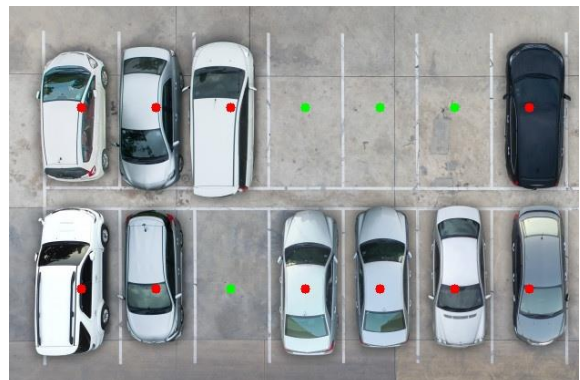


**Année Académique
2020 - 2021**

Développement d'une application pour drone aérien permettant la vérification de places libres dans un parking et suivi sur un site web

TRAVAIL DE FIN D'ETUDES PRESENTE EN VUE DE L'OBTENTION DU DIPLOME
DE BACHELIER EN INFORMATIQUE ET SYSTEMES ORIENTATION
TECHNOLOGIE DE L'INFORMATIQUE



AZZOUZ Souhaïb

3TL2 | RAPPORTEUR : BOUTERFA YUCEF

Remerciements

Je tiens tout d'abord à remercier Monsieur **Sidi Khaouani**, pour l'idée du projet, m'avoir fait confiance et pour avoir été un client qui proposant des solutions mais acceptant également les changements afin d'aboutir au projet.

Je pense également à Monsieur **Francois Wielant** et Monsieur **Etienne Huens** pour m'avoir donné du temps ainsi que mis à ma disposition les locaux et équipements de l'UCLouvain.

Pour finir, je remercie l'EPHEC, et particulièrement Monsieur **Youcef Bouterfa** pour l'accompagnement dont j'ai pu bénéficier. Il a été mon rapporteur et même si nous n'avons pas spécialement eu l'occasion de nous entretenir en dehors des retours de stage, il a su me donner des conseils me permettant d'arriver à bout du TFE.

Table des matières

Introduction	5
Problématique.....	5
Solutions existantes	5
Solution proposée	6
Analyse.....	7
Cahier des charges.....	7
Présentation du client	7
Présentation du projet.....	7
Intervenants	7
Objectif Client	7
Demandes Fonctionnelles.....	8
Ouvert à tous	8
Utilisation de drone	8
Sauvegarde	8
Besoin client.....	8
Charte graphique et Contraintes	8
Cible/Utilisateurs.....	8
Analyse technique	9
Choix des langages.....	9
Choix de l'hébergement	10
Choix des outils et logiciels.....	10
Choix de l'équipement	11
Analyse concurrentielle.....	12
Corps du Travail.....	14
Application Web.....	14
Onglet « Accueil ».....	15
Onglet « Parking ».....	15
Onglet « A Propos »	15
Onglet « Contact ».....	15
Frontend.....	16
Backend.....	16
Sécurité.....	16
Base de données	17
Entité « Parking »	17
Récupération de données.....	17
Gestion et Administration	18

Tests	18
Contour.....	18
Perspective.....	19
Détection de slot.....	19
Détection de coin.....	20
Petits tests	20
Script principal	21
Drone et composants	25
Premier choix	25
Choix final.....	26
Contact avec le client	28
Méthode de travail.....	28
Recommandations	29
Problèmes rencontrés	29
Améliorations futures.....	29
Parking Maximal	30
Confidentialité.....	31
Cybersécurité.....	31
Sécurité physique	31
Contrainte législative.....	32
Assurances.....	32
Licences et Formation	32
Aspect RSE	33
11 ^e ODD	33
13 ^e ODD	33
15 ^e ODD	34
Conclusion.....	34
Bibliographie.....	35

Introduction

Problématique

De nos jours, la voiture est un moyen de locomotion qui assure une grande partie des déplacements quotidiens. Le problème : plus il y a de voitures, plus compliquée et longue sera la recherche d'une place de parking. Moins de temps et de distance à parcourir pour trouver un lieu de stationnement, c'est moins de CO₂ dans l'atmosphère!

Une simple constatation : En diminuant le temps de recherche d'une place de parking, on diminuera fortement le taux de CO₂ dans l'atmosphère.

C'est alors que m'est venue l'idée de créer une petite application permettant le calcul des places libres dans un parking via des images de drones aériens, et de suivre les résultats sur un site web.

Dans ce projet, il est, entre autres, question de réduire cette pollution urbaine en facilitant la recherche de place de parking et ainsi réduire le temps de circulation et la pollution des voitures.

