



Projet de Fin de module

Logiciels et Systèmes Intelligents Systèmes Embarqués

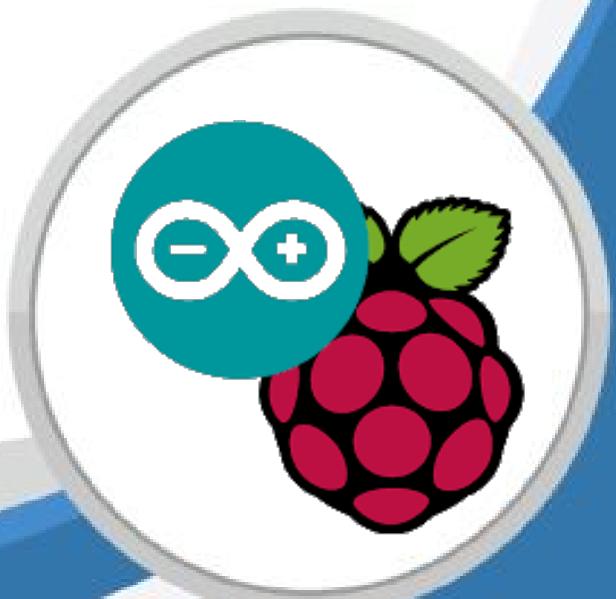
SMART PARKING : Une Approche Intelligente avec Arduino et Raspberry Pi

Réalisé par :

DAAOUAN MOHAMMED
FRIKH SAID

Encadré par :

Pr. BEN AHMED Mohamed



Année universitaire : 2023-2024

TABLE DES MATIÈRES

Table des figures	iii
Introduction	1
1 Analyse et Conception	2
1.1 Introduction	2
1.2 Identifications des acteurs	2
1.2.1 Un administrateur	2
1.2.2 Client	3
1.3 Les règles de gestion et d'organisation	3
1.4 Diagramme de cas d'utilisation	3
1.5 Diagramme de classes	4
1.6 Conclusion	5
1.7 Pipeline du projet	6
2 Intégration de YOLOv8 : Détection avec Deep Learning	7
2.1 YOLO	7
2.1.1 Contexte de YOLO	7
2.1.2 Applications de YOLO	8
2.2 Fine-tuning dans YOLO	8
2.2.1 Définition de fine tuning	8
2.2.2 Importance du Fine-tuning	8
2.3 Phase de Création du Modèle	8
2.3.1 Préparation du Dataset	8
2.3.2 Exemple de dataset	10
2.3.3 Entraînement et Évaluation du Modèle	11
3 les technologies utilisées	14
3.1 Django	14
3.2 Angular	14
3.3 mysql	15
3.4 git/github	15
3.5 Arduino	15

3.6	Raspberry Pi 4	16
3.7	Servomoteur	16
3.8	Montage dans tinkercard	17
3.9	Montage dans la réalité	17
4	Fonctionnalités de "SmartParking"	18
4.1	Introduction	18
4.2	Authentification et Interface Administrative	18
4.3	Gestion des Abonnements et Contrôle d'Accès	19
4.4	Suivi Clients	20
4.5	Génération de Rapports et Envoi d'Emails	22

TABLE DES FIGURES

1.1	Diagramme de cas d'utilisation	4
1.2	Diagramme de classe	5
1.3	Pipeline	6
2.1	yolov8	7
2.2	Dataset cercle	9
2.3	Dataset bar	9
2.4	Dataset images	10
2.5	dataset labels	10
2.6	schema de projet	11
2.7	configuration de modèle	11
2.9	résultats de modèle	13
4.1	Authentification	18
4.2	Contrôle d'Accès (1)	19
4.3	Gestion des Abonnements	19
4.4	Contrôle d'Accès (2)	20
4.5	Liste clients	20
4.6	Informations client	21
4.7	Prolongement	21
4.8	Exportation de Rapports et Envoi d'Emails	22
4.9	Exportation de Rapport	22
4.10	Envoi d'Email	23

INTRODUCTION

Dans un monde en constante évolution, les défis liés à la gestion des ressources urbaines ne cessent de croître. Parmi ces défis, le stationnement automobile demeure une préoccupation majeure, avec des conséquences directes sur la fluidité du trafic, l'efficacité énergétique et l'expérience globale des citoyens. C'est dans ce contexte que notre projet de parking intelligent, intégrant des technologies de pointe telles que l'intelligence artificielle (IA), les systèmes embarqués (Arduino et Raspberry Pi) et en combinaison avec des outils de développement modernes tels que Angular et Django.

Notre approche vise à transformer les espaces de stationnement en environnements intelligents, capables de prendre des décisions en temps réel pour optimiser l'utilisation des ressources disponibles. L'utilisation d'Arduino et Raspberry Pi garantit une intégration harmonieuse de capteurs et d'actuateurs, permettant ainsi une collecte de données précise et une réactivité accrue du système. De plus, l'utilisation de technologies de développement web comme Angular et Django offre une interface utilisateur conviviale et une gestion efficace des données.

Ce rapport détaillera la conception, la mise en œuvre et les résultats obtenus dans le cadre de la mise en place de notre système de parking intelligent, mettant en lumière les avantages significatifs qu'il apporte en termes d'efficacité opérationnelle et d'expérience utilisateur améliorée.

CHAPITRE 1

ANALYSE ET CONCEPTION

1.1 Introduction

Notre objectif principal est de créer une plateforme robuste et conviviale, permettant la gestion en temps réel des parkings. Nous nous focaliserons sur la modélisation à l'aide du langage de modélisation UML (Unified Modeling Language) pour détailler les différentes facettes de l'application. Cela englobera l'identification des acteurs impliqués, l'élaboration des cas d'utilisation spécifiques à chaque acteur, ainsi que la création de diagramme de classes. Cette approche méthodique fournira la base nécessaire à une implémentation solide et extensible.

1.2 Identifications des acteurs

Dans le cadre de la conception de notre système intelligent, la première étape consiste à identifier les acteurs qui interagiront avec notre système. Ces acteurs occupent des positions essentielles au sein de l'écosystème de notre projet, et il est donc impératif de comprendre leurs rôles et leurs responsabilités pour élaborer une conception efficace.

1.2.1 Un administrateur

L'administrateur, en tant qu'acteur clé de notre application, assume le rôle de gestionnaire et de communicateur au sein de notre plateforme. Son rôle principal consiste à superviser les différentes composantes de notre système, telles que les clients, les abonnements et les rapports. Il a le pouvoir d'ajouter des clients et des abonnements, de prolonger des abonnements, assurant ainsi la notification des clients par l'envoi d'Emails de notification et la génération de rapports.

1.2.2 Client

L'acteur client est un élément fondamental de notre système SmartParking. Ces utilisateurs jouent un rôle proactif en contactant l'administrateur pour initier le processus d'abonnement de leurs véhicules au parking intelligent.

Le client communique avec l'administrateur, exprimant son intention de souscrire à un abonnement. L'administrateur, en réponse à cette demande, prend en charge la création de l'abonnement dans le système. Cela inclut l'enregistrement des détails du client, la définition de la durée de l'abonnement, et l'attribution des autorisations d'accès appropriées. Une fois l'abonnement établi, le système peut générer des e-mails de notification qu'il envoie au client. Ces e-mails fournissent des informations cruciales telles que les détails de l'abonnement, les instructions d'accès au parking, et tout autre élément pertinent.

1.3 Les règles de gestion et d'organisation

- L'application peut avoir un ou plusieurs administrateurs.
- L'administrateur doit se connecter par son "UserName" et "Password".
- L'administrateur doit avoir un "UserName" et "Password" valide pour s'authentifier.
- Lorsque l'administrateur se connecte avec succès, un token lui est attribué.
- L'administrateur est le seul à pouvoir administrer les ressources clés du système, notamment les clients, les abonnements, les rapports et les e-mails.
- L'administrateur peut ajouter un ou plusieurs clients/abonnements.
- L'administrateur peut prolonger un ou plusieurs abonnements.
- L'administrateur peut générer un ou plusieurs rapports.
- L'administrateur a la possibilité d'envoyer un ou plusieurs e-mails de notification aux clients.
- Le client a la faculté de solliciter un ou plusieurs abonnements.
- Le client doit disposer d'un abonnement actif pour accéder au parking.
- Un client ayant un abonnement actif peut entrer et sortir du parking.
- Le client peut recevoir un ou plusieurs Emails de notification.
- Le client peut prolonger son abonnement une ou plusieurs fois.

