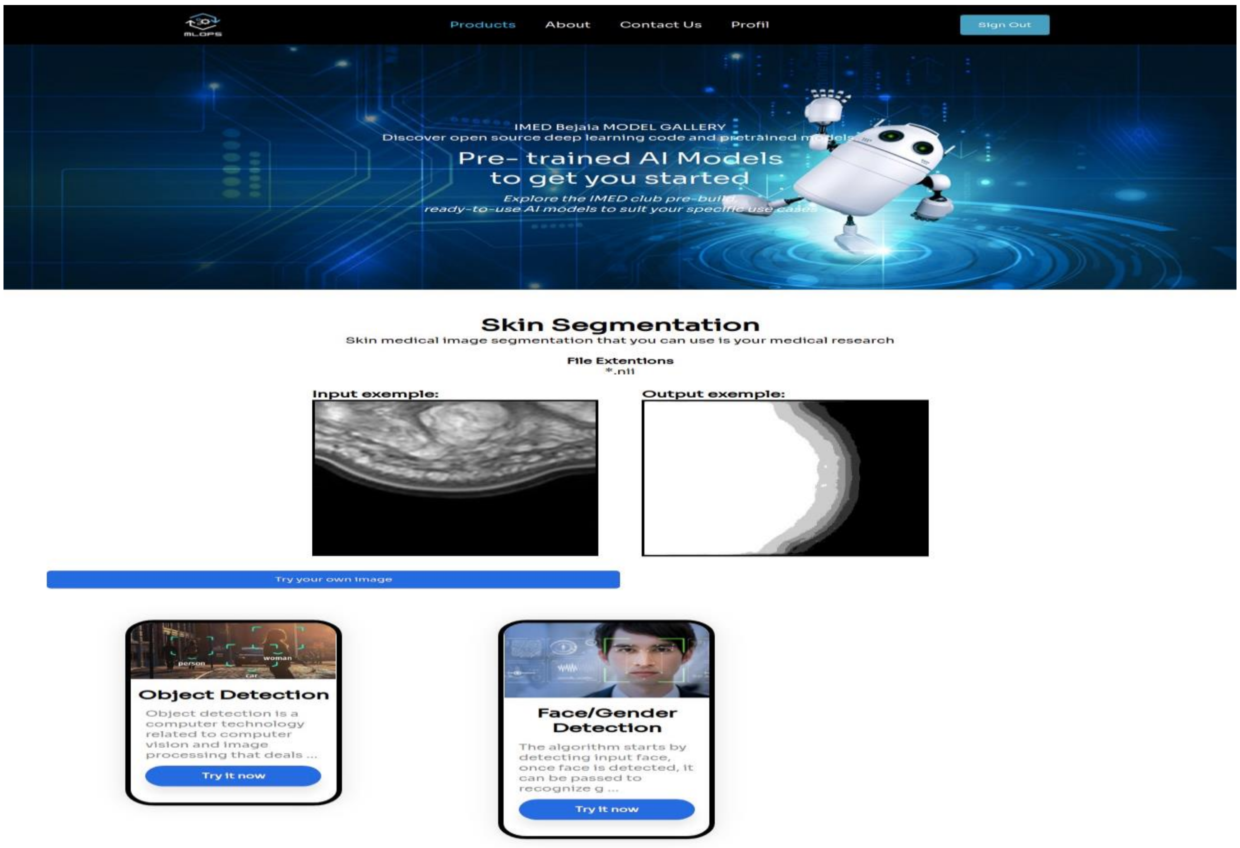
Gather valuable business insights from images, video, and text using computer vision and natural language processing in one integrated AI Computer Vision platform.

Skin Segmentation
Skin medical image segmentation that you can use is your medical research ...
Try it now

Object Detection
Object detection is a computer technology related to computer vision and image processing that deals ...
Try it now

Face/Gender Detection
The algorithm starts by detecting input face, once face is detected, it can be passed to recognize g ...
Try it now

Figure 21 Services webpage screenshot



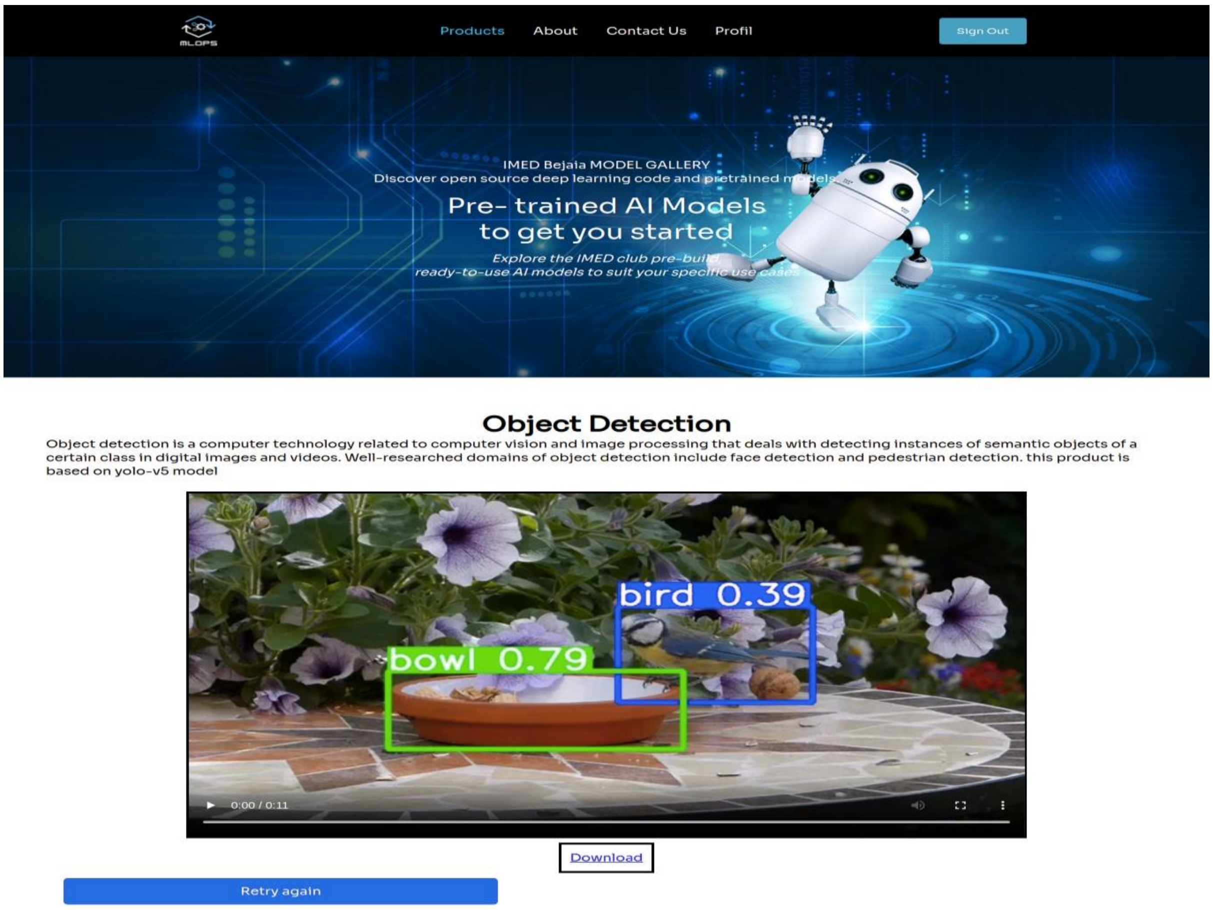
IMED Bejaia MODEL GALLERY
Discover open source deep learning code and pretrained models
Pre- trained AI Models to get you started
Explore the IMED club pre-build ready-to-use AI models to suit your specific use cases

Skin Segmentation
Skin medical image segmentation that you can use is your medical research
File Extentions
*.nii
Input exemple:
Output exemple:
Try your own image

Object Detection
Object detection is a computer technology related to computer vision and image processing that deals ...
Try it now

Face/Gender Detection
The algorithm starts by detecting input face, once face is detected, it can be passed to recognize g ...
Try it now

Figure 22 Skin segmentation webpage screenshot



IMED Bejaia MODEL GALLERY
Discover open source deep learning code and pretrained models.

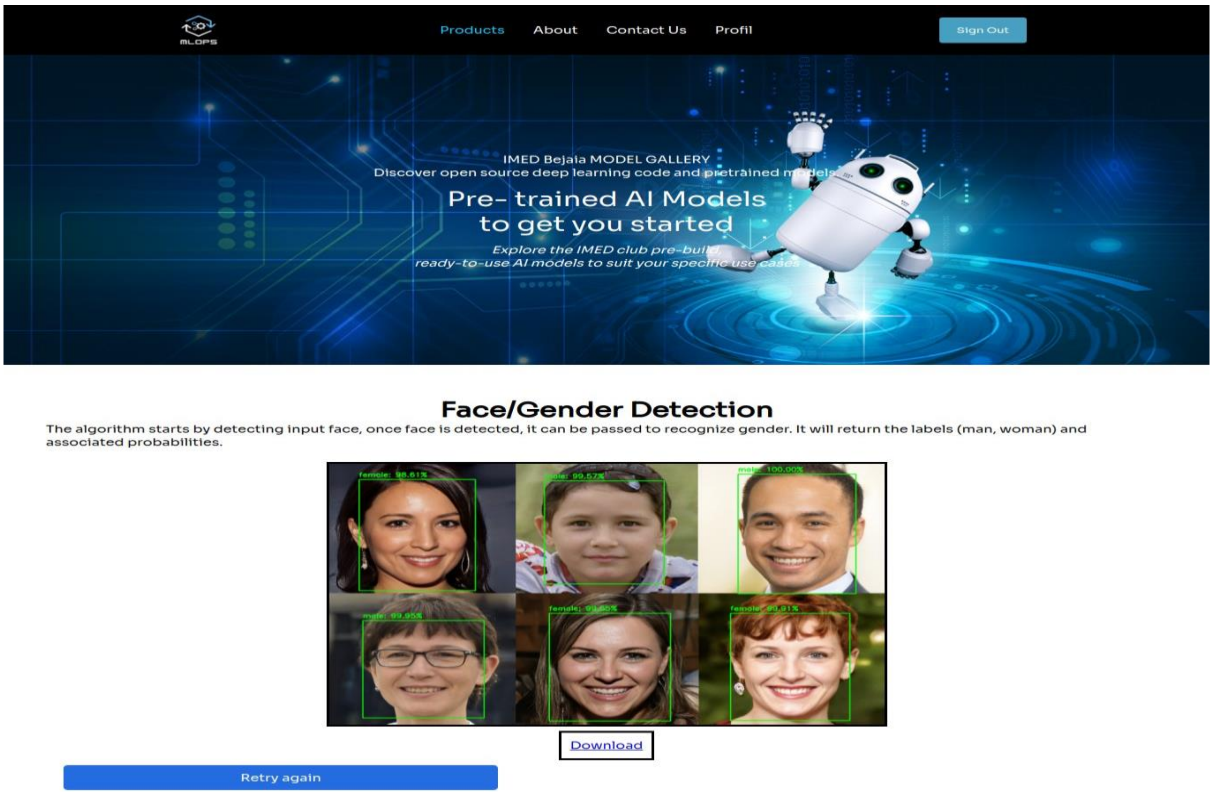
Pre- trained AI Models
to get you started

Explore the IMED club pre-build,
ready-to-use AI models to suit your specific use cases

Object Detection

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. this product is based on yolo-v5 model

Figure 23 Object detection webpage screenshot



IMED Bejaia MODEL GALLERY
Discover open source deep learning code and pretrained models

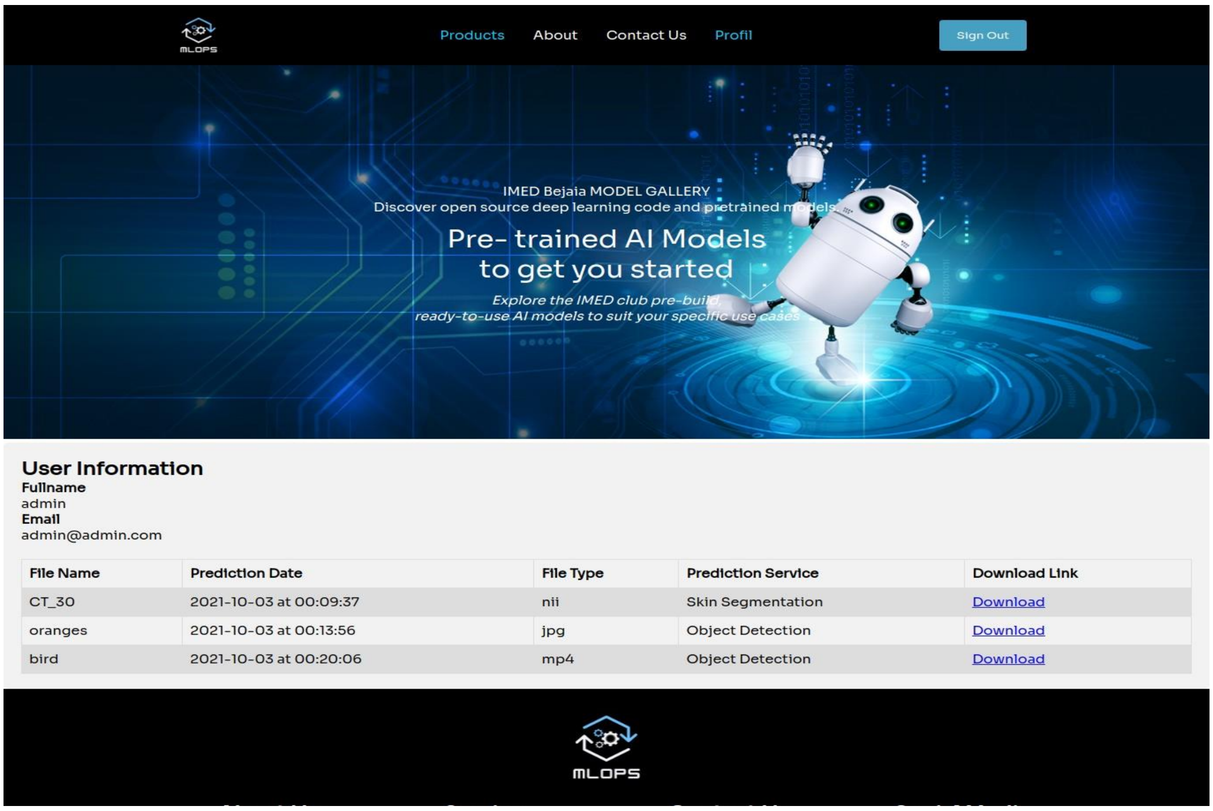
Pre- trained AI Models
to get you started

Explore the IMED club pre-build ready-to-use AI models to suit your specific use cases

Face/Gender Detection

The algorithm starts by detecting input face, once face is detected, it can be passed to recognize gender. It will return the labels (man, woman) and associated probabilities.

Figure 24 Face and gender detection webpage screenshot



IMED Bejaia MODEL GALLERY
Discover open source deep learning code and pretrained models

Pre- trained AI Models
to get you started

Explore the IMED club pre-build ready-to-use AI models to suit your specific use cases

User Information

Fullname
admin
Email
admin@admin.com

File Name	Prediction Date	File Type	Prediction Service	Download Link
CT_30	2021-10-03 at 00:09:37	nii	Skin Segmentation	Download
oranges	2021-10-03 at 00:13:56	jpg	Object Detection	Download
bird	2021-10-03 at 00:20:06	mp4	Object Detection	Download

Figure 25 Dashboard files list screenshot

2.7.2- Solution 2: Deploying Machine Learning Pipeline using Google Drive API and Google Colab virtual machine

2.7.2.1- Global view of the solution:

In this method, we worked on a Laravel application as the main application. The user can surf across the different pages, choose a service to test, upload a file to test the specific algorithm and finally get the result with the possibility of downloading this last one.

When the user uploads the file, it is stored on Google Drive using Google Drive API. the Google Colab virtual machine runs the Machine Learning model using this file, and the result is returned to the main application through an ssh tunnel. Then, it is displayed on the view and give the user the right to download it.