Solutions existantes

En ce qui concerne la recherche de places de parking, quelques solutions ont été mises en place. J'ai pu effectuer quelques recherches pour me rendre compte de ce qu'il existait et la manière dont ces solutions procèdent pour y arriver.

Par exemple, une des solutions est proposée par AeroSpaceLab. Cette société basée à Mont-Saint-Guibert et Leuven, s'engage à avoir un impact positif sur les entreprises à l'échelle mondiale en rendant l'intelligence géospatiale exploitable et abordable. Ils travaillent notamment avec des micro-satellites pour pouvoir faire de la détection d'objets. Ils analysent et améliorent les évaluations stratégiques, économiques et environnementales avec des signaux en temps quasi réel. Leur travail peut donc aider, en se focalisant sur un parking, à trouver des emplacements libres en vendant les données récoltées afin qu'elles les traitent. (cfr Bibliographie : AeroSpaceLab)

Une autre solution que j'ai pu trouver est implémentée dans la commune de Péruwelz. Un article Le Soir daté du dimanche 2 mai 2021, m'a fait découvrir celle-ci. Cette commune a placé dans le parking de sa Grande-Place une caméra intelligente située à 6 mètres de haut afin d'analyser les habitudes de stationnement à cet endroit. Elle peut donc dire si une voiture est garée ou non et si oui depuis combien de temps. (Le Soir du dimanche 2 mai 2021 :cfr Bibliographie : Péruwelz).

Pour élaborer les grandes lignes de mon TFE, j'ai analysé la solution proposée par Aerospacelab et j'ai remarqué que certaines fonctionnalités étaient présentes dans certains services, mais tout n'était pas abordable pour un particulier. Il fallait alors trouver une solution qui soit ouverte à tous, qu'un utilisateur lambda y accède gratuitement. Une solution qui soit conforme à la demande du client.

Solution proposée

Après accord et premières discussions avec le client, il était question de concevoir, fabriquer et tester un drone permettant la recherche de places de parking.

Ce projet allait se diviser en deux parties.

La première consiste en la conception d'un drone qui permet de vérifier s'il y a des places libres dans un parking à l'aide d'un système de rayons infrarouges. Le drone étant petit, il ne serait pas possible d'y attacher une caméra sachant que le poids compte pendant le vol.

La seconde partie consiste à transmettre l'information du drone au site web ou à l'application mobile afin qu'elle puisse être affichée et consultée par les utilisateurs du parking.

Après quelques recherches sur la conception du drone et par manque de temps, on a dû transformer notre première idée et se focaliser sur la question principale du TFE.

La demande de mon client était simple : trouver un moyen de s'avoir avant d'arriver dans un parking s'il avait la possibilité de s'y garer.

Nous nous sommes convenu d'une nouvelle solution : ne plus mettre en avant le drone mais l'application permettant la reconnaissance de place de parking.

Nous avons donc partagé le travail en trois parties dont deux importantes.

Premièrement, cette application reçoit une image d'un parking, la traite, détecte les places libres et envoie ces informations à la base de données du site.

Deuxièmement, l'application permet de récupérer les informations de la base de données et les affiche sur un site web qui, à l'aide d'une interface responsive pourra être consulté autant sur un ordinateur que sur un téléphone. Le site web possède plusieurs fonctionnalités qui permettent d'améliorer le visuel et donnent ainsi envie à un utilisateur de le visiter. La fonctionnalité principale reste l'affichage sur une map d'un point avec la date de la prise d'image ainsi que le nombre de places libres. Une autre fonctionnalité est de pouvoir envoyer un mail à l'aide du formulaire de contact avec un message personnalisé. D'autres possibilités que nous découvriront dans la suite de ce rapport sont présente sur le site.

La dernière partie de ce travail consiste à automatiser la capture d'images afin que le site fonctionne sans intervention humaine. L'idée est de connecter le drone à un ordinateur et le travail de l'ordinateur consiste à envoyer les positions géographiques au drone qui ira faire une capture à l'emplacement défini. Cette partie étant la moins importante, ne sera pas d'application pour la réalisation du TFE et sera une piste d'amélioration.

Analyse

Cahier des charges

Présentation du client

Mon client est Sidi Khaouani, riverain dans la commune de Wavre. C'est une personne avec qui j'ai énormément de contact car il fait partie de mon groupe d'amis proches.

Il suit une formation en Conseiller en solution d'objets connectés et a également fait la formation en Conseiller Technique PC-Réseau au centre IFAPME de Charleroi. Il était aussi un élève de l'EPHEC durant deux années et a vu les cours que l'on a pu suivre.