1.4 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation constituent une représentation visuelle essentielle de notre système. Ils décrivent de manière exhaustive les diverses actions et interactions possibles entre les acteurs et notre système.

Ce diagramme de cas d'utilisation met en évidence les fonctionnalités clés offertes par notre système, tout en illustrant les scénarios d'utilisation typiques, les dépendances entre les cas d'utilisation, et les interactions entre les acteurs.

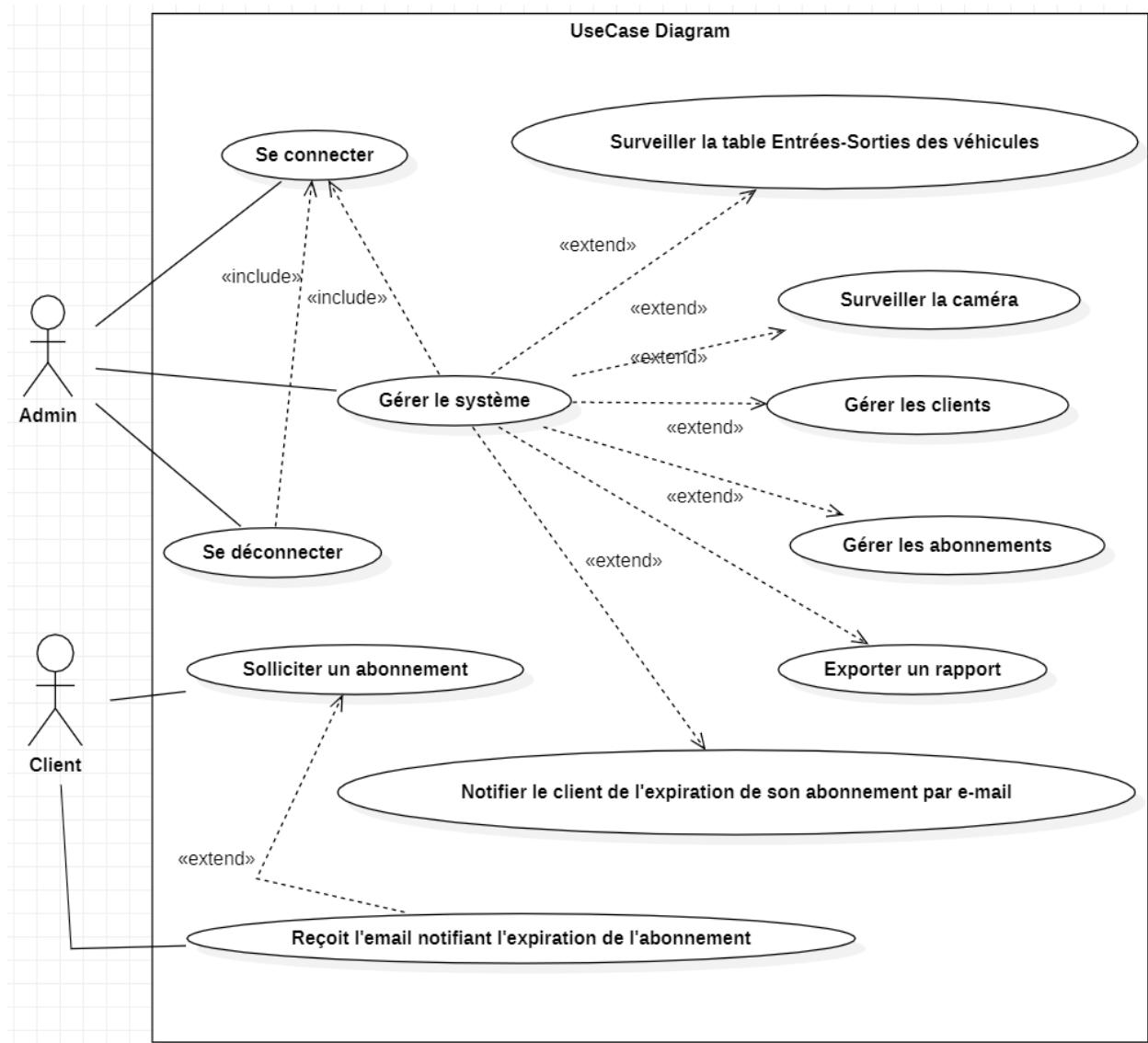


FIGURE 1.1 – Diagramme de cas d'utilisation

1.5 Diagramme de classes

Ce diagramme de classe permet de visualiser comment les différentes parties de notre système sont organisées et interagissent les unes avec les autres, ce qui nous permet de créer des modèles de données solides.

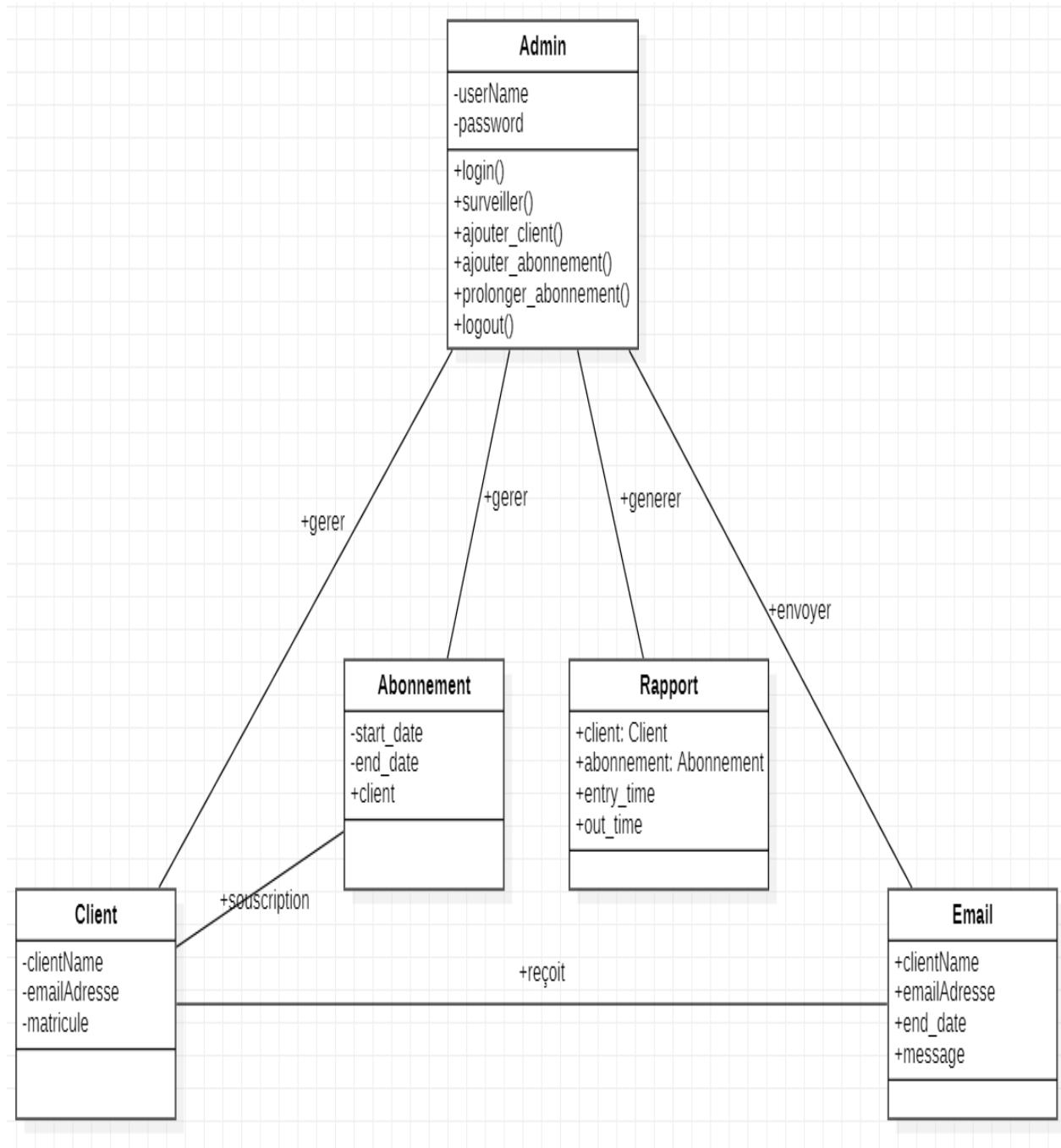


FIGURE 1.2 – Diagramme de classe

1.6 Conclusion

En conclusion, ce chapitre dédié à la conception de notre application pour notre système intelligent "SmartParking" a été essentiel pour la planification et la préparation de notre projet. Nous avons examiné les acteurs clés, les cas d'utilisation associés à chacun d'eux, ainsi que le diagramme de classe qui détaillent la manière dont notre application fonctionnera. Cette analyse en profondeur constitue le fondement solide sur lequel repose notre projet.