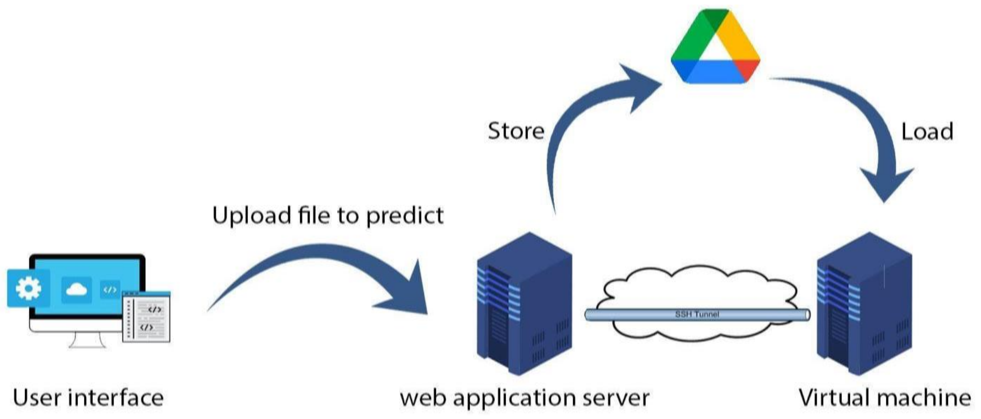


Figure 26 Deploying Machine Learning Pipeline using Google Drive API and Google Colab virtual machine

2.7.2.2- Sequence diagrams for the proposed solution:

As we have two applications with different architectures, the sequence diagrams are also different, so in the following, the sequence diagrams of the second solution:

Sequence diagram of the case "medical image processing":

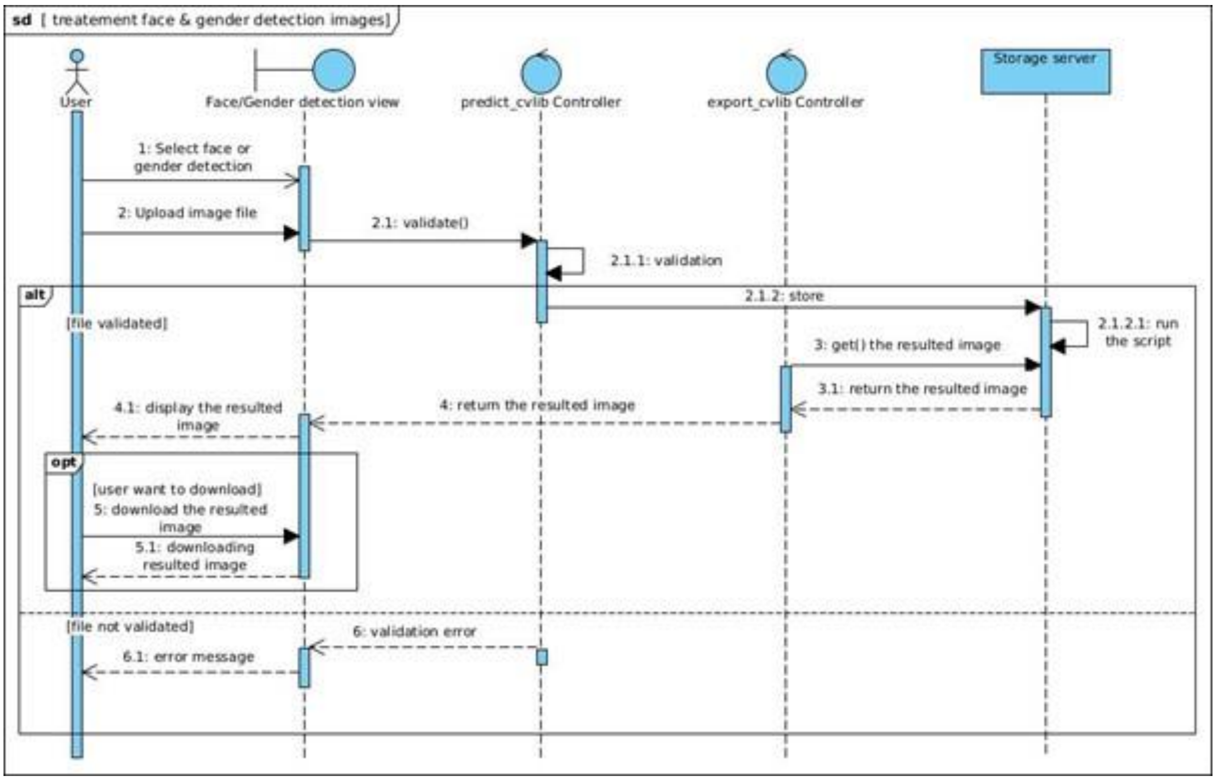

sd [treatement face & gender detection images]
User
Face/Gender detection view
predict_cvlib Controller
export_cvlib Controller
Storage server
1: Select face or gender detection
2: Upload image file
2.1: validate()
2.1.1: validation
alt
2.1.2: store
[file validated]
2.1.2.1: run the script
3: get() the resulted image
3.1: return the resulted image
4: return the resulted image
4.1: display the resulted image
opt
[user want to download]
5: download the resulted image
5.1: downloading resulted image
[file not validated]
6: validation error
6.1: error message

Figure 27 Sequence diagram of the case: medical image processing

Sequence diagram of the case "object detection processing":

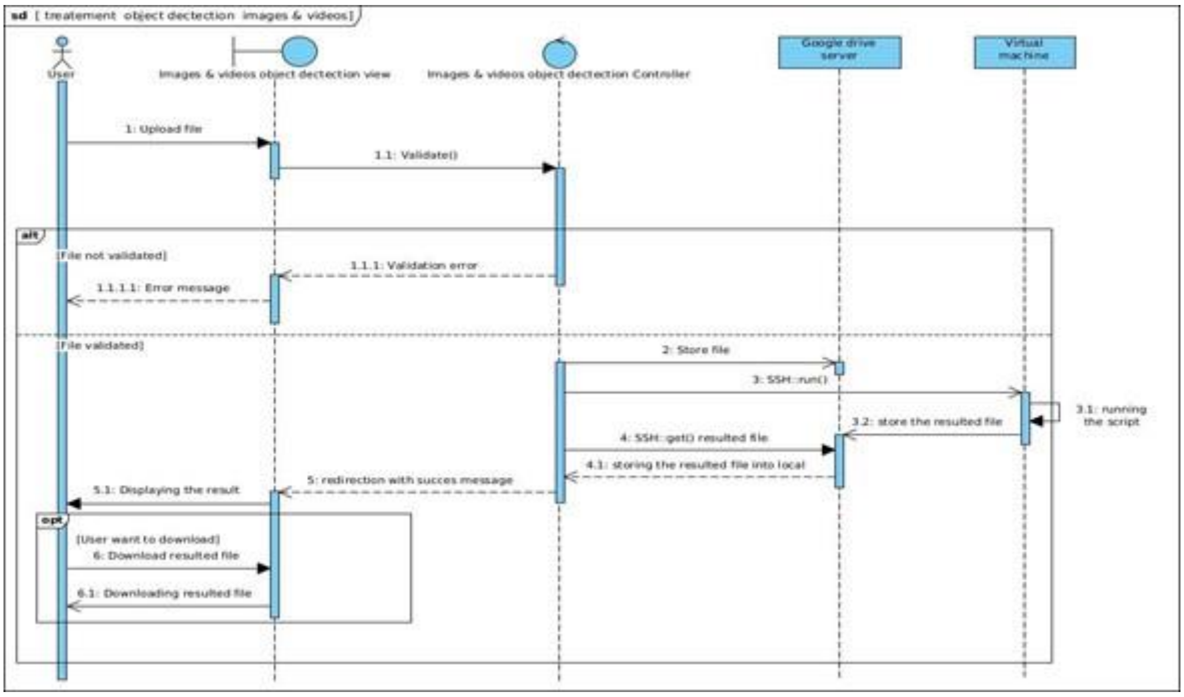

sd [treatement object dectection images & videos]
User
Images & videos object dectection view
Images & videos object dectection Controller
Google drive server
Virtual machine
1: Upload file
1.1: Validate()
alt
[File not validated]
1.1.1: Validation error
1.1.1.1: Error message
[File validated]
2: Store file
3: SSH::run()
3.1: running the script
3.2: store the resulted file
4: SSH::get() resulted file
4.1: storing the resulted file into local
5: redirection with succes message
5.1: Displaying the result
opt
[User want to download]
6: Download resulted file
6.1: Downloading resulted file

Figure 28 Sequence diagram of the case: object detection processing

Sequence diagram of the case "face and gender detection processing":

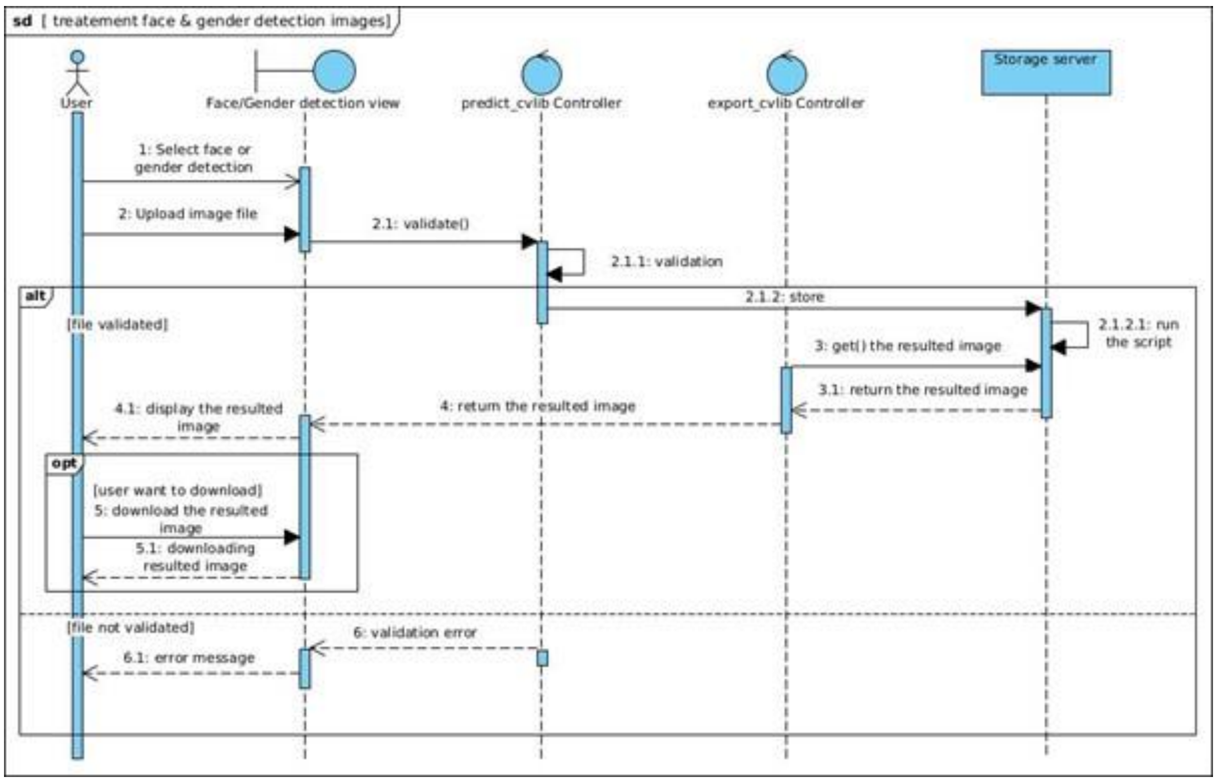

sd [treatement face & gender detection images]
User
Face/Gender detection view
predict_cvlib Controller
export_cvlib Controller
Storage server
1: Select face or gender detection
2: Upload image file
2.1: validate()
2.1.1: validation
alt
[file validated]
2.1.2: store
2.1.2.1: run the script
3: get() the resulted image
3.1: return the resulted image
4: return the resulted image
4.1: display the resulted image
opt
[user want to download]
5: download the resulted image
5.1: downloading resulted image
[file not validated]
6: validation error
6.1: error message

Figure 29 Diagram of the case: face and gender detection processing

Sequence diagram of the case "manage application":

This sequence diagram is the same as the sequence diagram of the case "manage application" in the first solution. (See the figure 30)

2.7.2.3- Deployment diagram

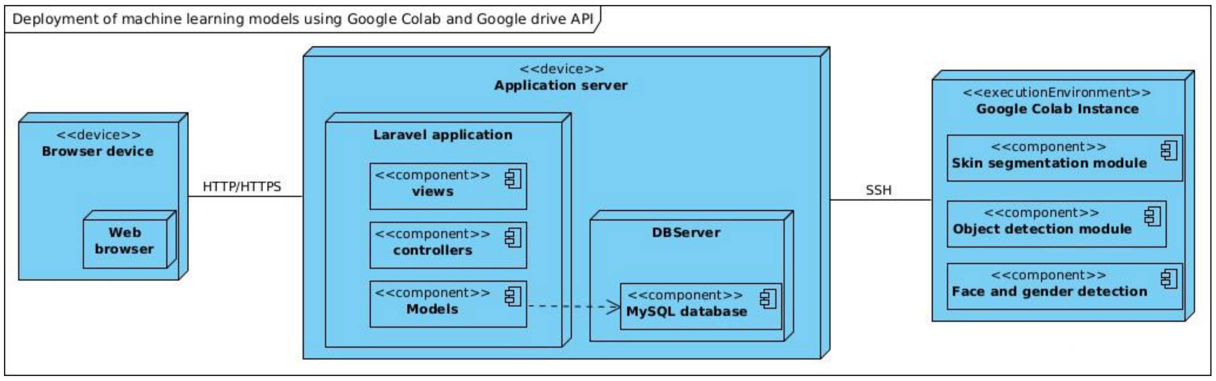


Figure 30 deployment diagram of the second solution

2.7.2.4- The required configurations for deployment:

For the deployment of this solution, we had to make many configurations that we may classify into:

- Frontend configurations
- Backend configurations
- Hosting server configurations

Frontend configurations :

For the frontend, we used the ***Bootstrap*** framework to style the website interfaces. It makes the web applications responsive and mobile-friendly.

***Bootstrap 4.x** integration:*

The installation of ***bootstrap 4*** requires ***nodejs*** to be installed, so for these following steps, we consider that the **npm** command is already installed.

So first, we need to add the ***laravel/ui*** package to the composer environment with the following commands:

```
~$ composer require laravel/ui
```

Then the installation of Bootstrap is done with:

```
~$ php artisan ui bootstrap

~$ php artisan ui bootstrap --auth
```

After the installation, it needs to be compiled with these commands:

```
~$ npm run dev


~$ npm run production
```

After this, ***Boostrap*** is ready to use. All we need is to import it into the project.

Backend configurations:

We have to configure our application to store the uploaded files in the Google Drive folder and make an SSH connection into a Google Colab virtual machine to run the ML model on the backend side.

Google Drive API configuration:

To upload/download files from the web application into ***Google Drive***, we have to use the google drive API.

we start by enabling this API on the desired google account from this URL: **console.cloud.google.com/apis/library/drive.googleapis.com**

Then to configure this API with the ***Laravel*** application, we need to get the ***clientId***, ***clientSecret***, and the ***refreshtoken*** that we can get by creating credentials to our API by following steps:

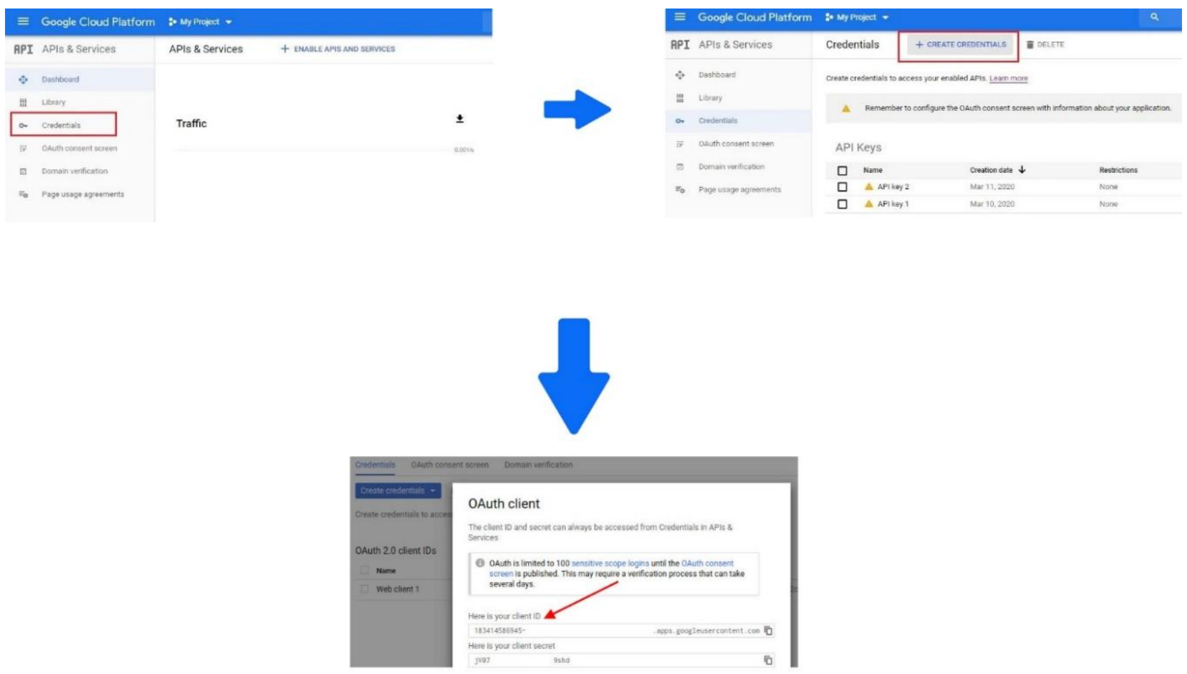
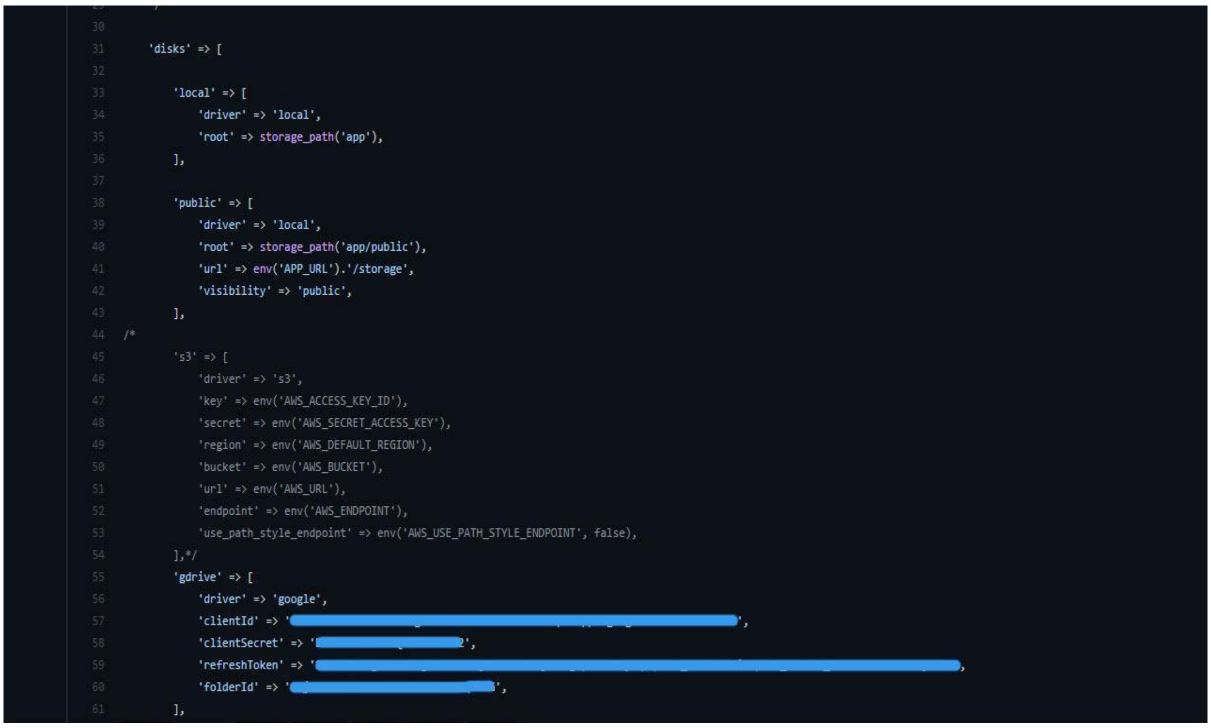