Présentation du projet

On a pu constater avec mon client la difficulté de trouver une place de parking lors de certaines balades. Il m'a donc proposé de faire, en vue du TFE, un projet qui permettra de trouver un moyen de savoir, avant d'arriver dans un parking, s'il est possible de s'y garer. Ce projet utilisera un drone qui pourra être une valeur sûre vu que mon client a une connaissance dans le domaine des objets connectés. Ce drone pourra soit être conçu, soit utiliser un drone déjà existant capable de capturer des images.

Intervenants

Rôle NOM-Prénom	Client KHAOUANI Sidi	Développeurs AZZOUZ Souhaïb
Mail	khaouani.sidi@hotmail.com	s.azzouz@students.ephec.be
Ville	Wavre	Wavre

Objectif Client

Le projet a pour but de permettre à n'importe quelle personne de connaître, avant d'arriver dans un parking s'il pourra s'y garer. Dans le cas contraire, l'utilisateur prendra l'initiative d'aller voir un autre parking. Quand l'application et le site seront fonctionnels, il faudra essayer de les implémenter sur différents parkings dont celui en face de chez lui.

Demandes Fonctionnelles

Ouvert à tous

Le but de ce projet est de pouvoir permettre à tous de pouvoir accéder aux données traitées par l'application. Avoir les services des autres concurrents mais avec une simplicité d'affichage et disponible gratuitement.

Utilisation de drone

Le drone prendra une place importante dans le monde de demain. Le fait d'utiliser un drone actuellement permet d'anticiper les évolutions et les mises à jour à venir. Le drone ne doit pas être spécialement conçu par le développeur. Il sera possible d'utiliser des drones tout fait afin d'alléger le travail et se concentrer sur le principal. En utilisant un drone, il sera possible de prendre des captures aériennes et donc éviter d'avoir des données confidentielles tel que les plaques d'immatriculation.

Sauvegarde

Le développeur prendra l'initiative de ne pas sauvegarder les images récupérées après les avoir traitées. De cette manière, nous évitons les questions du droit à l'image. Le projet fonctionnant sans le consentement de toutes les personnes utilisant un parking, le fait de ne pas sauvegarder ces dernières évite des problèmes juridiques.

Besoin client

En tant qu'utilisateur, je voudrai :

- Avoir accès aux informations gratuitement
- Avoir accès aux informations sur autant sur un ordinateur qu'un smartphone
- Avoir une interface claire et rapide d'utilisation

En tant que client, je voudrai :

- Avoir un débriefing des avancées
- Que l'accès aux informations du site soit ouvert à tous, mais l'application python utilisable uniquement par un responsable de parking

Charte graphique et Contraintes

Il ne doit pas y avoir de surplus d'informations sur le site. Vu l'utilisation de ce dernier, l'information doit être rapide d'accès et avec une interface claire.

Pas de code couleur imposé et l'application ayant pour but premier la réalisation du projet, pas de contraintes de langages tant que l'application respecte les différents objectifs.

Prendre connaissance des différentes contraintes liées à un drone et prendre contact avec la commune de Wavre pour savoir s'il est possible de faire les tests dans leurs parkings, sans quoi utiliser le drone sur un autre lieu accessible.

Cible/Utilisateurs

La cible du projet est toutes personnes souhaitant trouver une place de parking. Grâce à cette nouvelle application, la circulation dans les villes sera plus efficace et nous réduirons ainsi le taux de CO2.

Analyse technique

Choix des langages

Pour l'application, je me suis tourné vers le Python. C'est un langage de scripting très répandu, simple et adapté aux applications web. Le langage est populaire et possède une bibliothèque très riche. Nous pouvons travailler sur des intelligences artificielles, faire des affichages graphiques. Le développement de la partie backend d'un site est également faisable, nous pourrions de cette manière rapidement transférer l'information à la base de données.



Le traitement d'image devient de plus en plus populaire en Python et beaucoup d'entreprises délaissent Matlab pour apprendre à utiliser ce langage. Tout ce que peut faire Matlab peut être fait en Python mais pas l'inverse et c'est donc un choix de facilité mais de vision future. Si les entreprises prennent la décision d'abandonner Matlab pour Python, c'est que ce langage a un bon avenir et qu'il faut s'y consacrer dès maintenant.

Un autre aspect dont je n'ai pas eu le temps de m'y consacrer mais qui pourra, dans le futur du projet, être implémenté : le script détecte les places vides via une intelligence artificielle ou le deep learning. L'idée étant possible et afin d'éviter de perdre du temps, utiliser dès départ Python est la meilleure opportunité.