1.7 Pipeline du projet

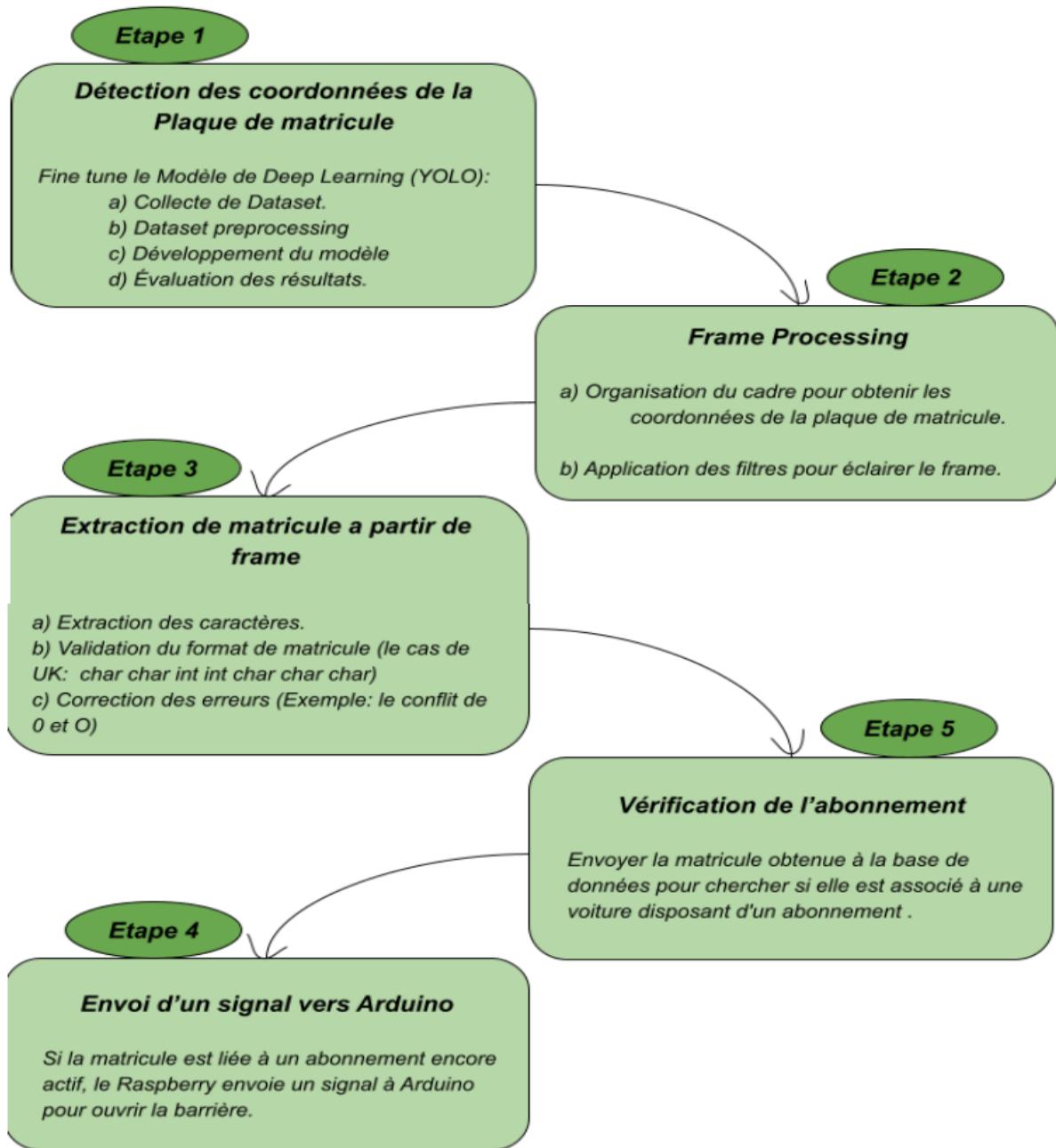


FIGURE 1.3 – Pipeline

— CHAPITRE 2 —

INTÉGRATION DE YOLOv8 : DÉTECTION AVEC DEEP LEARNING

2.1 YOLO

2.1.1 Contexte de YOLO

You Only Look Once (YOLO) est un algorithme de détection d'objets révolutionnaire dans le domaine de l'apprentissage profond. Conçu pour une efficacité maximale, YOLO divise l'image en une grille et prédit les boîtes englobantes et les classes d'objets pour chaque cellule de la grille en une seule passe. Cette approche unique offre une vitesse exceptionnelle, permettant la détection en temps réel d'objets dans des images et des vidéos. Le contexte de YOLO réside dans son architecture novatrice, qui résout le compromis entre la précision de la détection et la rapidité d'exécution, rendant YOLO particulièrement adapté à un large éventail d'applications.

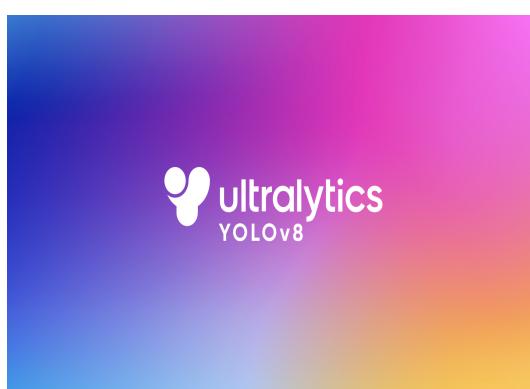


FIGURE 2.1 – yolov8

2.1.2 Applications de YOLO

Les applications de YOLO sont diverses et englobent plusieurs domaines. En vision par ordinateur, YOLO est largement utilisé pour la détection d'objets dans des scénarios en temps réel. Dans la surveillance vidéo, YOLO peut détecter et suivre des objets en mouvement, tandis que dans la conduite autonome, il est employé pour la détection d'obstacles et de piétons. YOLO trouve également des applications dans la réalité augmentée, la sécurité, la médecine et d'autres domaines où une détection rapide et précise d'objets est cruciale. Sa polyvalence et son efficacité ont fait de YOLO l'un des choix préférés pour des projets nécessitant une détection d'objets robuste et en temps réel.

2.2 Fine-tuning dans YOLO

2.2.1 Définition de fine tuning

Le "fine-tuning" (ajustement fin) est une technique d'apprentissage automatique qui consiste à prendre un modèle de réseau neuronal déjà pré-entraîné sur une tâche générale et à l'ajuster spécifiquement pour une tâche particulière. Plutôt que d'entraîner le modèle à partir de zéro, le fine-tuning capitalise sur les connaissances acquises par le modèle pendant sa phase de pré-entraînement. Cela se fait en ajustant les poids et les paramètres du modèle pour s'adapter à des données spécifiques liées à la nouvelle tâche.

2.2.2 Importance du Fine-tuning

Le fine-tuning revêt une importance cruciale dans l'intégration de YOLO pour la détection de plaques d'immatriculation. Lorsque YOLO est pré-entraîné sur des ensembles de données diversifiés, il acquiert une compréhension générale des objets, mais pour des tâches spécifiques telles que la détection de plaques d'immatriculation, le fine-tuning permet d'ajuster le modèle aux particularités de ces objets spécifiques. Par exemple, si le modèle YOLO est pré-entraîné sur des images de circulation routière génériques, le fine-tuning avec un ensemble de données annoté de plaques d'immatriculation spécifiques à un pays ou une région permet d'optimiser la capacité du modèle à identifier précisément ces éléments cruciaux dans des conditions variables telles que l'éclairage, l'angle de vue et les conditions météorologiques. Cela conduit à une détection plus robuste et spécialisée des plaques d'immatriculation dans des scénarios réels, tout en capitalisant sur les connaissances préalables acquises lors du pré-entraînement.