Google Cloud Platform
My Project
APIs & Services
ENABLE APIS AND SERVICES
Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements
Traffic
CREATE CREDENTIALS
DELETE
Create credentials to access your enabled APIs. Learn more
Remember to configure the OAuth consent screen with information about your application.
API Keys
Name
Creation date
Restrictions
API key 2
Mar 11, 2020
None
API key 1
Mar 10, 2020
None
Create credentials
OAuth 2.0 client IDs
Web client 1
OAuth client
The client ID and secret can always be accessed from Credentials in APIs & Services
OAuth is limited to 100 sensitive scope logins until the OAuth consent screen is published. This may require a verification process that can take several days.
Here is your client ID
Here is your client secret

Figure 31 Clientid,clientSecret and refreshtoken generation steps

These steps above permit the generation of the clientID and ClientSecret. The refreshtoken can be generated from https://developers.google.com/oauthplayground.

These three pieces of information are used in our Laravel filesystem configuration file shown below.



```
    'disks' => [

        'local' => [
            'driver' => 'local',
            'root' => storage_path('app'),
        ],

        'public' => [
            'driver' => 'local',
            'root' => storage_path('app/public'),
            'url' => env('APP_URL').'/storage',
            'visibility' => 'public',
        ],
/*
        's3' => [
            'driver' => 's3',
            'key' => env('AWS_ACCESS_KEY_ID'),
            'secret' => env('AWS_SECRET_ACCESS_KEY'),
            'region' => env('AWS_DEFAULT_REGION'),
            'bucket' => env('AWS_BUCKET'),
            'url' => env('AWS_URL'),
            'endpoint' => env('AWS_ENDPOINT'),
            'use_path_style_endpoint' => env('AWS_USE_PATH_STYLE_ENDPOINT', false),
        ],*/
        'gdrive' => [
            'driver' => 'google',
            'clientId' => '[redacted]',
            'clientSecret' => '[redacted]',
            'refreshToken' => '[redacted]',
            'folderId' => '[redacted]',
        ],
```

Figure 32 Laravel configuration file

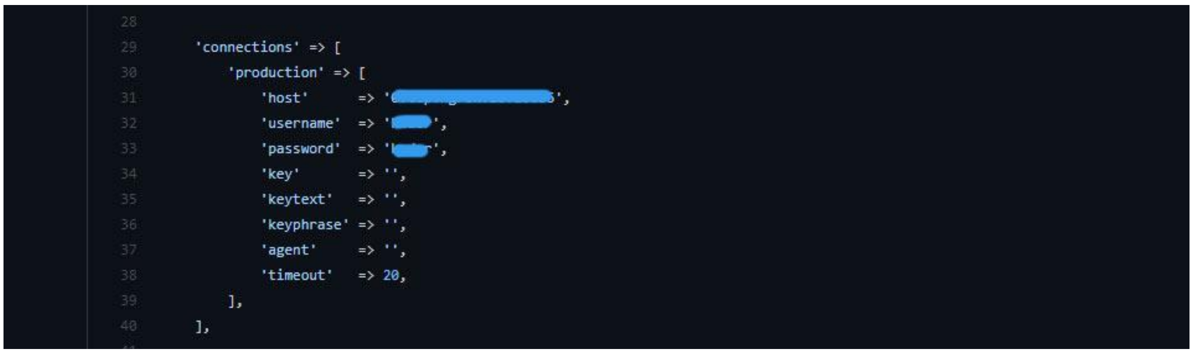
After these steps, the application uses google drive as a filesystem, storing the uploaded files.

SSH tunnel configuration:

The SSH configuration file doest appear in the application files since ***Laravel 5.x***. So to be able to configure a remote connection, an alternative is used, which is the Laravel collectives package that can be installed using the following steps:

```
~$ composer require laravelcollective/remote


~$ php artisan vendor:publish --provider="Collective\Remote\RemoteServiceProvider"
```

After installing the package, we add the virtual machine ***SSH*** accessing information to the ***SSH*** configuration file in ***Laravel*** (see the figure).



```
    'connections' => [
        'production' => [
            'host'      => '[redacted]',
            'username'  => '[redacted]',
            'password'  => '[redacted]',
            'key'       => '',
            'keytext'   => '',
            'keyphrase' => '',
            'agent'     => '',
            'timeout'   => 20,
        ],
    ],
```

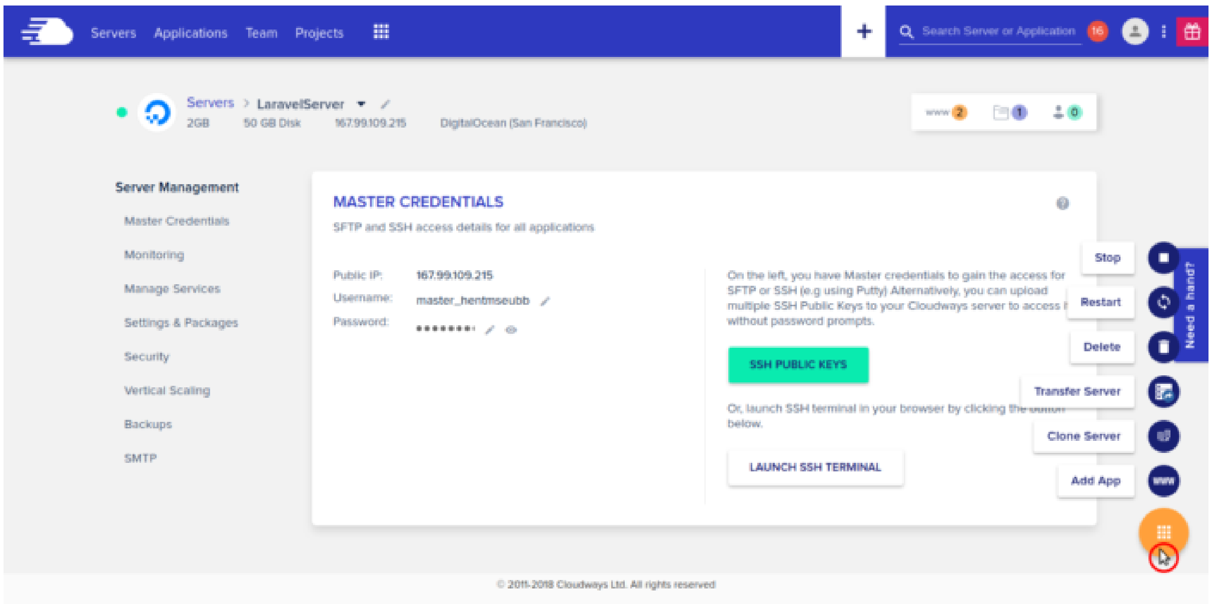
Figure 33 SSH configuration file in Laravel

Hosting server configurations:

We deploy our application using Cloudways hosting service. But before this, we have to push the application into GitHub to ease the deployment task.

So we select a server with 4GB RAM and a two-core processor to ensure that our application runs without problems.

After the server selection, we add our application to this server, as described in the figure below.



2GB 50 GB Disk 167.99.109.215 DigitalOcean (San Francisco)

MASTER CREDENTIALS

SFTP and SSH access details for all applications

Public IP: 167.99.109.215
Username: master_hentmseubb
Password: ••••••••

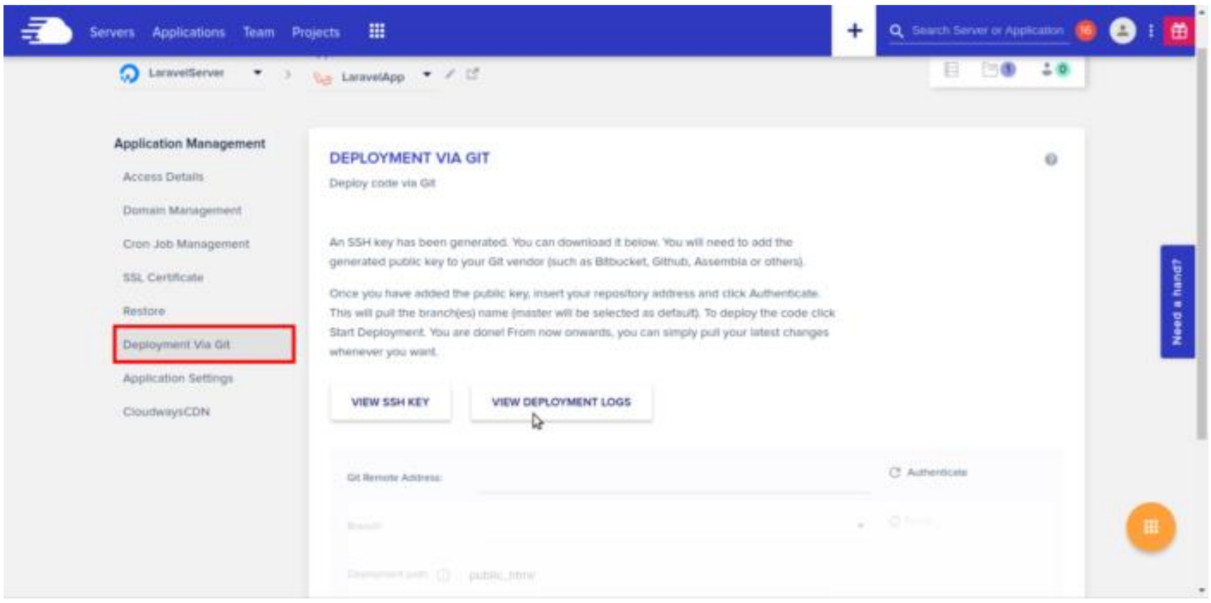
On the left, you have Master credentials to gain the access for SFTP or SSH (e.g using Putty) Alternatively, you can upload multiple SSH Public Keys to your Cloudways server to access it without password prompts.

Or, launch SSH terminal in your browser by clicking the button below.

© 2011-2018 Cloudways Ltd. All rights reserved

Figure 34 Adding the application to the Cloudways server

The reason for pushing our application to GitHub is to import it directly to its reserved space on the Cloudways server using the "deployment from git" option as displayed on this figure.



DEPLOYMENT VIA GIT

Deploy code via Git

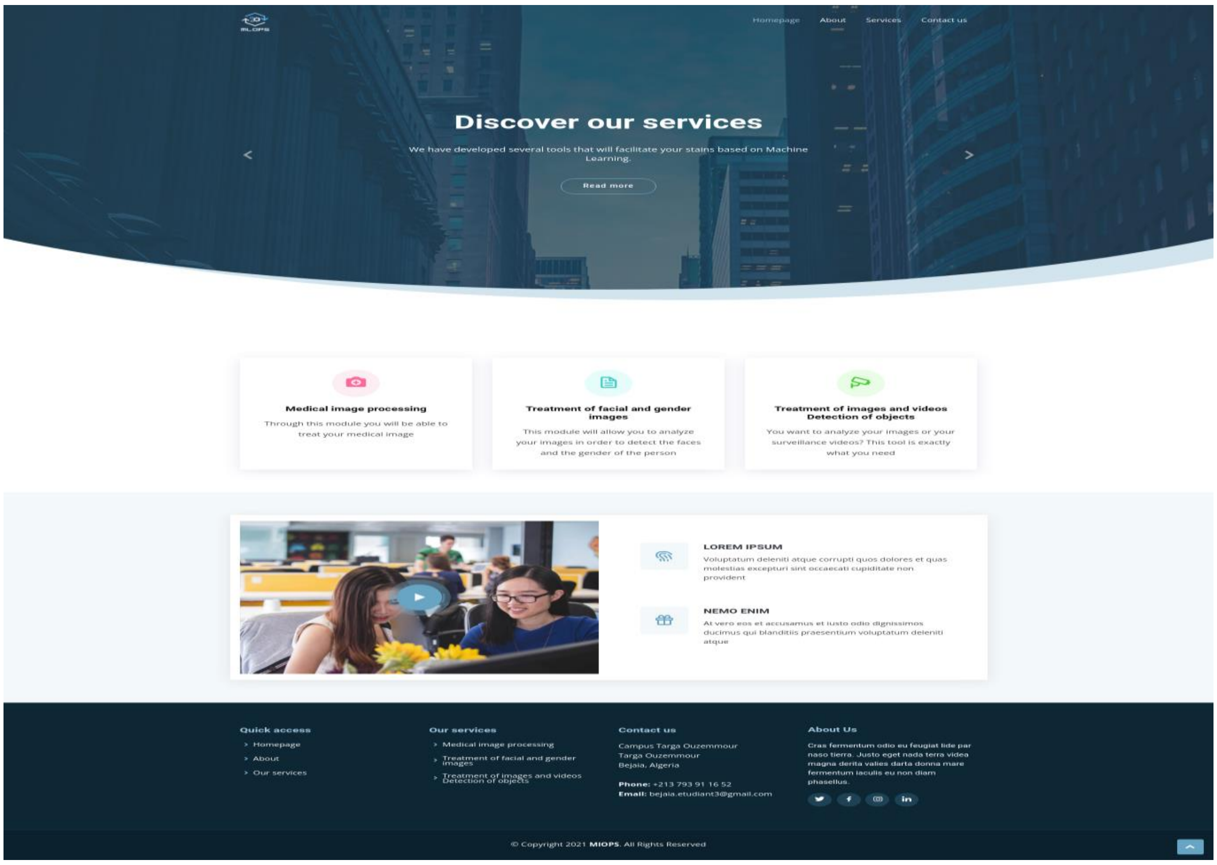
An SSH key has been generated. You can download it below. You will need to add the generated public key to your Git vendor (such as Bitbucket, Github, Assembla or others).

Once you have added the public key, insert your repository address and click Authenticate. This will pull the branch(es) name (master will be selected as default). To deploy the code click Start Deployment. You are done! From now onwards, you can simply pull your latest changes whenever you want.

Figure 35 Application deployment using GIT

2.7.2.5- Application screenshots

In this part, we have some screenshots of the deployed application according to the second solution.



Discover our services

We have developed several tools that will facilitate your stains based on Machine Learning.