En ce qui concerne le langage Frontend, je me suis tourné vers le développement du site en HTML, CSS et JS avec du Bootstrap. Ce sont des langages facile et rapide à mettre en place. En vue du temps qu'il me fallait pour faire le site, j'avais une préférence sur ceux-ci. Il n'est pas utile d'utiliser des plugins ou d'installer plusieurs éléments différents, un éditeur de texte est amplement suffisant pour faire toutes les pages que l'on souhaite. La communauté est grande et c'est un avantage lorsqu'il s'agit de trouver des solutions pour un problème.



Pour des raisons pratiques, j'ai décidé de partir sur une application web mono page (Single-Page Application ou SPA) afin d'éviter des chargements de pages après chaque interaction et ainsi fluidifier l'utilisation du site.



Pour faire simple, une SPA permet de naviguer entre les différents onglets d'un site, sans pour autant que la page entière soit rafraîchie. Pour ce faire, une manipulation se fait lors du clic des onglets qui affiche et cache les composants du site.



Mon choix au niveau du Backend se tourne vers le PHP. Les avantages de celui-ci sont la facilité à le marier avec l'HTML, la popularité, le dynamisme, les performances, la rapidité d'affichage ... Plusieurs Frameworks utilisent le PHP comme langage de base. On peut y retrouver notamment Laravel, Symfony, Cake PHP ... Si plus de la moitié des sites tournent en PHP, c'est que ce langage n'est pas si mauvais après tout. Nous pouvons trouver une documentation complète sur le site de PHP et facilement trouver solution à un problème. Pour finir, le PHP s'intègre facilement à une base de données. De cette manière le projet peut être mis en place plus rapidement. (cfr Bibliographie : PHP)



En parlant de base de données, j'ai opté pour la SGBD la plus répandue : MySQL. Elle est fournie avec la plupart des hébergements. Pour des raisons de temps, c'est la solution qui n'allait pas me faire perdre du temps. La base de données étant pour très peu et pas besoin d'avoir de nombreuses fonctionnalités, il fallait trouver une chose light et performante. MySQL est également une SGBD gratuite contrairement à par exemple Oracle.



Choix de l'hébergement

La solution d'hébergement qui me convenait était située sur la plateforme LWS. C'est une solution qui a un bon rapport qualité/prix et la plateforme offre un nom de domaine et j'ai pu en choisir une extension en « .be »

Lws offrait les outils nécessaires afin d'implémenter correctement le site web, à savoir la base de données, le nom de domaine, un serveur mail... En ce qui me concerne, je n'ai pas perdu du temps sur l'implémentation du site web. Cependant, une erreur qui est survenue était de permettre à l'application python de se connecter à la base de données. Pour ce faire, le script doit tourner dans ma machine locale et l'adresse IP de celle-ci est la seule pouvant accéder à la base de données.



L'hébergeur peut faire tourner des scripts python mais pour une question de confidentialité et ainsi donner la possibilité au propriétaire des parkings de lancer le script à sa convenance, il reste en local.

Choix des outils et logiciels

Liste des outils utilisés :

PyCharm : cet IDE, dont la licence est offerte par l'EPHEC a été utilisé afin de créer le script en python. L'ayant déjà utilisé lors de précédents projets, il était idéal pour moi de rester dessus. Les plugins peuvent facilement être ajoutés et le développement est facilité avec l'indication des différentes erreurs.



WebStorm : cet IDE, dont la licence est également offerte par l'EPHEC. Il est facile d'utilisation. Il ne fallait pas énormément de fonctionnalité mais sait dire l'endroit où il manque un élément et le développement de page HTML devient un jeu d'enfant.



WampServer : afin de faire fonctionner un serveur localement les scripts PHP. Ainsi, les tests sont faits localement et lorsque tout est correct, on peut faire une implémentation du site sans erreur.



MySQL Workbench : je l'ai plus souvent utilisé afin de manipuler la base de données, visualiser, modifier ou gérer celle-ci.

phpMyAdmin : beaucoup moins utilisé que le précédent, mais lorsque je devais visualiser une information en vitesse, j'avais un lien direct de mon hébergeur pour y accéder et je l'utilisais donc pour de simples vérifications.

A.R. FreeFlight : cette application mobile m'a permis de contrôler un drone Parrot AR 2.0 et faire des captures photos et directement les enregistrer dans ma galerie.



Github : il n'a été utilisé qu'à la fin du développement pour avoir une plateforme pour déposer les codes en ligne.



Trello : utilisé afin de résumer les différentes tâches à faire ainsi que pouvoir donner un aperçu au client pour voir l'état d'avancement.