2.3 Phase de Création du Modèle

2.3.1 Préparation du Dataset

2.3.1.1 Statistiques du Dataset

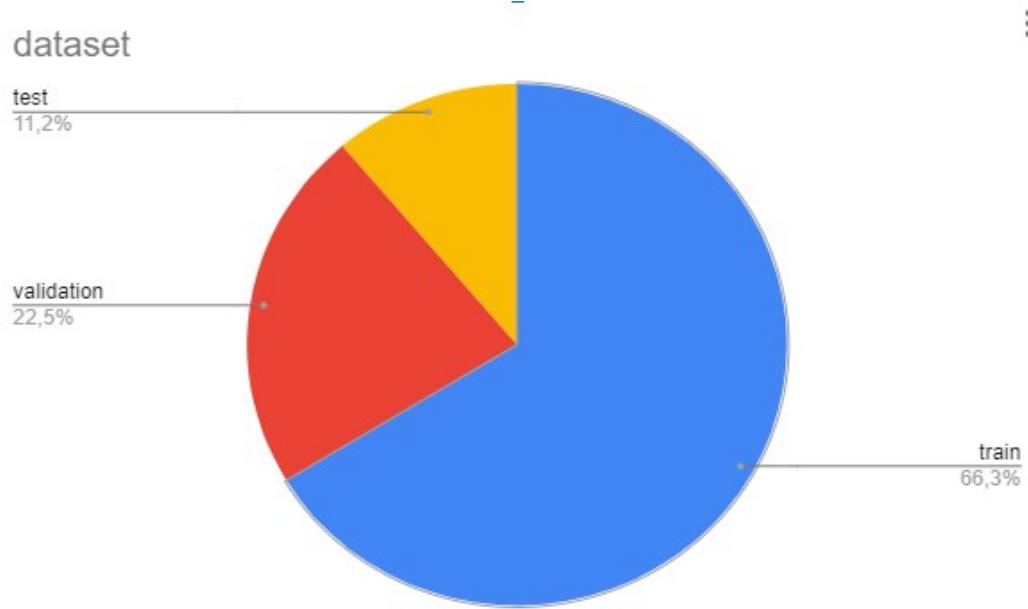


FIGURE 2.2 – Dataset cercle

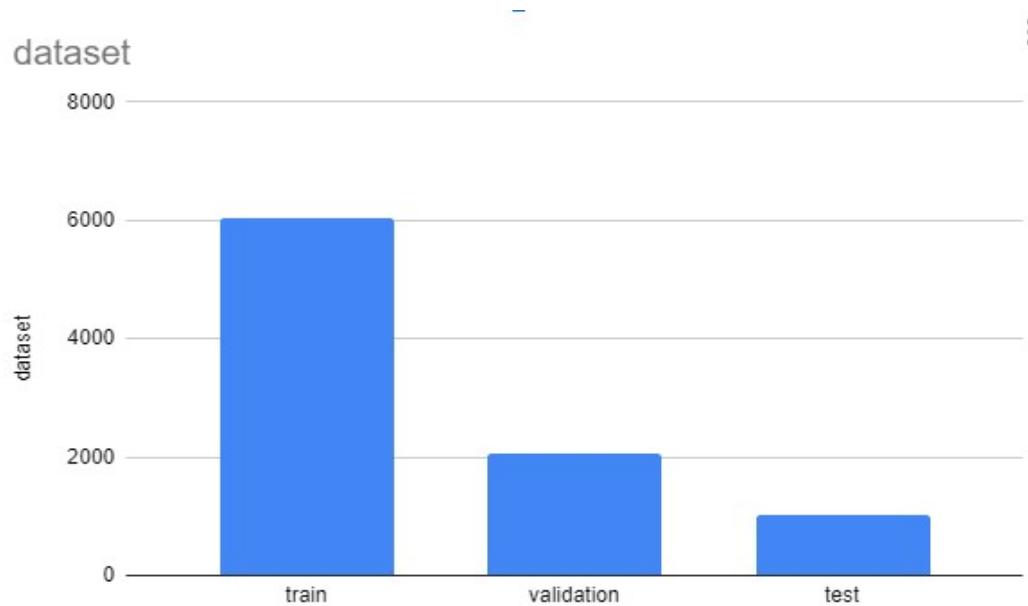


FIGURE 2.3 – Dataset bar

2.3.2 Exemple de dataset

2.3.2.1 Images

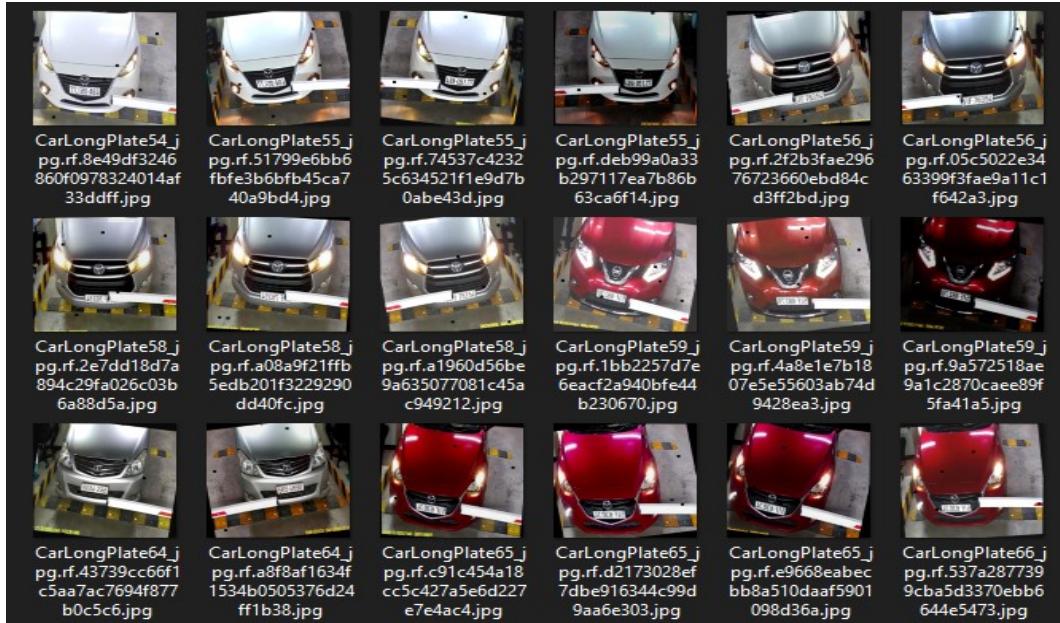


FIGURE 2.4 – Dataset images

2.3.2.2 Labels

| 0 0.62109375 0.7326166699156856 0.140625 0.07507382001909005

(a)

| 0 0.5880071193331544 0.7518030219207764 0.15049087939029934 0.08612040802036276

(b)

| 0 0.6232860891807293 0.7760877178044714 0.14731133723582168 0.07585449375348396

(c)

| 0 0.46268426883774366 0.50546875 0.022011355386194253 0.0234375

| 0 0.8360798204601911 0.78125 0.048682939695149585 0.0421875

(d)

| 0 0.425 0.51875 0.0203125 0.0234375

| 0 0.8078125 0.77109375 0.0453125 0.040625

(e)