Medical image processing
Through this module you will be able to treat your medical image

Treatment of facial and gender images
This module will allow you to analyze your images in order to detect the faces and the gender of the person

Treatment of images and videos Detection of objects
You want to analyze your images or your surveillance videos? This tool is exactly what you need

LOREM IPSUM
Voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident

NEMO ENIM
At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque

Quick access
- Homepage
- About
- Our services

Our services
- Medical image processing
- Treatment of facial and gender images
- Treatment of images and videos Detection of objects

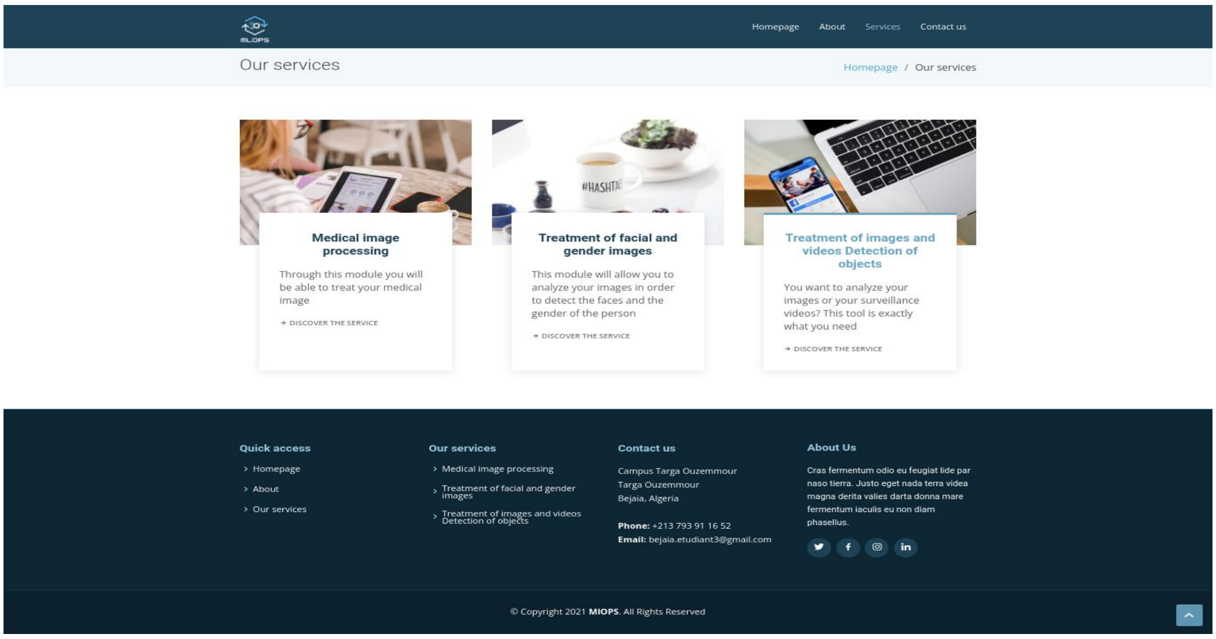
Contact us
Campus Targa Ouzemmour
Targa Ouzemmour
Bejaia, Algeria

Phone: +213 793 91 16 52
Email: bejaia.etudiant3@gmail.com

About Us
Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.

© Copyright 2021 **MIOPS**. All Rights Reserved

Figure 36 Homepage screenshot



Our services

Homepage / Our services

Medical image processing

Through this module you will be able to treat your medical image

→ DISCOVER THE SERVICE

Treatment of facial and gender images

This module will allow you to analyze your images in order to detect the faces and the gender of the person

→ DISCOVER THE SERVICE

Treatment of images and videos Detection of objects

You want to analyze your images or your surveillance videos? This tool is exactly what you need

→ DISCOVER THE SERVICE

Quick access

- Homepage
- About
- Our services

Our services

- Medical image processing
- Treatment of facial and gender images
- Treatment of images and videos Detection of objects

Contact us

Campus Targa Ouzemmour
Targa Ouzemmour
Bejaia, Algeria

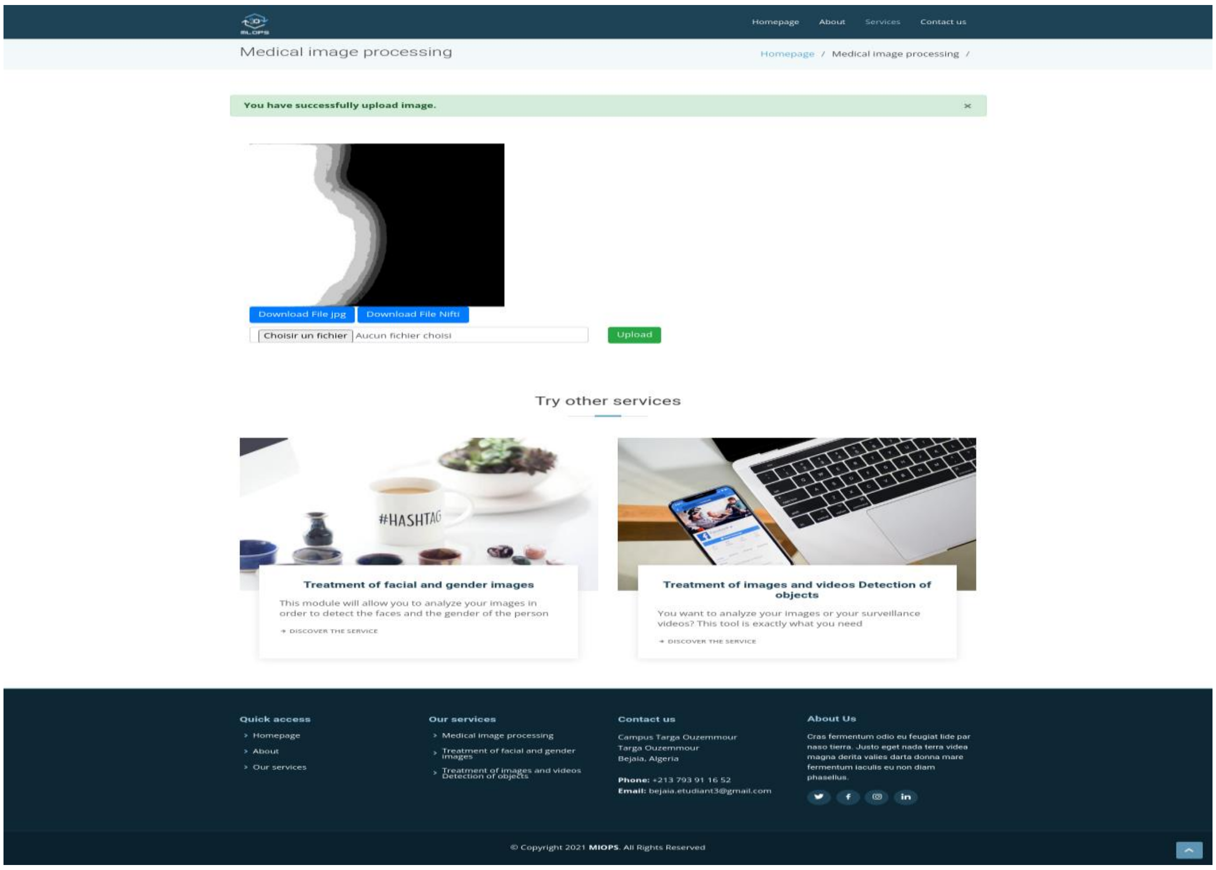
Phone: +213 793 91 16 52
Email: bejaia.etudiant3@gmail.com

About Us

Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.

© Copyright 2021 **MIOPS**. All Rights Reserved

Figure 37 Services webpage screenshot



Medical image processing

Homepage / Medical image processing /

You have successfully upload image.

Try other services

Treatment of facial and gender images

This module will allow you to analyze your images in order to detect the faces and the gender of the person

→ DISCOVER THE SERVICE

Treatment of images and videos Detection of objects

You want to analyze your images or your surveillance videos? This tool is exactly what you need

→ DISCOVER THE SERVICE

Quick access

- Homepage
- About
- Our services

Our services

- Medical image processing
- Treatment of facial and gender images
- Treatment of images and videos Detection of objects

Contact us

Campus Targa Ouzemmour
Targa Ouzemmour
Bejaia, Algeria

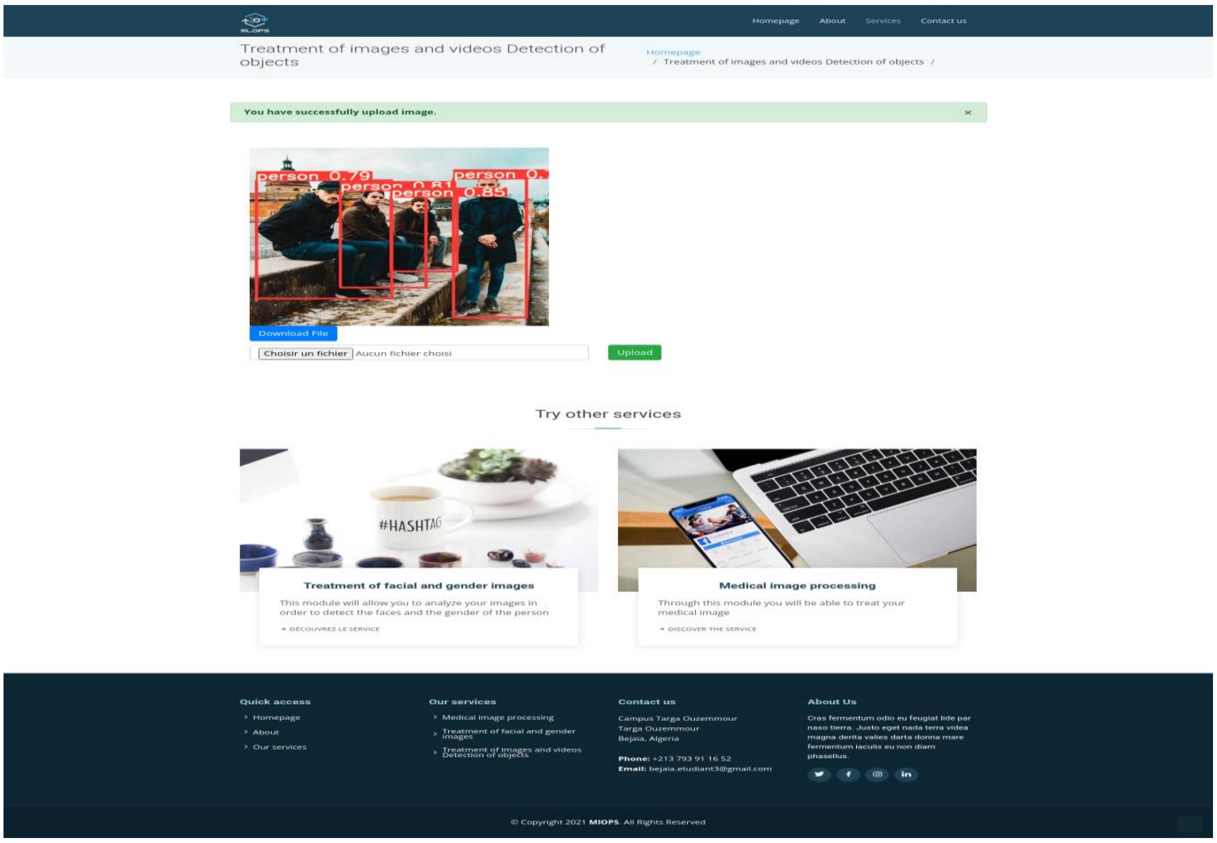
Phone: +213 793 91 16 52
Email: bejaia.etudiant3@gmail.com

About Us

Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.

© Copyright 2021 **MIOPS**. All Rights Reserved

Figure 38 Skin segmentation screenshot



Homepage About Services Contact us

Treatment of images and videos Detection of objects

Homepage / Treatment of images and videos Detection of objects /

You have successfully upload image. ×

Download File

Choisir un fichier Aucun fichier choisi Upload

Try other services

Treatment of facial and gender images

This module will allow you to analyze your images in order to detect the faces and the gender of the person

→ DÉCOUVREZ LE SERVICE

Medical image processing

Through this module you will be able to treat your medical image

→ DISCOVER THE SERVICE

Quick access
- Homepage
- About
- Our services

Our services
- Medical image processing
- Treatment of facial and gender images
- Treatment of images and videos Detection of objects

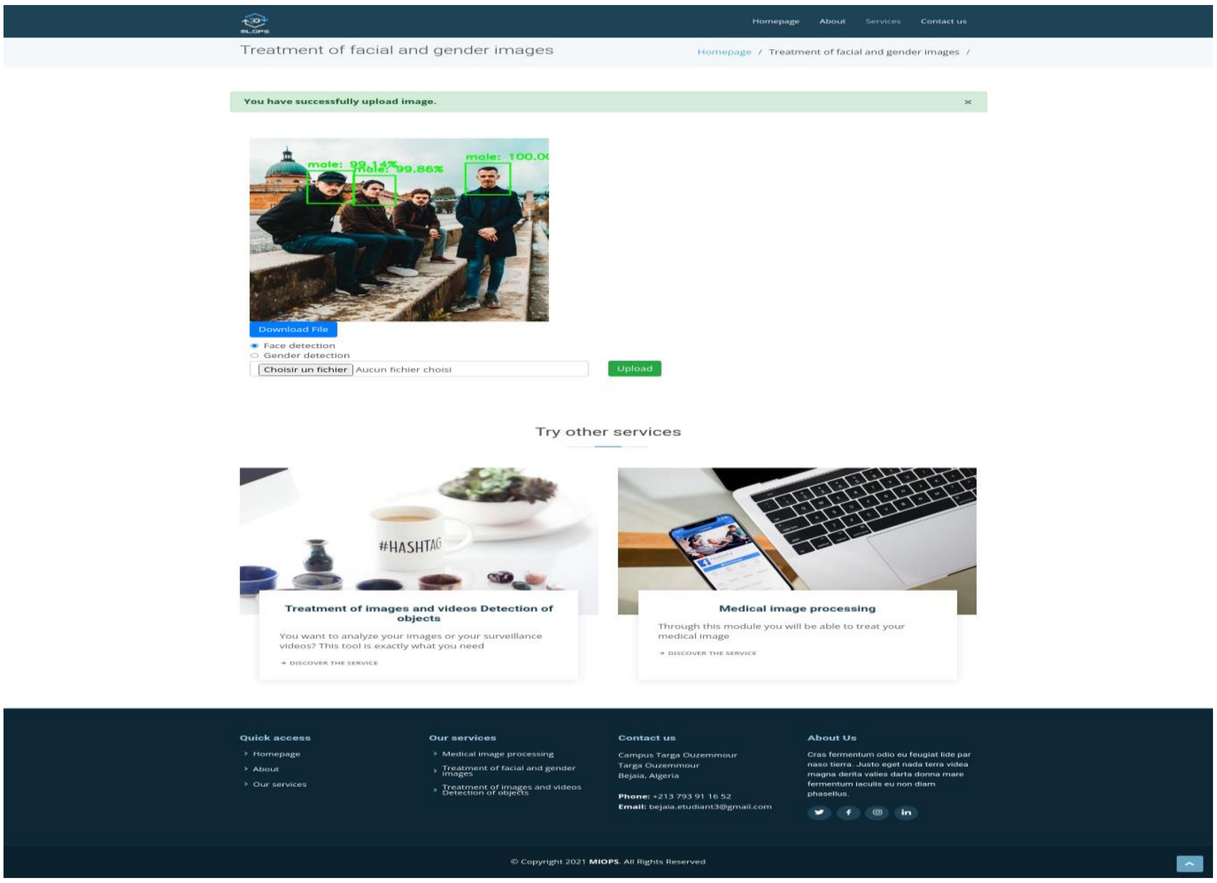
Contact us
Campus Targa Ouzemmour
Targa Ouzemmour
Bejaia, Algeria

Phone: +213 793 91 16 52
Email: bejaia.etudiant3@gmail.com

About Us
Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.

© Copyright 2021 **MIOPS**. All Rights Reserved

Figure 39 Object detection webpage screenshot



Homepage About Services Contact us

Treatment of facial and gender images

Homepage / Treatment of facial and gender images /

You have successfully upload image. ×

Download File

● Face detection
○ Gender detection

Choisir un fichier Aucun fichier choisi Upload

Try other services

Treatment of images and videos Detection of objects

You want to analyze your images or your surveillance videos? This tool is exactly what you need

→ DISCOVER THE SERVICE

Medical image processing

Through this module you will be able to treat your medical image

→ DISCOVER THE SERVICE

Quick access
- Homepage
- About
- Our services

Our services
- Medical image processing
- Treatment of facial and gender images
- Treatment of images and videos Detection of objects

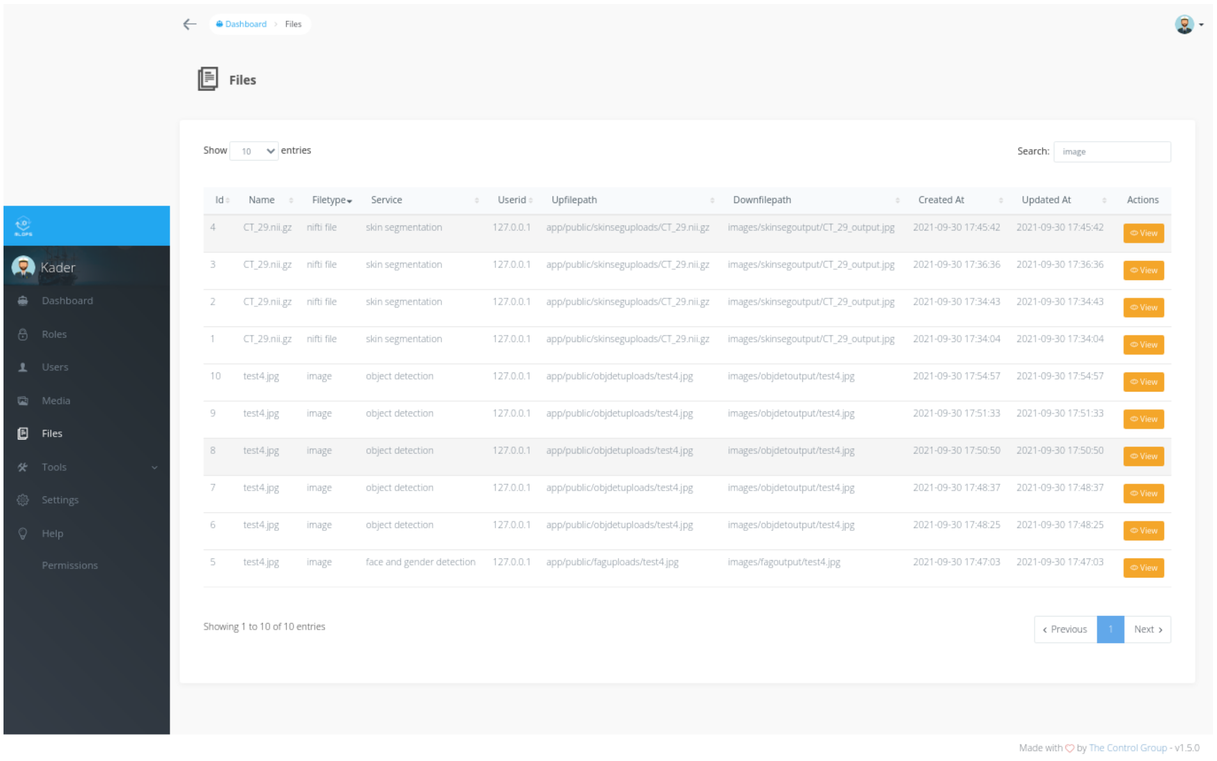
Contact us
Campus Targa Ouzemmour
Targa Ouzemmour
Bejaia, Algeria

Phone: +213 793 91 16 52
Email: bejaia.etudiant3@gmail.com

About Us
Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.