FIGURE 2.5 – dataset labels

2.3.3 Entraînement et Évaluation du Modèle

2.3.3.1 Entraînement du modèle

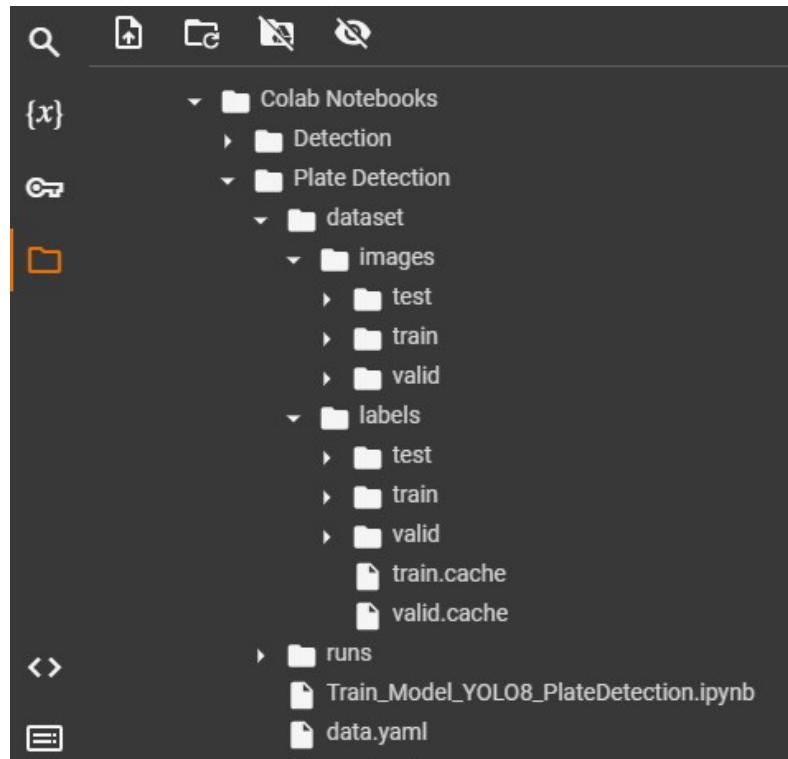


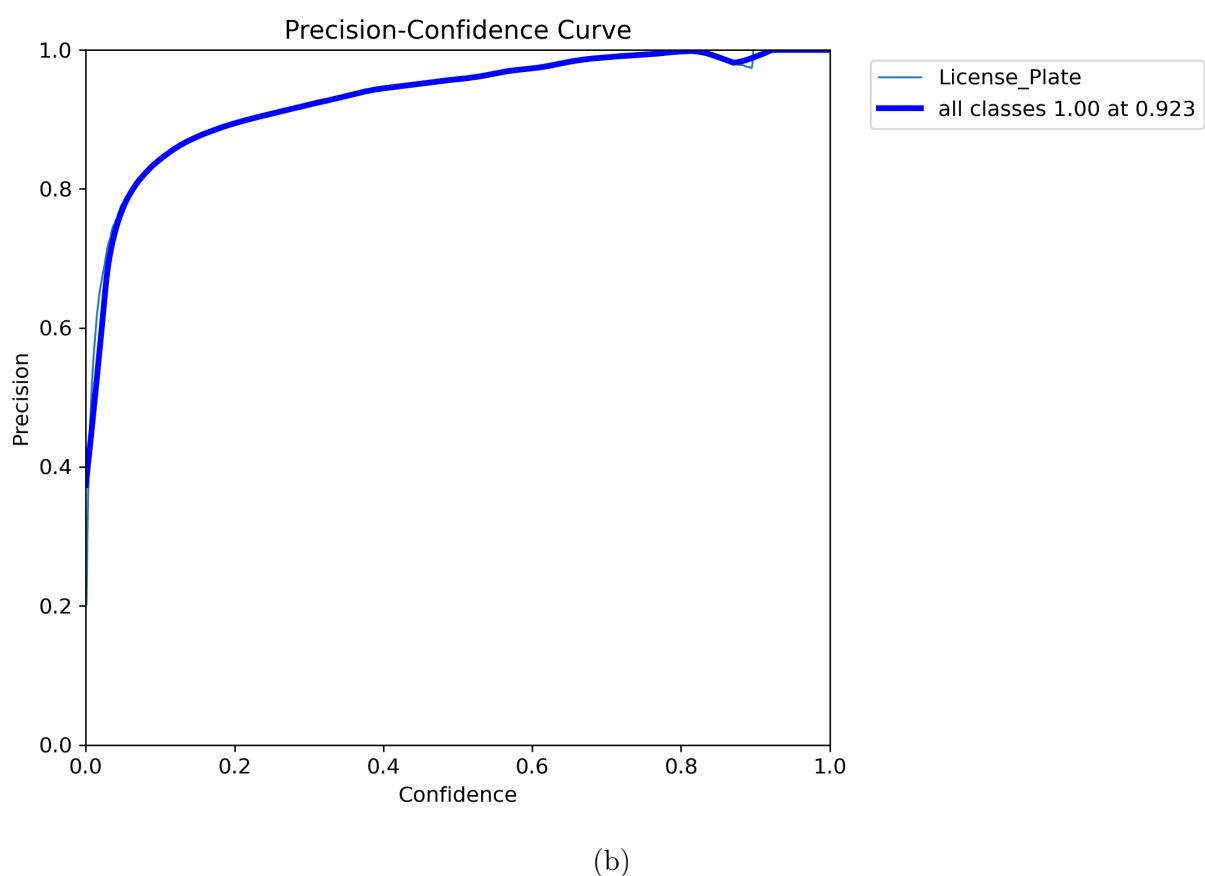
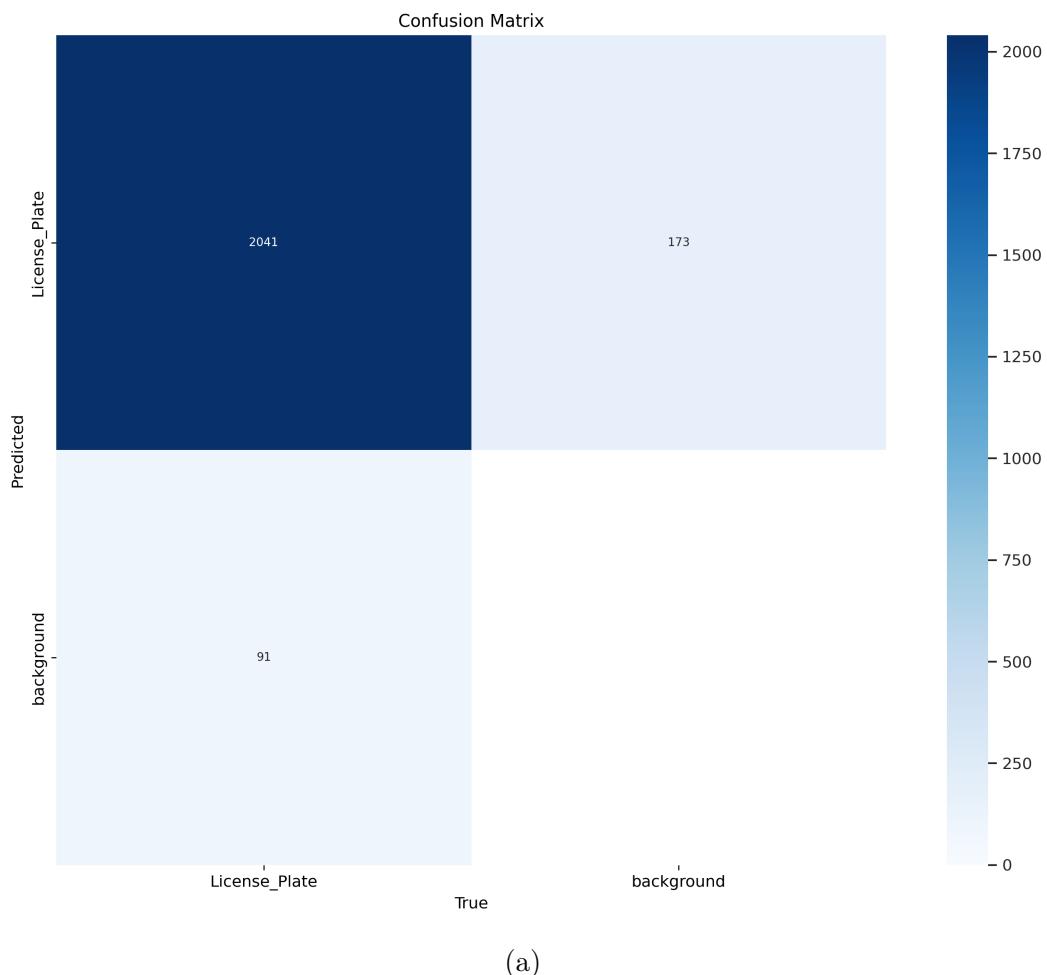
FIGURE 2.6 – schema de projet

L’entraînement du modèle YOLOv8 avec 20 epochs et la configuration définie dans le fichier data.yaml a consisté à exposer le réseau à travers l’ensemble d’entraînement à vingt reprises. Le fichier de configuration correctement spécifié a défini les chemins vers les ensembles d’entraînement, de validation et de test, ainsi que les noms de classe, ici limités à "License_Plate". L’objectif principal de ces epochs était d’optimiser les paramètres du modèle en minimisant la fonction de perte, mesurant la disparité entre les prédictions du modèle et les annotations réelles

```
data.yaml x
1 path : '/content/gdrive/MyDrive/Colab Notebooks/Plate Detection/dataset'
2 train: images/train
3 val: images/valid
4 test: images/test
5
6 names:
7 | 0: License_Plate
```