© Copyright 2021 **MIOPS**. All Rights Reserved

Figure 40 Face and gender detection screenshot



Files

Show 10 entries Search: image

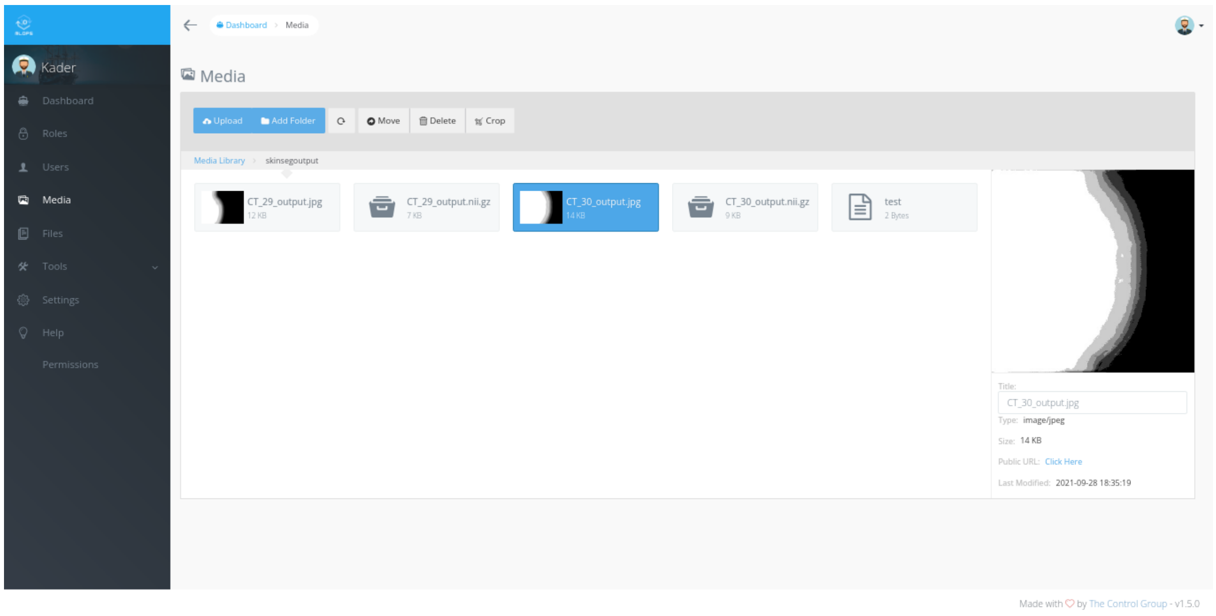
Id	Name	Filetype	Service	Userid	Upfilepath	Downfilepath	Created At	Updated At	Actions
4	CT_29.nii.gz	nifti file	skin segmentation	127.0.0.1	app/public/skinseguploads/CT_29.nii.gz	images/skinsegoutput/CT_29_output.jpg	2021-09-30 17:45:42	2021-09-30 17:45:42	View
3	CT_29.nii.gz	nifti file	skin segmentation	127.0.0.1	app/public/skinseguploads/CT_29.nii.gz	images/skinsegoutput/CT_29_output.jpg	2021-09-30 17:36:36	2021-09-30 17:36:36	View
2	CT_29.nii.gz	nifti file	skin segmentation	127.0.0.1	app/public/skinseguploads/CT_29.nii.gz	images/skinsegoutput/CT_29_output.jpg	2021-09-30 17:34:43	2021-09-30 17:34:43	View
1	CT_29.nii.gz	nifti file	skin segmentation	127.0.0.1	app/public/skinseguploads/CT_29.nii.gz	images/skinsegoutput/CT_29_output.jpg	2021-09-30 17:34:04	2021-09-30 17:34:04	View
10	test4.jpg	image	object detection	127.0.0.1	app/public/objdetuploads/test4.jpg	images/objdetoutput/test4.jpg	2021-09-30 17:54:57	2021-09-30 17:54:57	View
9	test4.jpg	image	object detection	127.0.0.1	app/public/objdetuploads/test4.jpg	images/objdetoutput/test4.jpg	2021-09-30 17:51:33	2021-09-30 17:51:33	View
8	test4.jpg	image	object detection	127.0.0.1	app/public/objdetuploads/test4.jpg	images/objdetoutput/test4.jpg	2021-09-30 17:50:50	2021-09-30 17:50:50	View
7	test4.jpg	image	object detection	127.0.0.1	app/public/objdetuploads/test4.jpg	images/objdetoutput/test4.jpg	2021-09-30 17:48:37	2021-09-30 17:48:37	View
6	test4.jpg	image	object detection	127.0.0.1	app/public/objdetuploads/test4.jpg	images/objdetoutput/test4.jpg	2021-09-30 17:48:25	2021-09-30 17:48:25	View
5	test4.jpg	image	face and gender detection	127.0.0.1	app/public/faguploads/test4.jpg	images/fagoutput/test4.jpg	2021-09-30 17:47:03	2021-09-30 17:47:03	View

Showing 1 to 10 of 10 entries

‹ Previous 1 Next ›

Made with ♡ by The Control Group - v1.5.0

Figure 41 Dashboard files list screenshot



Title: CT_30_output.jpg
Type: image/jpeg
Size: 14 KB
Public URL: Click Here
Last Modified: 2021-09-28 18:35:19

Made with ♡ by The Control Group - v1.5.0

Figure 42 Dashboard files display screenshot

2.8- Solutions comparison

We used two different solutions based on various programming languages and frameworks because we wanted to replicate our first thought while getting the project idea from our supervisor.

From the beginning, we considered using an existing virtual machine, which could be a good solution if data preservation is not a priority. Then we got an idea of developing our virtual machine, which can be a good solution if data preservation is a priority.

At the end, we made a small comparison between the two solutions, and we got these results:

Solution	advantages	inconvenient
Solution 1	The uploading and downloading time are fast.	The prediction time is slow.
	The deployment cost of the solution may be handled as needed.	He has to add configurations to install the needed ML libraries.
	The data are saved and between our hands since it's a costume VM	
Solution 2	Separate the display and the processing servers	The connection between the servers may be slow as they are not in the same location
	Possibility to use the exact data for other applications due to google drive API.	The deployment of this solution may be expensive for meaningful use.
	Google Colab environment contains almost all the libraries used in ML, so compatibility problems rarely occur during the execution.	the data are sored my google, so they have the right to use theme since we agreed on the Terms, Data Policy

2.9- Chapter conclusion:

This chapter gave us a vision of our work and gave the conceptual aspect of the application through the different diagrams described in UML. A diagram illustrating the different use cases of sequence diagrams illustrates the visualization process—cases of use.

We have also described the steps and configurations necessary for implementing the proposed solutions and compared the two solutions.

In the next chapter, we will discuss the different materials used during this developement and deployment process.

Chapter 3

Materials used in this project

3.1- Introduction:

In this project, we focused on the deployment of some machine learning models for web applications.

For this, we used many tools that are reviewed in this chapter. We can classify them into different categories, which are detailed below:

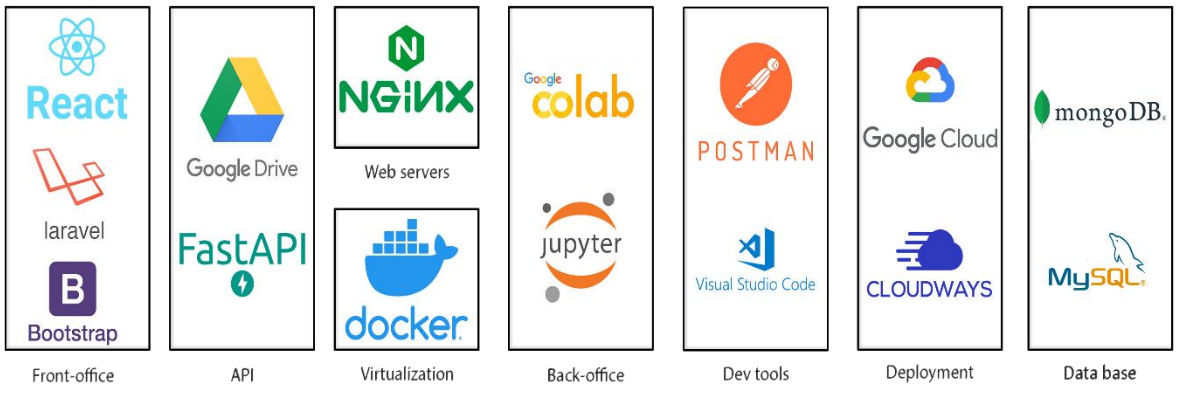


Figure 43 The different tools used in this project

3.2- Front office

Our project consists in making life easier for the user by using different services. Is this could not be possible without an ergonomic user interface. That's why we have developed using the materials explained below:

There is a multitude of tools. We have chosen the following ones, and the motivation is presented for each one.

3.2.1- ReactJS

React (also known as React.js or ReactJS) is a free frontend JavaScript toolkit for building user interfaces and components. Several developer communities use it, including social networking giant Facebook.

React can be used to create single-page or mobile apps as a foundation. Because it only manages data and renders in the DOM (Document Object Model), building React apps typically demands the use of other frameworks for routing and client-side functionality.

According to the official ReactJS description, it promotes the development of reusable user interface components that take advantage of dynamic data.

The MVC model is frequently referred to as React (Model-View-Controller), where React ensures the view. since it isolates the DOM, resulting in a more straightforward programming approach and improved speed. [8] [9]

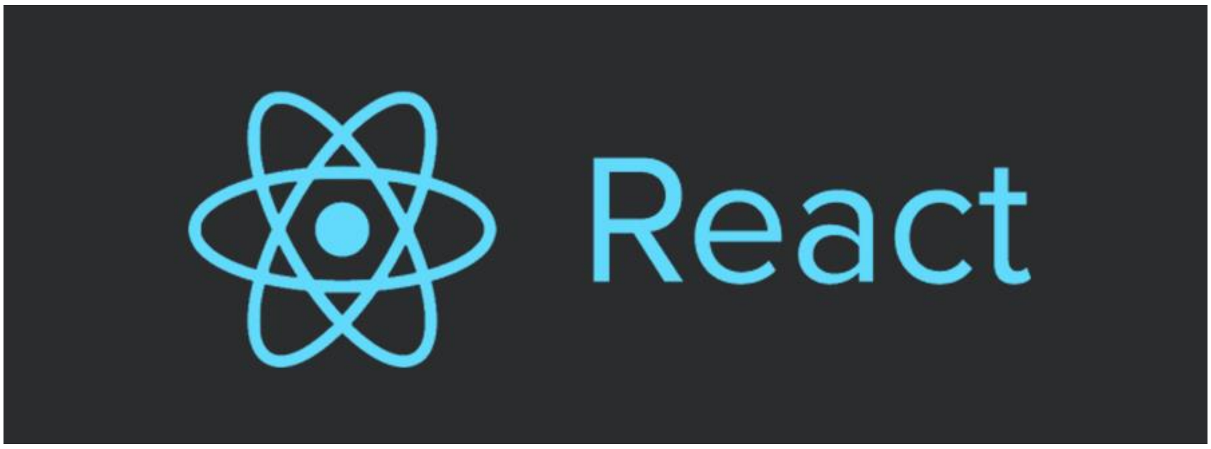


Figure 44 React logo

It may also render on the server using an API and power native apps using React Native. Compared to traditional data binding, React creates a one-way reactive data flow, which eliminates mistakes and simplifies reasoning.

3.2.1.1- Advantages of React

- The virtual DOM is a JavaScript object that is used. Because the JavaScript virtual DOM is quicker than the regular DOM, this enhances application speed.
- It works on both the client and server sides, as well as with other frameworks.
- Components and data models increase readability, allowing for the maintenance of more significant programs.
- The material can be cited.
- ReactJS is a lightning-fast framework.
- Components dominate web development in the future.
- ReactJS quickly gained popularity among JS developers.
- Because Intelligibility generates "clean" (easy-to-read) code, reading it reveals the program's functionality right away.
- ReactJS uses a unique syntax known as JSX, which allows HTML and JavaScript to be mixed. Although it is not compulsory, it is strongly recommended to try this new syntax to make writing components much more accessible. [10] [11]

3.2.2- Laravel

Laravel is a web application framework with a syntax that is both expressive and elegant. To be meaningful, we believe development must be a fun and creative experience. Laravel aims to simplify development by simplifying typical chores seen in most online applications, such as authentication, routing, sessions and caching.

This framework includes the best libraries helpful in creating a website. In addition, the excellent framework also integrates many other exclusive features. It is particularly true of its Blade template engine.

Laravel has its template blade system that it uses. As a result, it has been viewed with the blade.php extension, and the contents of PHP variables may be shown. In addition, it's important to note that such a material uses an MVC design for its view controller model. This framework may be used to create forms and layouts. These layouts serve as templates for HTML pages. With Blade, you can quickly and easily provide various instructions, including conditional and iterative instructions. They have a high level of efficiency when it comes to code management.

Laravel, on the other hand, provides a high level of security through its operation. Among other things, the created forms generate tokens that prevent several attacks, such as CSRF attacks (Cross-Site Request Forgery), which involves circumventing a site's authentication to carry out malicious activities that can be sent through a form.

Furthermore, the framework has a routing system that is compatible with all HTTP methods. [12] [13]

Figure 45 Laravel logo

3.2.2.1- Advantages of Laravel

- Interfaces, overloading, shorter array syntax, Namespaces, and Anonymous functions are just a few of the new features available in PHP.
- The documentation is meant to make Laravel developer-friendly.
- The framework provides an API instead of utilizing the SwiftMailer library. It also contains SMTP, Mailgun, SparkPost, Mandrill, PHP's "mail" function, Amazon SES, and "send email" drivers, allowing you to send emails using local or cloud-based services.
- Backends for standard caches are supported.
- Packages and the availability of resources are two important factors to consider.
- It gives an easy and beneficial ActiveRecord implementation for running with your database. Put differently, and it intends that the models you build in the MVC have a corresponding table in the database. [14]

3.2.3- Bootstrap 4

Bootstrap is a frontend programming framework for building websites and online apps that is free and open source. It uses HTML, CSS, and JavaScript (JS) to create responsive, mobile-first websites and apps.

The mobile-first approach assumes that smartphones, tablets, and task-specific devices can access a website or app. Responsive design allows a website or app to detect the visitor's screen size and orientation and adapt the display accordingly; the mobile-first approach assumes that smartphones, tablets, and task-specific devices are used to access a website or app. Employees' primary materials for getting work done are mobile applications, which satisfy the design needs of such technologies.

Along with the framework for implementation, Bootstrap contains user interface components, layouts, and JS materials. Pre-compiled or source code versions of the program are available. [15] [16]

Figure 46 Bootstrap logo

3.2.3.1- Advantages of Bootstrap

- Bootstrap is Basic to Use: Bootstrap is an effortless and easy to use framework for designing and developing websites. There is a lot to learn from this framework because it is so fresh. It may be utilized with CSS, LESS, or SaaS, among other things.
- It's Simple to Form a Partnership: Bootstrap is a framework that can be simply linked with other frameworks without disrupting current or new sites.
- Bootstrap is a quick and time-saving framework. It is an agile framework that is significantly faster than other frameworks. Because of its standard ready-made code blocks, responsiveness, and cross-browser features, it saves time. [17] [18]

3.3- Virtual machine management (Container Management)

A virtual machine is a computer-generated illusion created by emulation software and installed on a computer. In this scenario, we engaged:

3.3.1- Docker

Docker automates tedious configuration chores and is utilized across the development lifecycle for quick, simple, portable desktop and cloud application development. Docker's end-to-end platform comprises user interfaces, command line interfaces, APIs, and security features designed to operate together across the application development lifecycle. Docker has over 318 billion container image downloads. Container-based apps are popular and easy to consume and publish, with millions of applications accessible on Docker Hub.

Docker leverages the Linux kernel and technologies such as Cgroups and namespaces to separate processes and allow them to operate independently. The ability to execute many processes and programs separately to make greater use of your infrastructure while maintaining the security you'd get with separate systems is the goal of containers.

Container technologies, such as Docker, allow for image-based deployment. It makes it simple to share an application, or a collection of services, across various contexts, along with all of its dependencies. Docker also automates the deployment of a program (or a collection of processes that make up an app) within this container environment.

Users have unparalleled access to programs, the ability to deploy swiftly, and control over versions and version distribution thanks to these technologies built on top of Linux containers, making Docker user-friendly and unique.

Docker is a technology that allows you to distribute your software together with all of its dependencies.

Within the Docker ecosystem, images play an essential role. They are considered a collection of all the components (libraries, files and other stuff) that software needs to execute. [19] [20]

Figure 47 Docker logo

3.3.1.1- Advantages of Docker

- Cost-effectiveness with rapid deployment in a consistent and isolated environment.
- The ability to run anywhere is known as mobility. Docker makes any deployment consistent, movable (portable), and scalable. Containers have the added benefit of running anywhere.
- Test for repeatability and automation, as well as rollback and deployment flexibility.
- Collaboration, modularity, and scalability are all critical factors to consider.
- Consistent and Isolated Environment. [21] [22]

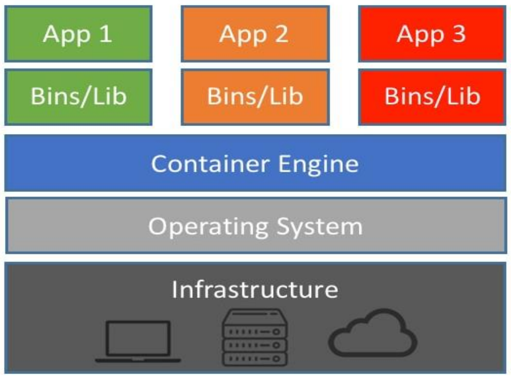


Figure 48 Docker Containerization Technology for DevOps

3.4- HTTP Webserver

A web server is either software or a computer server that answers World Wide Web queries across a public or private network, mainly utilizing the HTTP protocol. in our case, we used:

3.4.1- Nginx

NGINX, pronounced "engine-ex," is an open-source web server that is now being utilized as a reverse proxy, HTTP cache, and load balancer, thanks to its early popularity as a web server.

Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple, Intel, and many other well-known businesses utilize NGINX (source). It was designed by Igor Sysoev and initially released to the public in October 2004. Igor created the program to solve the C10k problem, which is a performance issue involving 10,000 concurrent connections. NGINX typically outperforms other popular web servers in benchmark testing due to its roots in performance optimization at scale, especially in scenarios with static content and large concurrent requests, which is why Kinsta utilizes it to power its hosting. This web server is designed to have a low memory footprint and a high concurrency level, rather than creating new processes for each Web request.

With NGINX, a single process master may manage many worker processes. The master oversees the worker's processes while the workers carry out the treatment as directed. Because it is asynchronous, the worker may process each request simultaneously without interfering with other requests. It is an event-driven approach in which all requests are handled in a single thread. [23]

Figure 49 Nginx logo

3.4.1.1- Advantages of NGINX

- The setup and configuration are basic and straightforward.
- It's a quick and effective method for providing static files.
- When compared to Apache, there are four times as many concurrent connections handled.
- Compatibility with widely used web applications
- Support for Load Balancing
- Nginx accelerates websites, allowing them to get higher Google rankings. [24]

3.5- Application Programming Interface (API)

An application programming interface (API) is a standardized collection of classes, methods, functions, and constants that act as a front end for software to deliver services to other software. In our situation, we employed:

3.5.1- FastApi

FastAPI is a modern and robust web framework for creating APIs using standard Python type indices in Python 3.6 and above. FastAPI is a Starlette subclass that is built on asyncio. The bulk of the features are recognizable to you if you're familiar with or have used Starlette before. Fast API, similar to Nodejs, promises to be one of the fastest web frameworks accessible. [25]

Figure 50 FastAPI logo

3.5.1.1- Advantages of FastAPI

- Fast: On par with NodeJS and Go in terms of performance (thanks to Starlette and Pydantic). One of the quickest Python frameworks on the market.
- Increase the pace with which features are developed by 200 per cent to 300 per cent.
- Fewer bugs: Reduce human (developer)-caused mistakes by roughly 40%.
- Great editor support. Intuitive: Everywhere, there is completion. Debugging takes less time.
- Simple: Designed to be simple to use and understand and less time spent reading documents.
- In a nutshell, reduce code duplication. Each parameter declaration has several characteristics. There are few bugs.
- Robust: Get code that is ready for production. With interactive documentation that is generated automatically.
- Standards-based: Based on (and completely compatible with) the OpenAPI (formerly known as Swagger) and JSON Schema open API standards. [26] [27]

3.5.2- Google Drive

Google Drive is a Google-developed file storage and syncing service, allows users to store files in the cloud (on Google's servers), sync files across devices, and share files. Google Drive offers programs for Windows and macOS PCs, as well as Android and iOS smartphones and tablets, in addition to a web interface.

Figure 51 Google drive logo

The Google Drive API allows us to create apps that take full use of Google's cloud storage. So we may use this API to connect our app to Google Drive and access and save files. [28]

The link between your Google Drive app, Google Drive, and the Google Drive API is depicted in the figure below:

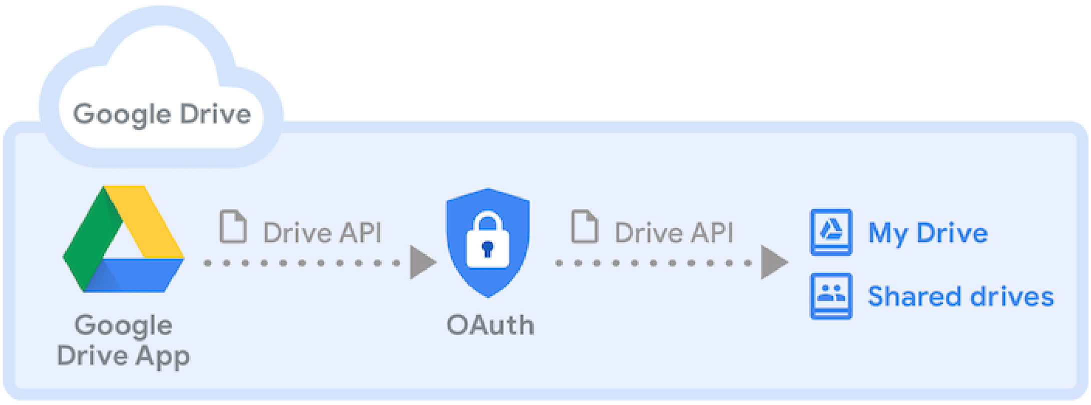


Figure 52 Relationship between Google Drive app, Google Drive, and Google Drive API.

3.5.2.1- Advantages of Google Drive

- Streaming File Drive, by enabling direct access to Drive files stored in the cloud, you may save space on your hard drive and network bandwidth. As a result, you'll always have the most recent version of the papers.
- Shared drives, the files in a shared Drive belong to the team rather than to a single person, ensuring that all team members have constant access to the information they want.
- Drive's data loss prevention (DLP) feature may identify files containing sensitive information and block access to anybody who isn't a member of your company.
- Offline access, changes made offline are automatically synchronized as soon as your device is connected to the Internet.

3.6- Back Office

The back office collects support, control, and administrative activities carried out within an organization, so to manage our VMs files manually, we deployed:

3.6.1- Jupyter

The Jupyter project (for Julia, Python, and R) is the outcome of the IPython project, which consists of an advanced Python interpreter that allows you to increase the productivity of your Python code. IPython was progressively expanded with Notebooks, a JSON-based online interface that allows users to execute a Python kernel and code directly in the browser, with intermediate results shown. It was a significant step forward in terms of collaboration and "interactive" creation.

IPython is python-oriented, as the name implies, and the solution's creators rapidly discovered that Python was only one of many potential languages. To develop directly in this language, we need to add a kernel connected with another language (R or Julia). As a result, the Jupyter project and its Jupyter Notebooks were born (the Notebooks files' names are *.ipynb files, derived from IPython Notebooks). The IPython project is still alive and well, but it focuses solely on Python, with the Notebook component removed.

Overwhelmed by their success, the Jupyter Notebooks developers had to respond to a slew of development requests, and the notebooks quickly evolved to include notebook extensions (which I'll discuss in more detail in a later article), as well as a new and more advanced development interface: the JupyterLab! [29]

Figure 53 Jupyter logo

3.6.1.1- Advantages of Jupyter

- Users are welcome to participate (working with notebooks, code and data).
- JupyterLab allows you to integrate notebooks, documents, and activities to a great degree.
- Drag and drop notebook cells to reorganize them and copy them across notebooks.
- Interactively run code chunks from text files (.py, .R, .md, .tex, etc.).
- Connect a coding terminal to a notebook kernel to communicate with code while keeping the notebook clean. [30]

3.6.2- Google Colab

Colaboratory, or 'Colab' for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary Python code through the browser and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup while providing free access to computing resources, including GPUs.

We used this service by creating a virtual machine that is accessed from the main application using an SSH tunnel then execute the machine learning model using the specific data uploaded by the user. [31]

Figure 54 Google colaboratory

3.6.2.1- Advantages of Google Colab

- You can simply share your Google Colab notes.
- With versioning, you can easily save your notebook to Github with a single click.
- Google Colab provides a beautiful collection of code snippets that you can copy and paste into your code.
- Google Colab provides a beautiful collection of code snippets that you can simply put into your code.
- In the case of non-technical users, Python is beneficial for nearly everyone in an office job, not only programmers, who must evaluate data.
- Python programs demand a lot of computer power and might take a long time to run. You won't have to worry about anything if you execute your scripts on the cloud. While running Python programs on your local system, the performance of your machine does not suffer.
- The best part is that it is entirely free. [32]

3.7- Database

In simple words, data can be facts about any object under investigation. Databases are necessary in our case because we deal with files and users. In our circumstance, we used the following:

3.7.1- MySQL

MySQL is a free and open-source relational database server. Instead of storing all of the data in a single table, a database server stores the data in several tables. It improves the overall speed and consistency. Tables are linked together by defined relationships, allowing data to be combined from several tables within a single query. The SQL in "MySQL" stands for "Structured Query Language," a standard language for dealing with databases.

Figure 55 MySQL logo

3.7.1.1- Advantages of MySQL

- MySQL is a multi-user database.
- It may be used with other programming languages.
- You don't need a lot of RAM.
- A portable software
- The table's structure is adaptable.
- Open Source SGBD
- It is possible to use it with low material specifications.
- Comprehensive administrative tools
- Variable data types
- Guaranteed security is number ten. [33]

3.7.2- MongoDB

MongoDB is cross-platform document-oriented database software that is open source. MongoDB is a NoSQL database software that stores data as flexible JSON documents. It means that the fields can change from one document to the next, and the data structure can change over time. [34]

Figure 56 MongoDB logo

3.7.2.1- Advantages of MongoDB

- The document model maps the objects in your application's code, making data processing easier.
- Ad hoc requests, indexing, and real-time aggregation provide potent tools for accessing and analyzing your data.
- MongoDB is a distributed database with high availability, horizontal scaling, and geographic dispersion, all of which are integrated and simple to use.
- MongoDB is entirely free to use. All versions published after October 16, 2018, including corrections for previous versions, are licensed under the Server Side Public License (SSPL) v1. [35]

3.8- Deployment

The term "software deployment" refers to the activities that go into making a software system available. In general, the deployment process consists of several interdependent activities with probable transitions between them; for this need, we used the following servers:

3.8.1- Google Cloud Platform

The Google Cloud Platform is a grouping of Google's public cloud computing services. The platform provides a range of hosted services for compute, storage, and application development run on Google infrastructure. Google Cloud Platform services are available to software developers, cloud administrators, and other corporate IT experts through the public Internet or a customized network connection.

Google Cloud Platform offers services such as computing, storage, networking, big data, machine learning, the Internet of things (IoT), and cloud management, security, and developer materials. Among the effective cloud computing solutions provided by Google Cloud Platform are:

- Google Compute Engine is an infrastructure-as-a-service (IaaS) provider that provides virtual machine instances to customers for workload hosting.

- Google App Engine is a platform-as-a-service (PaaS) offering by Google that provides scalable hosting for software developers. Developers may also utilize a software development kit (SDK) to create software applications that run on App Engine.
- Google Cloud Storage is a cloud storing system that allows for the storage of massive volumes of unstructured data. Google also offers database storage services, including Cloud Datastore for NoSQL non-relational storage, Cloud SQL for MySQL fully relational storage, and Google's proprietary Cloud Bigtable database.
- Google Container Engine is a Docker container management and orchestration system that runs in Google Cloud. Google Container Engine is built on Google Kubernetes, a container orchestration engine.

Figure 57 Google Cloud Platform logo

3.8.1.1- Advantages of Google Cloud Platform

- Pricing that is lower than competitors'
- Live Migration of Virtual Machines over a Private Global Fiber Network
- Improved Performance Cutting-Edge Security Commitment to Continued Expansion
- Redundant backups [36]

3.8.1.2- Advantages of Google Compute Engine

- Across all regions, Compute Engine's network input/output is far faster than AWS's.
- Efficient Block Storage's persistent disks have a storage capacity of 257 TB.
- It delivers more dependable services because of its ability to facilitate live VM movement across hosts.
- GCP is equipped with a robust, redundant backup system. This platform provides the foundation for Google's essential products, including the Search Engine and Gmail.

- For more than two decades, Google has been around. When you use Google Cloud Platform, you get access to the security measures that Google has built over the years to secure its powerful products like Google Search and Gmail. [37] [38]

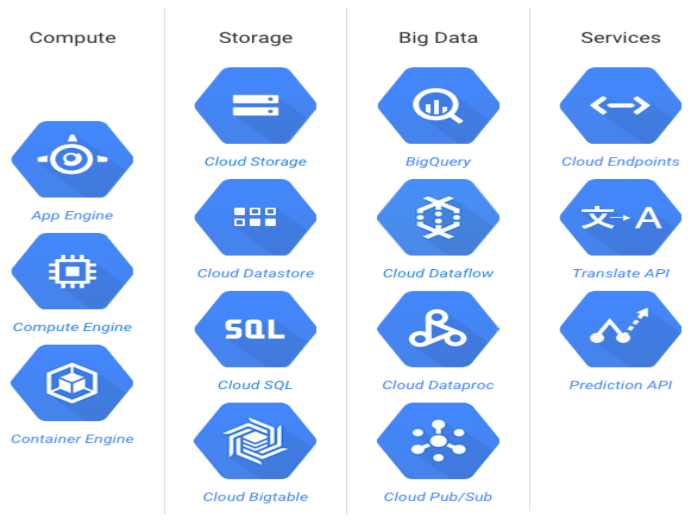


Figure 58 Google Compute Engine Services

3.8.2- Cloudways [39]

Cloudways is a well-known platform as a service provider that enables developers to build, grow, and manage apps rapidly. This platform supports various programming languages, including Java, Ruby, PHP, Node.js, Python, Scala, and Clojure.

Cloudways charges clients based on the number of virtual machines required for their apps.

CLOUDWAYS

Figure 59 Cloudways logo

3.8.2.1- Advantages of Cloudways

- Online assistance is provided for free, with possible add-ons to improve your service.
- With pay-as-you-go billing, you only pay for the resources you use.
- Your first website is migrated to Cloudways for free.
- SSL certificates are simple to install and are available for free.
- Unrestricted access to services without the need to sign a contract.
- On every subscription, you have the option of hosting an infinite number of applications.
- Lets developers concentrate on coding and not server management

- It offers developers a secure way of developing applications due to its set of security features. [40]

3.9- Programming Tools

The processes that go into making a software system available are referred to as "software deployment." In general, the deployment process consists of numerous interdependent actions with possible transitions between them; for this purpose, we utilized the servers listed below.

3.9.1- Visual Studio Code

Visual Studio Code is a free and open-source code editor created by Microsoft that supports a wide range of programming languages through extensions. It supports autocompletion, syntax colouring, debugging, and git commands. [41]

Figure 60 Visual Studio Code logo

3.9.1.1- Advantages of Visual Studio Code

- Customize each feature to your taste and install as many third-party extensions as you like.
- VS Code provides enhanced built-in support for Node.js programming using JavaScript and TypeScript, driven by the same underlying technologies that power Visual Studio. VS Code also provides excellent tooling for web technologies like JSX/React, HTML, CSS, SCSS, Less, and JSON.
- Architecturally, Visual Studio Code blends the finest of web, native, and language-specific technology. [42]

3.9.2- Postman

Postman allows you to create and execute HTTP requests, save them in history so you can re-use them, and organize them into Collections. This categorization allows for grouping queries in a "functional" manner (for example, the addition of an item to a shopping cart or an identifying procedure).

Postman also manages environments, allowing you to contextualize variables and execute queries or series of queries in various settings. [43]

Figure 61 Postman logo

3.9.2.1- Advantages of Postman

- User-friendliness. Testers may quickly create test suites by filling out templates with a simple UI. in addition to code snippets
- Accessibility. Signing into the account on a device with the Postman application or the Postman browser extension allows Postman users to access their files effortlessly.
- Postman lets you make and execute HTTP requests, preserve them in a history for later use, and arrange them into Collections.
- Postman also maintains environments, which allow variables to be contextualized and queries to be run.