FIGURE 2.7 – configuration de modèle

2.3.3.2 Évaluation du modèle



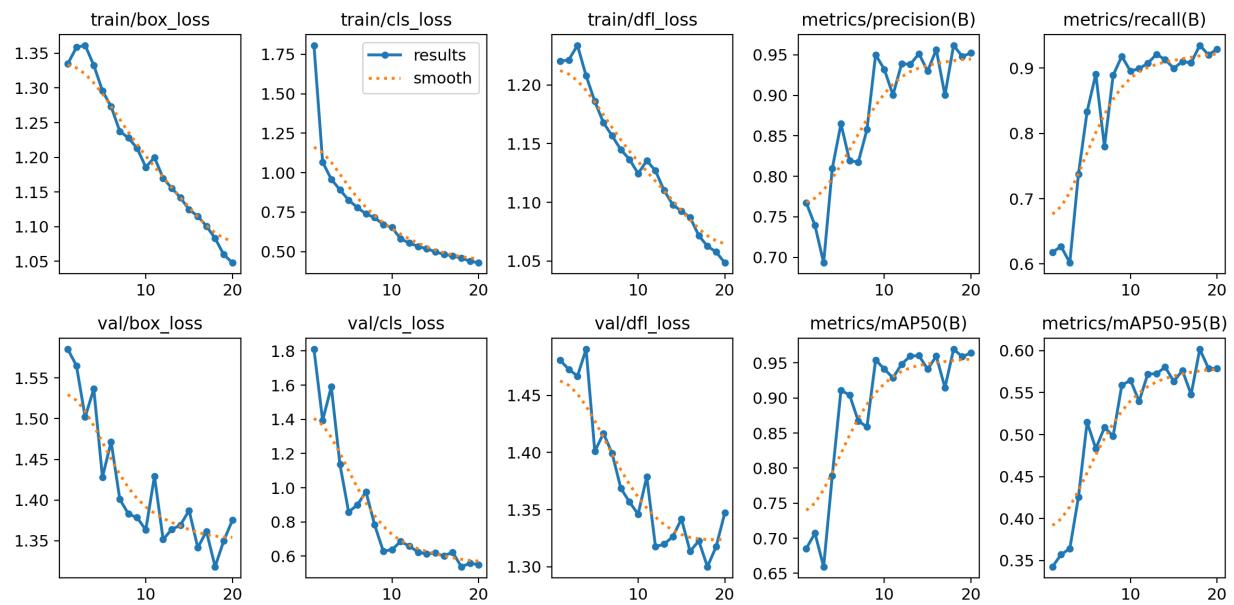


FIGURE 2.9 – résultats de modèle

— CHAPITRE 3 —

LES TECHNOLOGIES UTILISÉES

3.1 Django

Django est un framework web open-source, écrit en Python, qui facilite le développement rapide d'applications web robustes et évolutives. Il suit le modèle de conception MVC (Modèle-Vue-Contrôleur) et offre des fonctionnalités telles que la gestion de base de données, l'authentification utilisateur, et un système de routage pour simplifier le processus de développement.



3.2 Angular

Angular est un framework front-end développé par Google, permettant la création d'applications web dynamiques et interactives. Il utilise le langage TypeScript et suit le modèle de conception MVVM (Modèle-Vue-VueModèle). Angular facilite la liaison bidirectionnelle des données et offre des fonctionnalités telles que l'injection de dépendances et la gestion des états.



3.3 mysql

MySQL est un système de gestion de base de données relationnelle open-source largement utilisé. Il offre une solution fiable pour stocker, organiser et récupérer des données de manière efficace. MySQL prend en charge le langage SQL (Structured Query Language) et est souvent utilisé en tandem avec des applications web pour la gestion des données.



3.4 git/github

Git est un système de contrôle de version décentralisé qui permet de suivre et de gérer les modifications du code source lors du développement logiciel. GitHub, quant à lui, est une plateforme de gestion de projet basée sur Git, permettant la collaboration entre développeurs, le suivi des problèmes et le partage de code.



3.5 Arduino

Arduino est une plateforme matérielle open-source, composée de cartes électroniques et d'un environnement de développement logiciel. Elle est largement utilisée dans le prototypage électronique et l'Internet des objets (IoT), offrant une approche conviviale pour les amateurs et les professionnels de l'électronique.



3.6 Raspberry Pi 4

Le Raspberry Pi 4 est un ordinateur monocarte abordable et compact. Il est utilisé dans une variété d'applications, y compris les projets informatiques personnels, l'automatisation résidentielle et l'apprentissage de la programmation. Sa polyvalence et ses performances le rendent populaire pour des projets informatiques divers.