3.10- Tools comparison

We used various programming languages and frameworks, and hosting servers and finally, we made a comparison between them under different categories, and we got these results:

Tools	Laravel	FastAPI
Service Web	• It's useful for creating RESTful APIs. • It has a fantastic built-in ORM. • Documentation is excellent. • Handles event queuing • It has its own CLI. • Web apps with authentication are simple to create. • Gives developers a lot of flexibility in how they build their projects. • Template system with a lot of power • Excellent Ecosystem • Fantastic Community • It has too many magic methods, which makes debugging and autocompletion more difficult.	• Dependency injection system • Based on Starlette and Pydantic, so, it's one of the fastest Python frameworks • Automatic docs, it generates interactive API documentation automatically from your code • Editor completion • Based on Async IO, which gives it high concurrency • World class documentation • Independent of database or ORM, but compatible with all of them • Support for background tasks, thanks to being based on Starlette • Data validation • Its community is smaller than Django Rest Framework

	• Model attributes are difficult to use.	

Tools	Cloudways	Colaboratory
Deployment	• It is based on local hardware. • The system processor is used, and There is no access to an external GPU, CPU, or TPU. • Installation of a library by hand • The amount of time it takes to complete a task is determined by the amount of memory available on the machine. • It is possible to share it with others without having to download it. • Require anaconda or python to be installed on your machine • Without your hard drive, we won't be able to access our notebooks • Free of charge	• Google's server is used. • The GPU and TPU are offered for free, and the local machine can still be used. • The majority of the essential libraries are already installed. • It is possible to share it with others without having to download anything. • Runtime is 12/24 hours and can be halted by Google. • There is no need to install anything; it may be accessed through a web browser. • Because it's saved in Google Disk, it can be accessed from anywhere without a hard drive. • It is partially costless.

Tools	Jupyter Lab	Colaboratory
Deployement	• It is based on local hardware. • The system processor is used, and There is no access to an external GPU, CPU, or TPU. • Installation of a library by hand • The amount of time it takes to complete a task is determined by the amount of memory available on the machine. • It is possible to share it with others without having to download it. • Require anaconda or python to be installed on your machine	• Google's server is used. • The GPU and TPU are offered for free, and the local machine can still be used. • The majority of the essential libraries are already installed. • It is possible to share it with others without having to download anything. • Runtime is 12/24 hours and can be halted by Google. • There is no need to install anything; it may be accessed through a web browser. • Because it's saved in Google Disk, it can be accessed from anywhere without a hard drive. • It is partially costless

	• Without your hard drive, we won't be able to access our notebooks • Free of charge	

3.11- Chapter Conclusion

We have seen various materials used in the development phase of the life cycle, which we have defined in this chapter, and the advantages that motivated us to make such choices. Yet, as we can see. These were the tools used without explaining the “what”, so what exactly are we going to deploy?

In order to answer these questions, we need to explain the algorithms deployed in the next chapter.

Chapter 4

Deployed models and their architectures

4.1- Introduction:

This last chapter is dedicated to the description of the deployed models in the proposed applications. Our project focused mainly on deploying four different machine learning models into production with different architectures. The first model concers the medical field, it is developed to the purpose of skin segmentation images. The second one is a model that aims to detect objects within images and videos. The third and the fourth ones are models based on the cvlib python library that permits facial and gender detection.

4.2- Skin segmentation model:

Medical image segmentation has an essential impact on computer-aided diagnosis systems. It's considered one of the most vital medical imaging processes. It helps to divide the image into areas based on a specific description.

The problem with medical imaging, in general, is the lack of data to train the models because it is considered confidential data. The second major problem is that these data are not labelled, so training the model with a few unlabelled images is complex and avoiding the overfitting is almost impossible.

Primarily, medical image segmentation is based on Unet and EfficientNet architectures. But the proposed model in this project has a different architecture that uses the model B0 of EfficientNet to optimize the predictions speed and economize the used memory. [44]

4.2.1- Unet Architecture:

It's an architecture developed by Olaf Ronneberger for biomedical images segmentation's use. It is a convolutional architecture consisting of two parts: the applet encoder's contraction part, which captures the context in the image and the decoder part used for symmetric expansion. [44]

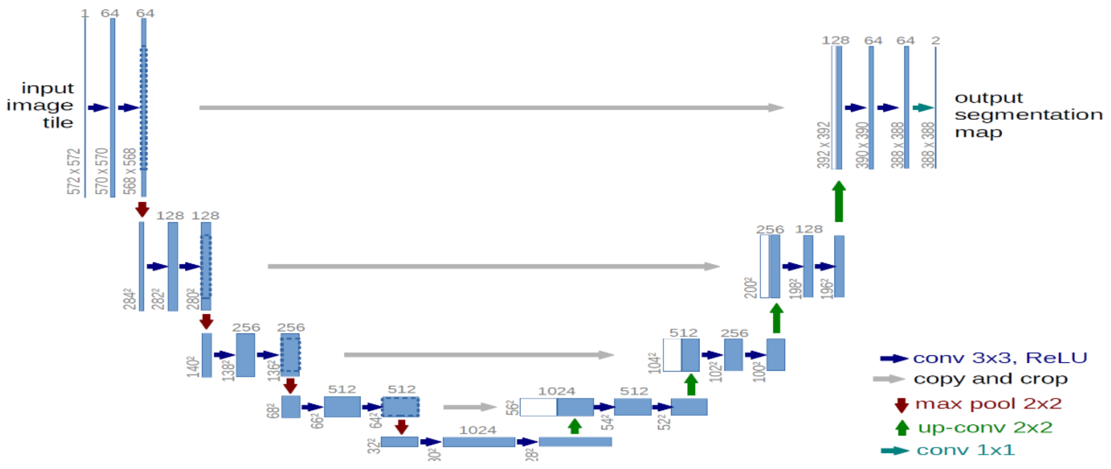


Figure 62 Unet architecture

4.2.2- Image processing:

This method performs multiple operations on the images to prepare them to adapt them to the neural network. The reason to make these operations is to get more images to train the model. These operations consist of Creating labels for the images, Data augmentation and Data normalization.

4.2.3- Execution process:

The skin segmentation model needs input data that must be a medical image resulting from skin MRI. There are many medical image formats, but the model deals only with nifti files in this case.

After the execution, it generates two different files. The first is a medical image file of nifti format, and the second is a .jpg image representing the 2D mask image.

Here is an example of an input file and its appropriate resulted image:

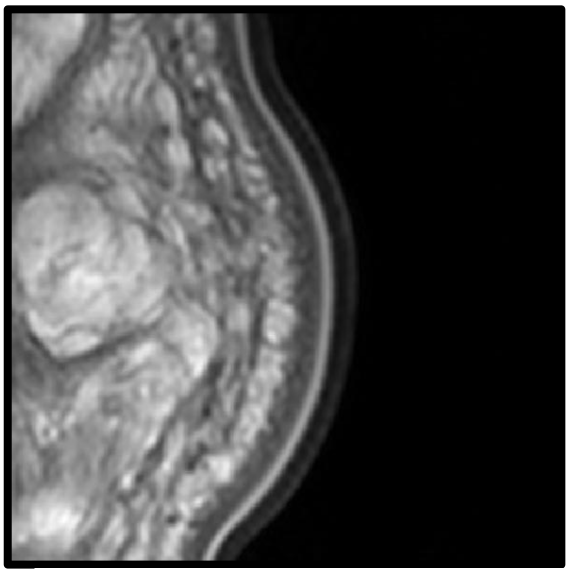


Figure 63 Skin segmentation input image

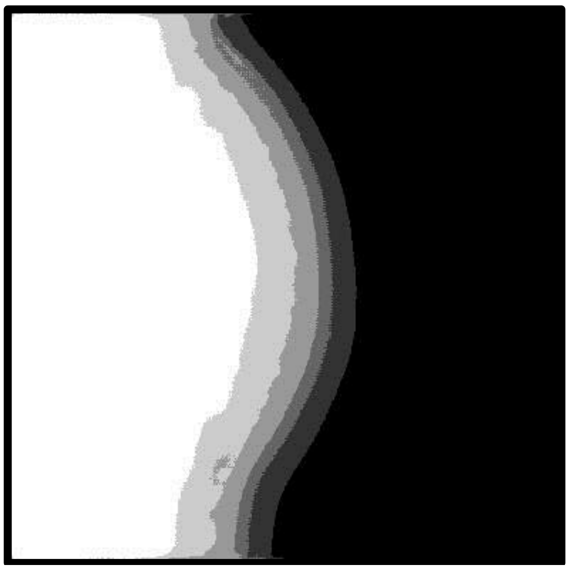


Figure 64 Skin segmentation output image

4.3- Object detection model:

Object detection is a critical computer vision functionality that detects objects of predefined classes (such as persons, cars, animals, and other classes) from digital supports, including photos and video frames.

This project integrates the YOLO v5 model, a family of compound-scaled object detection models trained on the COCO (Common Objects in Context) dataset. [45]

4.3.1- Yolo v5 architecture:

The Yolo v5 model is based on Darknet architecture, an Open-Source convolutional neural network written in C and CUDA. It is considered the **backbone** of the Yolo v5 architecture.

This model also integrates the PANet in the **neck** of the general architecture. The model mainly adopts it to improve the process of instance segmentation by preserving spatial information.

Finally, the architecture's **head** comprises Yolo Layers to mix and combine image features to pass them to prediction. [46]

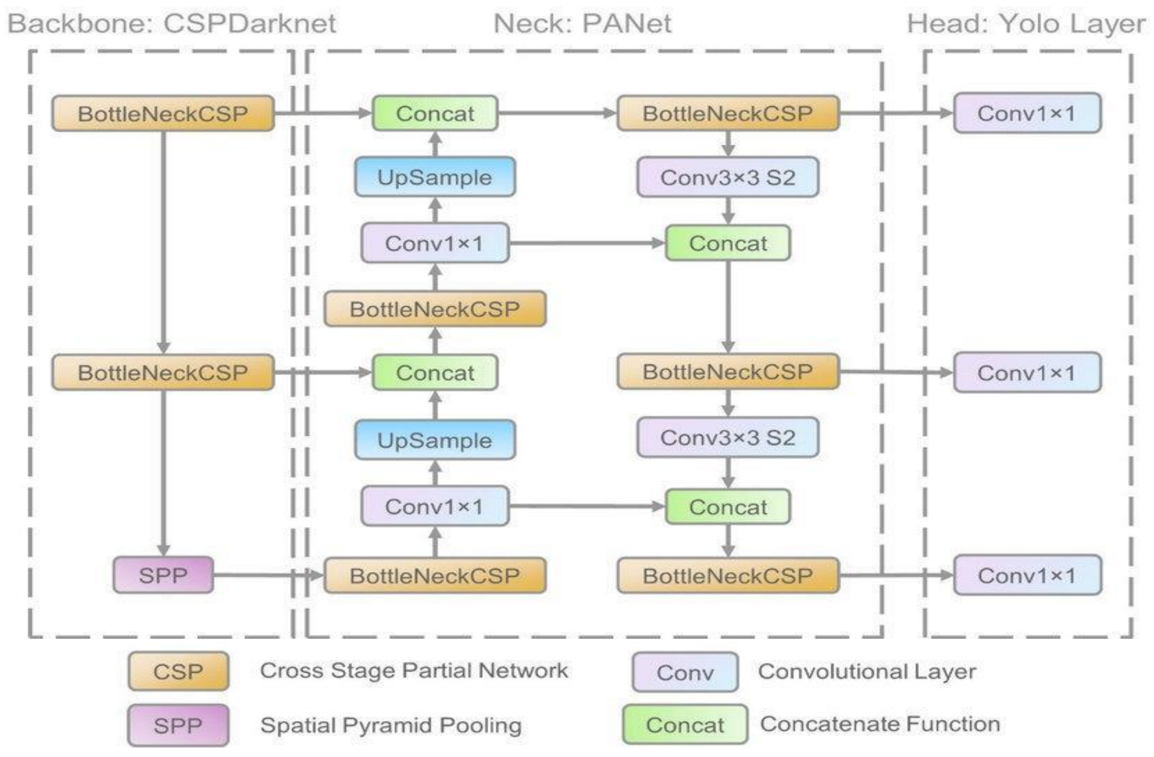


Figure 65 Yolo v5 architecture

4.3.2- Execution process:

The Yolo v5 model can detect objects from diverse digital media types as images and videos. It decomposes the input data into frames for video detection and treats each frame as a single image.At the end of the execution, the model calls the OpenCV library to build the result file before returning it to the output path.

The major problem we faced at this level was to specify a codec used to build the resulted video. As the returned files are being used on the web, we had to find a specific codec supported by the OpenCV library and at the same time compatible with the web browser. For this reason, we adopted the VP09 videos codec to generate the resulted videos.

Here is an example of the Yolo v5 execution:

Figure 67 Yolo v5 input example

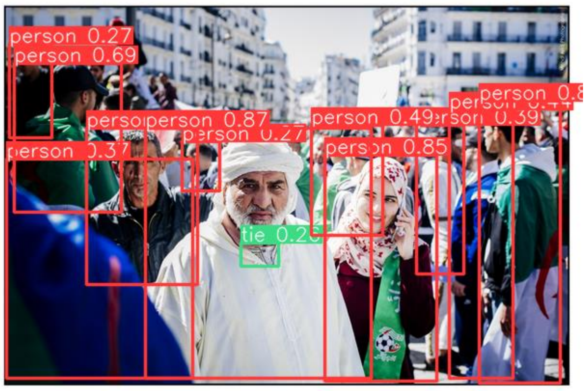


Figure 66 Yolo v5 output example

4.4- Face and gender detection:

The last model deployed in this project is a face and gender detection model. It is based on the CVlib python library, which is a specialized library in object detection.

This library is fed principally by two compelling libraries: OpenCV and TensorFlow. [47]

It supports both image and video files as input and includes three principal functions:

- Face detection
- Gender detection
- Object detection

We focused only on the image input files, the face and the gender detection options in this project. [48]

4.4.1- Execution process:

4.4.1.1- Face detection:

The python script needs to call the detect_face() function from the CVlib library to detect the face. The function returns an array of all the faces it found and an array of numbers to show how sure it is that those are faces which represents the confidence of the prediction.

The figure below represents an example of an input and output file using this function:

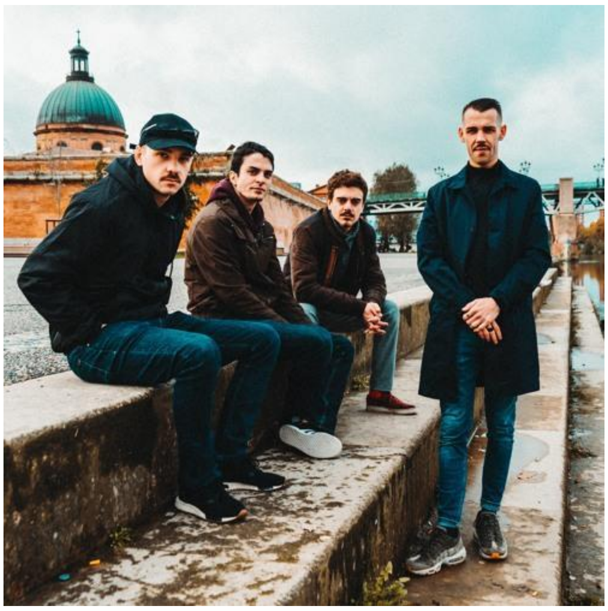


Figure 68 CVlib face detection output file

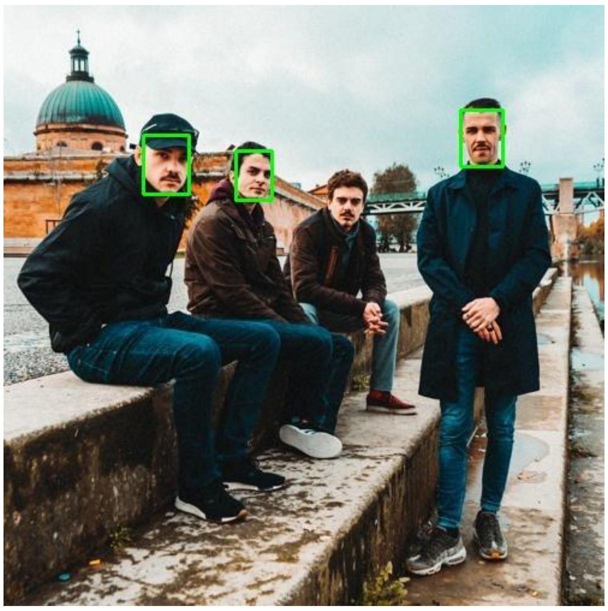


Figure 69 CVlib face detection input file

4.4.1.2- Gender detection:

Gender detection is one of the popular computer vision applications. It is possible to make gender detection using CVlib by calling the prebuilt function **detect_gender()** associating with the face frame already detected.

The following figures are an example of the execution of this option.

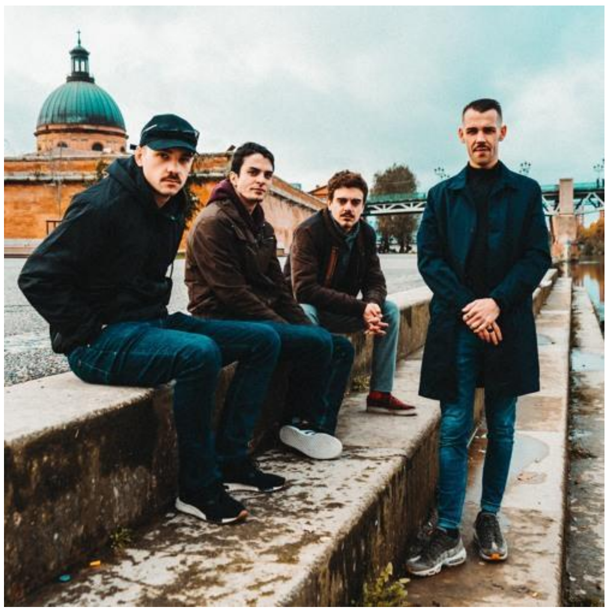


Figure 71 CVlib gender detection input file

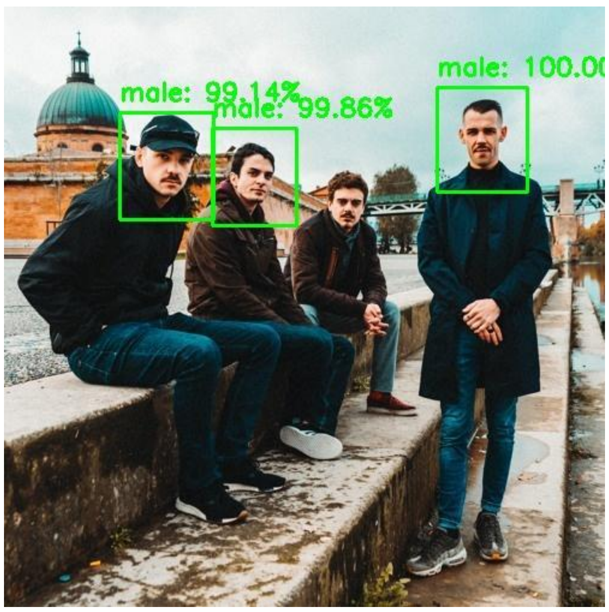


Figure 70 CVlib gender detection output file

4.5- Chapter conclusion:

Along with this chapter, we described the deployed models in our project. We also highlighted the architecture of each model as the Unet architecture of the skin segmentation model and the Yolo v5 architecture based on DarkNet and PAnet neural networks. We finished with giving an example of inputs and outputs of each case to give a general idea of the use of these scripts.

Conclusion:

The popularity of web applications has increased dramatically with agencies specializing in artificial intelligence, to use their intelligent software in an entrepreneurial setting to generate revenue, hence the term MLOps which is a booming new research field. The objective of our project was to offer simple, secure and above all fast deployment requests, while setting up ergonomic interfaces, with options that can facilitate users' navigation. This Work resulted in the creation and deployment of two web applications, separate from one to another that take care of the different tasks that allow users to test the different algorithms offered, which we can quote:

- Facilitate the insertion of the image or video with the appropriate extention
- Test the algorithm on the uploaded input.
- Show up the output with the possibility of downloading if weshed
- Possibility to create a user profile for monitoring, which provides access to dashboard presenting the past operations.
- Ability to contact the administrator at any time through the contact page.

we have implemented the options that machine learning users may need, these features that we call "Sprint" have been translated into UML diagrams, the design methodology used was driven by the SCRUM method which is considered among the most used agile methods, in order to ensure the smooth running of the project accompanied by UML which is considered among the most used modeling languages to illustrate the design process In perspective, the Web application can be improved by adding other features such as:

- Include full virtual machine management from the administrator interface
- Include full file and profile management from the user interface
- Deploy even more models such as “Color Model” “Demographics Face Recognition Model”, “AI Food Recognition Model”, “Subject Visual Segment”, “Arabic manuscript recognition” and many more 'others.
- Add a rather revolutionary product that we call "Custom Model" which allows the user to create his own model, use his own inputs or texts with his own concepts.
- Create an API that allows our customers to use our models and our servers
- Develop a payment system according to a market study that we will carry out soon. We hope that these two platforms will be perpetually improving in terms of performance and functionality as new technologies appear. [33]

Bibliography

[1] A. Ng, "Apprentissage automatique," 2015. [Online]. Available: www.coursera.org/learn/machine-learning. [Accessed 17 06 2021].

[2] K. Wakefield, "A guide to the types of machine learning algorithms and their applications," 6 May 2018. [Online]. Available: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html. [Accessed 3 07 2021].

[3] P. Canuma, "MLOps: What It Is, Why it Matters, and How To Implement It - neptune.ai," 14 1 2021. [Online]. Available: https://neptune.ai/blog/mlops-what-it-is-why-it-matters-and-how-to-implement-it-from-a-data-scientist-perspective. [Accessed 12 6 2021].

[4] H. Tyagi, "What is MLOps — Everything You Must Know to Get Started," 2020 march 25. [Online]. Available: https://towardsdatascience.com/what-is-mlops-everything-you-must-know-to-get-started-523f2d0b8bd8. [Accessed 18 06 2021].

[5] "Introduction to MLOps," 30 11 2020. [Online]. Available: https://medium.com/illumination/introduction-to-mlops-f877ccf10db1. [Accessed 25 07 2021].

[6] mark treveil and dataiku team, Introducing MLOps, 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc, November 2020.

[7] A. NG, "Introduction-to-machine-learning-in-production," 2021. [Online]. Available: https://www.coursera.org/learn/introduction-to-machine-learning-in-production. [Accessed 10 06 2021].

[8] T. Sufiyan, "What is ReactJS: Introduction To React and Its Features," 30 09 2021. [Online]. Available: https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs. [Accessed 07 2021].

[9] S. Morris, "Tech 101: What Is React JS?," 23 08 2020. [Online]. Available: https://skillcrush.com/blog/what-is-react-js/. [Accessed 12 06 2021].

[10] "7 Advantages of ReactJS for Building Interactive User Interfaces," 05 08 2020. [Online]. Available: https://www.clariontech.com/blog/7-advantages-of-reactjs-for-building-interactive-user-interfaces. [Accessed 03 05 2021].

[11] O. Kopachovets, "ReactJS Advantages: Scalability, Challenges, Pitfalls," 20 11 2020. [Online]. Available: https://procoders.tech/blog/advantages-of-using-reactjs/. [Accessed 28 07 2021].

[12] Laravel, "Introduction," 01 01 2021. [Online]. Available: https://laravel.com/docs/4.2/introduction.

[13] Kinsta, "Le framework PHP Laravel – la construction d'applications web pour tous," 16 04 2021. [Online]. Available: https://www.pappleweb.com/index/definition-de-laravel/. [Accessed 25 08 2021].

[14] D. Garbar, "The Top 10 Advantages Of Using Laravel PHP Framework," 05 01 2020. [Online]. Available: https://belitsoft.com/laravel-development-services/10-benefits-using-laravel-php-framework. [Accessed 20 07 2021].

[15] JDN, "Bootstrap : définition, tutoriels, astuces, pratiques," 28 08 2021. [Online]. Available: https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/. [Accessed 08 07 2021].

[16] T. Contributor, "DEFINITION," 01 01 2017. [Online]. Available: https://whatis.techtarget.com/definition/bootstrap. [Accessed 10 08 2021].

[17] graygrids, "The 7 Great Advantage and Reasons to Use Bootstrap as a Front-end CSS Framework," 2021. [Online]. Available: https://graygrids.com/advantage-reasons-use-bootstrap-front-end-css-framework/. [Accessed 20 07 2021].

[18] "Top 10 Benefits of Using Bootstrap!," 01 01 2020. [Online]. Available: https://www.xeliumtech.com/blog/Top-10-Benefits-of-Using-Bootstrap. [Accessed 15 07 2021].

[19] Redhat, "What is Docker?," 09 01 2018. [Online]. Available: https://www.redhat.com/en/topics/containers/what-is-docker. [Accessed 01 09 2021].

[20] S. Yegulalp, "Why you should use Docker and containers," 10 10 2028. [Online]. Available: https://www.infoworld.com/article/3310941/why-you-should-use-docker-and-containers.html. [Accessed 15 08 2021].

[21] Simplilearn, "What is Docker: Advantages and Components," 18 09 2021. [Online]. Available: https://www.simplilearn.com/tutorials/docker-tutorial/what-is-docker. [Accessed 21 07 021].

[22] Docker, "Use containers to Build, Share and Run your applications," 2021. [Online]. Available: https://www.docker.com/resources/what-container. [Accessed 01 05 2021].

[23] Nginx, "What is NGINX?," 2021. [Online]. Available: https://www.nginx.com/resources/glossary/nginx/. [Accessed 10 06 2021].

[24] Coolicehost, "Ten Great Advantages of Nginx," 18 07 2015. [Online]. Available: https://blog.coolicehost.com/ten-great-advantages-of-nginx/. [Accessed 05 08 2021].

[25] S. Yegulalp, "Get started with FastAPI," 18 09 2021. [Online]. Available: https://www.infoworld.com/article/3629409/get-started-with-fastapi.html. [Accessed 01 05 2021].

[26] J. Sandy, "Choosing between Django, Flask, and FastAPI," 04 01 2021. [Online]. Available: https://www.section.io/engineering-education/choosing-between-django-flask-and-fastapi/. [Accessed 14 05 2021].

[27] R. Naushad, "Flask Vs FastAPI which one should you choose?," 11 11 2020. [Online]. Available: https://blog.accubits.com/flask-vs-fastapi-which-one-should-you-choose/. [Accessed 18 05 2021].

[28] Developers.google, "Introduction to Google Drive API," 2021. [Online]. Available: https://developers.google.com/drive/api/v3/about-sdk. [Accessed 04 06 2021].

[29] Jupyter, "https://jupyter.org/," 2021. [Online]. Available: https://jupyter.org/documentation. [Accessed 12 06 2021].

[30] A. Bhandari, "10 Compelling Reasons you Should Use JupyterLab for Data Science Coding," 25 06 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/06/10-powerful-reasons-jupyterlab-data-science/. [Accessed 24 05 2021].

[31] R. google, "Colaboratory, Frequently Asked Questions," 2021. [Online]. Available: https://research.google.com/colaboratory/faq.html?hl=fr. [Accessed 10 04 2021].

[32] M. V. D. Reym, "7 Advantages of Using Google Colab for Python," 04 01 2020. [Online]. Available: https://python.plainenglish.io/7-advantages-of-using-google-colab-for-python-82ac5166fd4b. [Accessed 28 04 2021].

[33] CONTRIBUTOR, "8 Advantages of Using MySQL," 07 26 2017. [Online]. Available: https://devops.com/8-advantages-using-mysql/. [Accessed 05 09 2021].

[34] Mongodb, "Qu'est-ce que MongoDB ?," 2021. [Online]. Available: https://www.mongodb.com/fr-fr/what-is-mongodb. [Accessed 13 09 2021].

[35] Mongodb, "Advantages of MongoDB," 2021. [Online]. Available: https://www.mongodb.com/advantages-of-mongodb. [Accessed 12 09 2021].

[36] "Les 7 meilleurs avantages à choisir l'hébergement Google Cloud," 02 03 2021. [Online]. Available: https://kinsta.com/fr/blog/hebergement-google-cloud/. [Accessed 01 05 2021].

[37] "Google Compute Engine: Features and Advantages," 27 09 2021. [Online]. Available: https://www.whizlabs.com/blog/google-compute-engine-features-and-advantages/. [Accessed 23 06 2021].

[38] "What Are the Advantages of Google Compute Engine?," 3 September 2021. [Online]. Available: https://www.parallels.com/blogs/ras/compute-engine/. [Accessed 01 03 2021].

[39] Digital, "Best web hosting," 3 10 2021. [Online]. Available: https://digital.com/best-web-hosting/cloudways/. [Accessed 20 06 2021].

[40] Cloudways, "Hosting Made Simple, Fast and Convenient," 2021. [Online]. Available: https://www.cloudways.com/en/features.php. [Accessed 27 06 2021].

[41] P. Pedamkar, "What is Visual Studio Code?," 2020. [Online]. Available: https://www.educba.com/what-is-visual-studio-code/. [Accessed 02 03 2021].

[42] O. A. ,. P. Greg Van Liew, "Why did we build Visual Studio Code?," 02 9 2021. [Online]. Available: https://code.visualstudio.com/docs/editor/whyvscode. [Accessed 9 03 2021].

[43] O. Howard, "Postman for API Testing — Pros, Cons, and Alternative Solutions," 14 09 2020. [Online]. Available: https://dzone.com/articles/postman-for-api-testing-pros-cons-and-alternative. [Accessed 10 03 2021].

[44] A. M. Lamine and Z. Ahcene, "MASTER THESIS 2D/3D medical image segmentation by embedding EfficientNet in Convolutional neural network: Application on BraTS challenge 2020," 2020.

[45] J. Solawetz, "How to Train A Custom Object Detection Model with YOLO v5," 15 01 2020. [Online]. Available: https://towardsdatascience.com/how-to-train-a-custom-object-detection-model-with-yolo-v5-917e9ce13208. [Accessed 15 08 2021].

[46] R. Xu, K. Lu, Lin Cao and Yunfei Liu, "A Forest Fire Detection System Based on Ensemble Learning," *forests,* 2021.

[47] G. Sharma, "cvlib : Yolo Object Detection in Seconds!," 26 06 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/yolo-object-detection-in-seconds-cvlib/. [Accessed 12 08 2021].

[48] "Home - cvlib," [Online]. Available: https://docs.cvlib.net/.

Abstract:

The thesis was written with a view to obtaining the diploma at the end of study in computer science, software engineering course. It deals with the problems encountered during the step of the deployment of several Machine Learning algorithms on the web.

To implement the solutions, we used an agile method called SCRUM which is based on UML as a modeling language, and the Model-View-Controller as a design pattern.

The implementation was done under the integrated development environment Visual Studio Codes using several languages such as Javascript, PHP and Python, and programming frameworks ReactJS, Laravel and FastAPI, regarding the creation of isolated virtual machines we used Docker and Google Colab, we used NGINX too in order to set the HTTP server, when it comes to the deployment cloud servers we had faith in Google Cloud Compute engine and Cloudways, an since we had to manage data we trusted a relational database with MySQL and NoSQL database with MongoDB

Meanwhile the goal is to run the Machine Learning algorithms so we were obliged to understand and to modify the programmes, especially dealing with inputs and outputs.

The security layer of the application was exposed by a certain number of mechanisms allowing to increase the achievement of the security objectives (authentication, access control etc.).

Resumé:

Le mémoire a été rédigé en vue de l'obtention du diplôme de fin d'études en informatique spécialité génie logiciel. Il traite des problèmes rencontrés lors de l'étape de déploiement de plusieurs algorithmes de Machine Learning sur le web.

Pour implémenter les solutions, nous avons utilisé une méthode agile appelée SCRUM qui est basée sur l'UML comme langage de modélisation et le Modèle-vue-contrôleur comme modèle de conception.

La mise en œuvre a été faite sur l'environnement de développement intégré Visual Studio Code utilisant plusieurs langages tels que Javascript, PHP et Python, et les frameworks de programmation ReactJS, Laravel et FastAPI, en ce qui concerne la création de machines virtuelles isolées, nous avons utilisé Docker et Google Colab. Nous avons utilisé NGINX afin de définir le serveur HTTP. En ce qui concerne les serveurs de déploiement, nous avons fait confiance au Google Cloud Engine et à Cloudways, et comme nous devions gérer des données, nous avons fait appel à une base de données relationnelle avec MySQL et à une base de données NoSQL avec MongoDB.

Tandis que le but est d'exécuter les algorithmes de Machine Learning, nous avons donc été obligés de comprendre et de modifier les programmes, en particulier en ce qui concerne les entrées et les sorties de données.

La couche de sécurité de l'application a été exposée par un certain nombre de mécanismes permettant d'augmenter l'atteinte des objectifs de sécurité (authentification, contrôle d'accès etc.)

Tasenselkimt:

Nura tazrawt-a iwakken ad nawi yes-s agerdas ɣer taggara n useggas. Taɣult n tselselkimt, tuzzigt n tesmedna n useɣzan.

S umata, iqeddec ɣef uguren i d-yemlal lawan n umecwar n uzuzer n waṭas n ilguritmen n tmacint Learning deg Web.

Iwakken ad nselkem tifratin, nessemres tarrayt i yettusemman SCRUM, tin i isenden ɣef UML am tutlayt n tmudemt akked MVC tamudemt n useɣwes.

Deg lawan n uselkem, yella-d waya s ukeččum n twennaḍt n usnerni n Visual Studio code akked useqdec n tutlayin i yemgaraden am Javascript, PHP d Python akked usihel ReactJS, Laravel d FastAPI. Ma yella deg wayen yerzan asnulfu n tmacinin tuḥlisin yettwaɛezlen, nessemras Docker d Google Colab. Nesseqdec daɣen NGINX iwakken ad d-nesbadu aqeddac http. Deg wayen yerzan iqeddacen n uzuzer, nexdem taflest deg Google Cloud Engine akked Cloudways imi i d-yewwi ad nesselḥu inefkan anida i nger tiɣri i yiwen n wadda n yinefkan assaɣan akked MySQL rnu yiwen n wadda n yinefkan NoSQL akked MongoDB.

Seg tama-nniḍen, iswi d aselkem n ilguritmen n Machine Learning, dɣa yessefk ad negzu, ad nbeddel ahilen ladɣa deg wayen yerzan akeččum d tuffɣa n yinefkan.

Tettwafser tissi n laman n usnas sɣur kra n umḍan n tmacinin i izemren ad ǧǧent asemɣer (zzyada) deg usiweḍ ɣer yiswan n laman (asesteb, asuddes n ukeččum atg.)