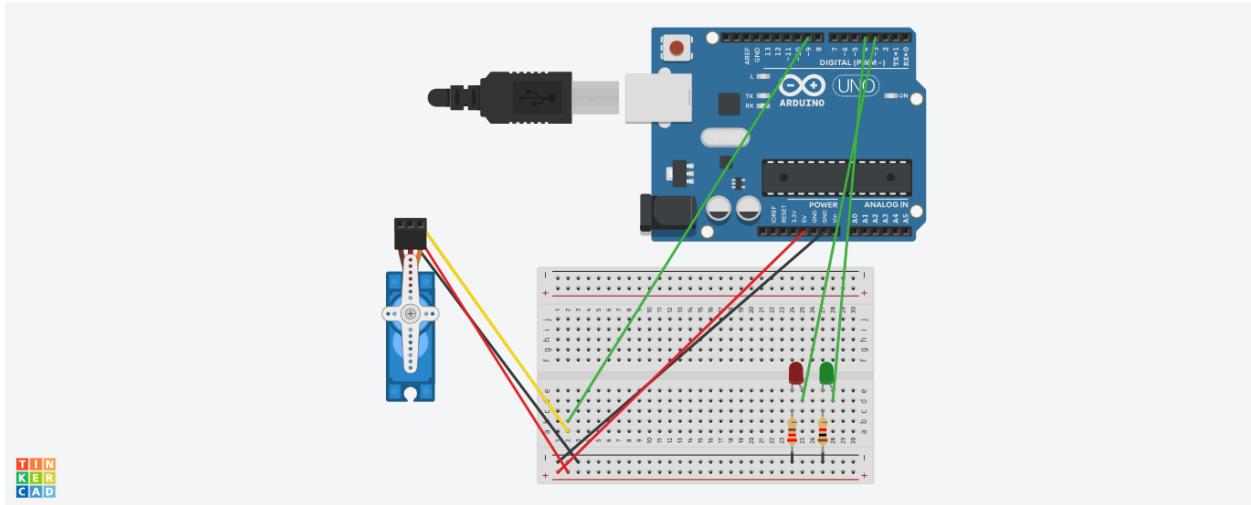


3.7 Servomoteur

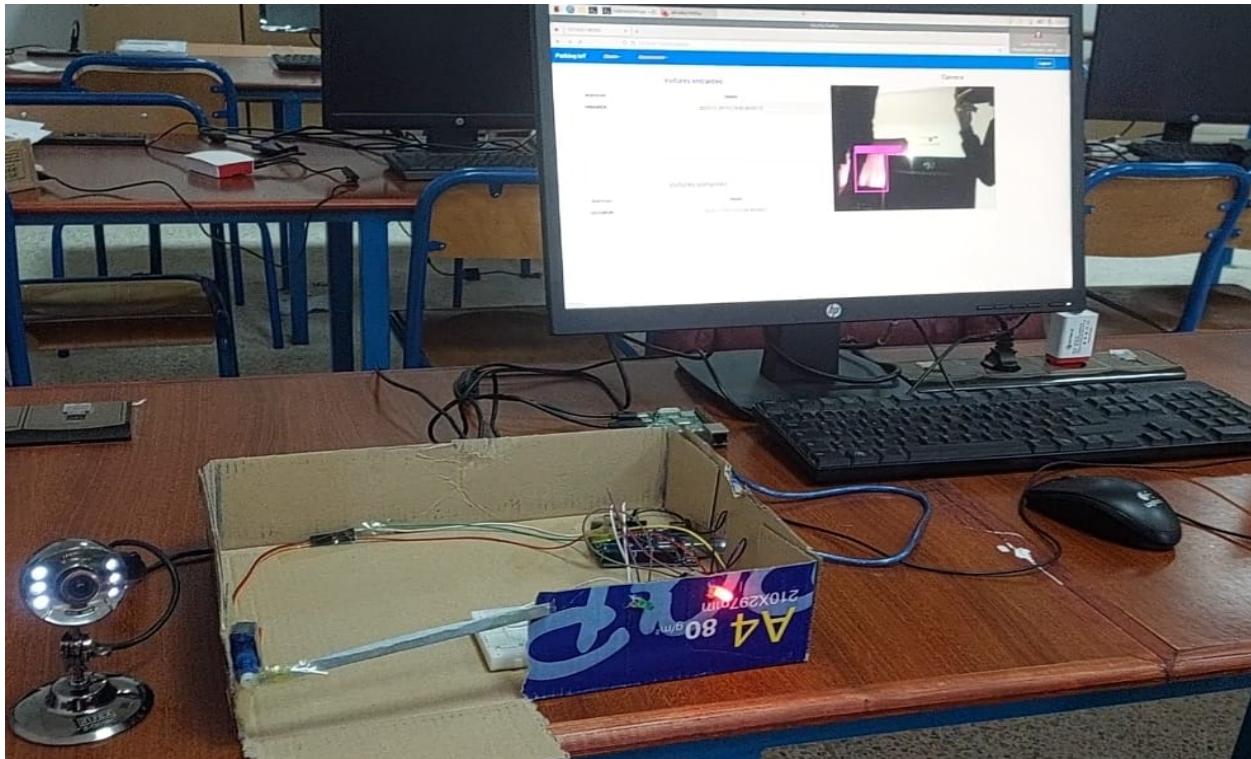
Un servomoteur est un moteur électrique équipé d'un mécanisme de rétroaction, permettant un contrôle précis de la position angulaire. Il est souvent utilisé pour le mouvement précis dans les projets robotiques, les systèmes de contrôle automatique et d'autres applications nécessitant une rotation contrôlée.



3.8 Montage dans tinkercard



3.9 Montage dans la réalité



— CHAPITRE 4 —

FONCTIONNALITÉS DE "SMARTPARKING"

4.1 Introduction

Notre SmartParking se distingue par ses fonctionnalités intelligentes qui redéfinissent la gestion des stationnements. Cette section vous guidera à travers une démonstration complète, mettant en évidence la convivialité et l'efficacité opérationnelle de notre système innovant.

4.2 Authentification et Interface Administrative

Pour amorcer l'expérience SmartParking, l'administrateur s'authentifie pour accéder à l'interface administrative. Cette plateforme centralisée offre une vue complète, avec l'affichage de la caméra d'un côté et un tableau de surveillance affichant les détails des véhicules entrants et sortants de l'autre, accompagné des heures correspondantes.



The image shows a wireframe-style user interface for authentication. At the top, a header bar contains the text "Authentication". Below this, there are two input fields: "UserName" and "Password", each with a corresponding text input box. At the bottom of the form is a blue rectangular button labeled "Login".

FIGURE 4.1 – Authentification

4.3 Gestion des Abonnements et Contrôle d'Accès

Une détection de plaque d'immatriculation non associée à un abonnement actif est un cas d'école. Dans ce scénario, la barrière du parking reste fermée.

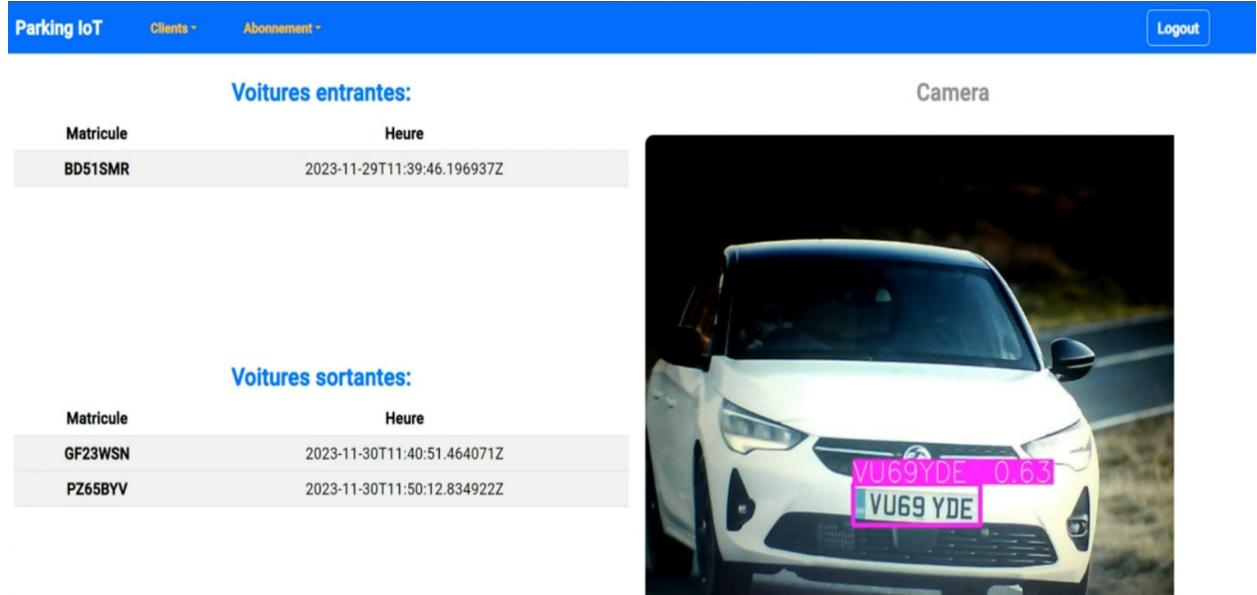


FIGURE 4.2 – Contrôle d'Accès (1)

Ajouter le client via le formulaire dédié et lui attribuer un abonnement mensuel.

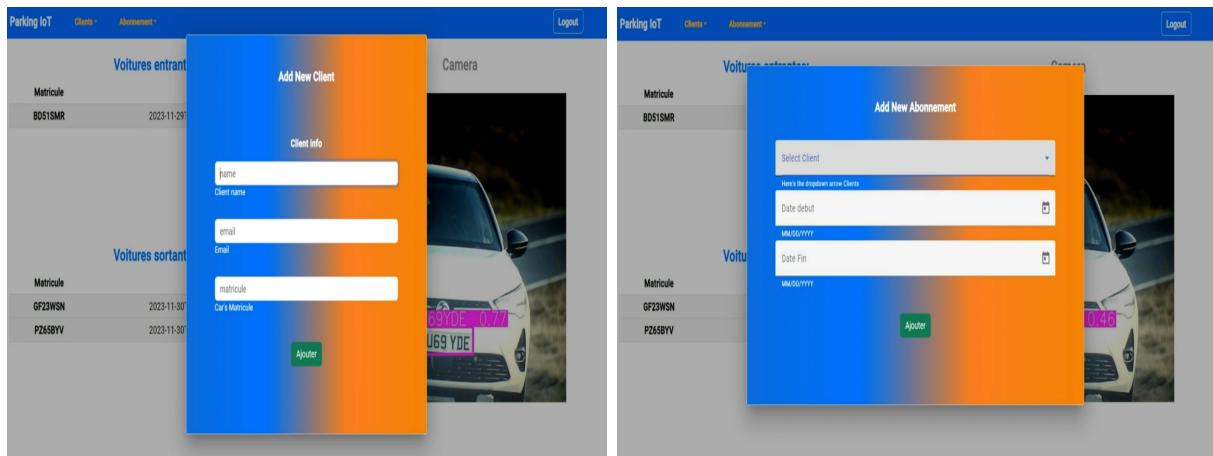


FIGURE 4.3 – Gestion des Abonnements

La plaque d'immatriculation est alors acceptée, ajoutée à la table des véhicules entrants, et la barrière s'ouvre pour permettre l'accès au parking.

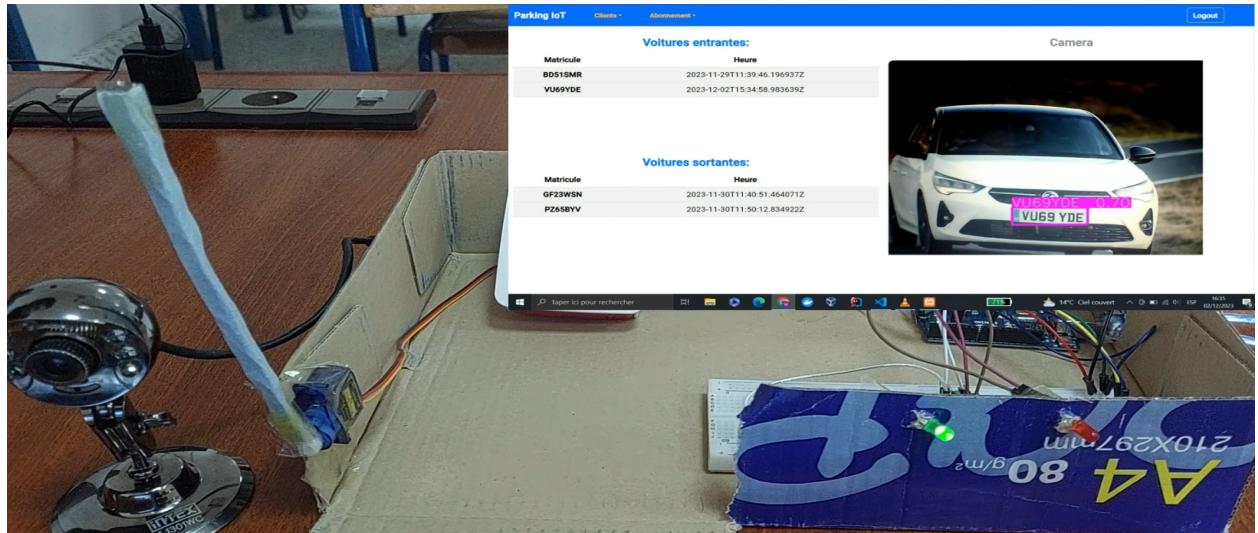


FIGURE 4.4 – Contrôle d'Accès (2)

4.4 Suivi Clients

Explorez la liste des clients ainsi que les détails de chaque client directement depuis la liste. Un simple clic vous donne accès à toutes les données associées, incluant les spécificités de l'abonnement ainsi que la chronologie précise des heures d'entrée et de sortie du client.

Clients	Abonnement	
ClientId	ClientName	N°.Matricule
5	mohammed	BD51SMR
6	said	GF23WSN
7	client1	PZ65BYV
8	mohammed daouan	VU69YDE

FIGURE 4.5 – Liste clients

The screenshot shows a user interface for managing client information. At the top, there's a section titled "Client Informations" with fields for "Client Name" (mohammed daaouan), "Email" (medDaa@gmail.com), "Car Id" (8), "Matricule" (VU69YDE), and "Abonnement". Below this, a table lists entry/exit times. A prominent orange button labeled "Prolongement" is located at the bottom right of the main content area.

Temp d'Entrée	Temp de Sortie
2023-12-02T15:26:32.154746Z	
2023-12-02T15:26:32.154746Z	2023-12-02T15:26:32.597568Z
2023-12-02T15:26:33.400520Z	
2023-12-02T15:26:33.400520Z	2023-12-02T15:26:33.839192Z
2023-12-02T15:26:34.243146Z	

FIGURE 4.6 – Informations client

Notre système intègre une fonctionnalité de prolongement d'abonnement. Cette option intuitive permet au admin de prolonger la durée d'abonnement en quelques clics seulement.

This screenshot shows the same interface as Figure 4.6, but with a modal dialog box overlaid. The dialog is titled "Update Abonnement" and contains fields for "Select Client" (mohammed daaouan), "Date debut" (12/1/2023), and "Date Fin" (2/1/2024). A green "Update" button is at the bottom. The background of the main page is dimmed.

FIGURE 4.7 – Prolongement

4.5 Génération de Rapports et Envoi d'Emails

2023-12-02T15:26:37.916656Z		
2023-12-02T15:26:37.916656Z		2023-12-02T15:26:38.345591Z
2023-12-02T15:26:38.768691Z		
2023-12-02T15:26:38.768691Z		2023-12-02T15:26:40.344762Z
2023-12-02T15:26:40.762523Z		
2023-12-02T15:26:40.762523Z		2023-12-02T15:26:41.782664Z
2023-12-02T15:26:42.605892Z		
2023-12-02T15:26:42.605892Z		2023-12-02T15:26:43.427122Z
2023-12-02T15:26:43.816429Z		
2023-12-02T15:26:43.816429Z		2023-12-02T15:26:45.133833Z
2023-12-02T15:26:45.573103Z		
2023-12-02T15:26:45.573103Z		2023-12-02T15:26:46.197350Z
2023-12-02T15:26:46.500872Z		

[Exporter](#) [Send Email](#)

FIGURE 4.8 – Exportation de Rapports et Envoi d'Emails

La fonctionnalité de génération de rapports constitue le pilier analytique de notre système. Elle offre la possibilité de créer des rapports détaillés, fournissant des insights pertinents sur l'utilisation du parking, les tendances de fréquentation, et d'autres métriques cruciales.

2023-12-02T15:26:36.442430Z		
2023-12-02T15:26:37.916656Z		
2023-12-02T15:26:37.916656Z		
2023-12-02T15:26:38.768691Z		
2023-12-02T15:26:38.768691Z		
2023-12-02T15:26:40.762523Z		
2023-12-02T15:26:40.762523Z		
2023-12-02T15:26:42.605892Z		
2023-12-02T15:26:42.605892Z		
2023-12-02T15:26:43.816429Z		
2023-12-02T15:26:43.816429Z		
2023-12-02T15:26:45.573103Z		
2023-12-02T15:26:45.573103Z		
2023-12-02T15:26:46.500872Z		

[Exporter](#) [Send Email](#)

FIGURE 4.9 – Exportation de Rapport

En parallèle, notre système assure une communication proactive avec les clients grâce à l'envoi d'emails. Les notifications informatives concernant la fin imminente d'un abonnement sont ainsi transmises aux clients. Cette approche proactivité encourage les utilisateurs à renouveler rapidement leurs abonnements, contribuant ainsi à une gestion fluide et efficace du stationnement.

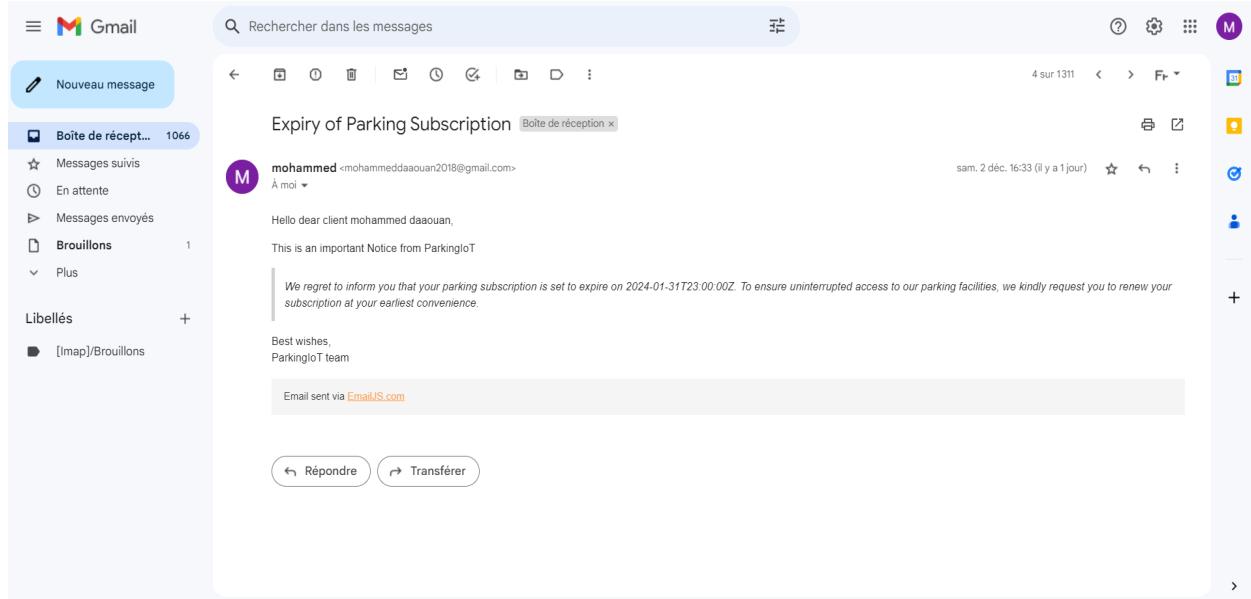


FIGURE 4.10 – Envoi d’Email