Lucidchart : c'est une plateforme web assez qui permet de modéliser des schémas et, dans ce cas-ci, la base de données.



Photoshop : ce logiciel très connu m'a permis de modifier des images afin de faire des tests pour le projet.



Choix de l'équipement

Pour permettre de faire des tests avec un drone aérien, j'ai pu avoir à ma disposition un AR drone Parrot 2.0. Celui-ci a plusieurs avantages et certains inconvénients.

Premièrement, si nous voulons avoir une vision future du projet, ce drone fonctionne correctement avec le logiciel libre Robot Operating System (ROS). Cet outil fonctionne sur un ordinateur et permettra de faire fonctionner le drone via celle-ci. De cette façon, on pourra automatiser le drone et lui donner des positions géographiques aux endroits où les captures doivent être prises. Cela permettra de ne pas avoir d'interventions humaines.



Ce drone possède deux caméras embarquées : une frontale et une ventrale. La qualité de la première supérieure à la deuxième (720p vs 320p). Sachant que nous prenons des captures du ciel, il faudra faire une manipulation pour que la caméra frontale se dirige vers le bas, car la résolution de l'autre caméra est trop faible pour le traitement de ses images. (cfr Corps du texte : Drone)

De plus, les parties électroniques visibles sous le drone, sont recouvertes d'une fine pellicule étanche afin de protéger celui-ci d'une rosée ou de la neige et protéger les composants.

Pour finir, ce drone possède une stabilisation automatique à haute altitude. Cette fonctionnalité permettra de prendre des photos sans être flou. Sur le papier, cela facilite également les tonneaux avec le drone (bien que cette utilité ne soit pas intéressante pour le projet, il est toujours bon de le savoir).

Analyse concurrentielle

Après quelques recherches, peu de concurrents sont connus. Quelques questions ont été posées afin de voir les avantages et inconvénients sur ce projet.

Pour ce faire, nous avons identifié deux concurrents, AeroSpaceLab et le projet de la commune de Péruwelz, sans tenir compte que ce dernier a été découvert bien après que le TFE soit entamé.

Questions	Nom Projet	AeroSpaceLab	Commune de Péruwelz	Mon TFE
Où se situe le projet ?		Dans le monde	A Péruwelz	Dans les parkings acceptant le projet
Avec quelles technologies fonctionne le projet ?		Des Satellites Une Intelligence géospatiale	Une caméra fixe Un programme	Un drone avec caméra embarquée Un script python
Comment accéder aux résultats ?		En demandant un partenariat	Via un Dashboard accessible uniquement à la commune	Sur un site web (www.2par.be)
Quelles sont les forces du projet ?		-Réponse en temps réel -Vol du satellite impossible	-Projet déjà avancé -Déjà en fonction -Caméra bonne qualité	-Le drone survole tout le parking -Résultat disponible à tout le monde
Quelles sont les faiblesses du projet ?		-Le coût de la mise en place -Pas spécifiquement lié aux parkings	- Caméra fixe donc certaines places cachées -Résultat pas disponible au public	-Drone sur batterie -Limité à une caméra de drone

Lorsqu'on regarde le tableau , dans les trois projets, il y a des fonctionnalités propres à eux-mêmes. Ces différents services proposent des avantages et des fonctionnalités bien développées tout en répondant à des critères. Cependant, il y a toujours des possibilités manquantes et qui sont selon moi indispensables pour le projet. Le but principal est que l'utilisateur pourra avoir l'information du nombre de places libres dans un parking et dans les deux concurrents, cette information n'est pas disponible au public.

L'avantage du drone à la caméra fixe : pouvoir prendre tout le parking et ne pas perdre des places dues au mauvais positionnement de la caméra. Lorsque celle-ci est placée à un endroit, les places se situant en dessous ainsi que celles cachées par un objet ne sont pas prises en compte. Le fait d'utiliser un drone permet de prendre toutes les places et de faire les calculs dans le script.

Pour finir, il est bon de préciser que le développement de cette solution ne sera utilisé que pour les heures de pointe et les parkings souvent remplis. Il est inutile de l'implémenter dans les endroits souvent vides. L'avantage à grande échelle permettra de réduire drastiquement le taux de CO2.

Corps du Travail

Le projet se divise en deux grandes parties. La première étant accessible publiquement via une application web, l'utilisateur peut ainsi consulter la carte des parkings, les informations liées au projet/ entreprise et pouvoir contacter cette dernière afin de demander des renseignements, proposer une aide ou un endroit où implémenter le projet...

L'application web est accessible via l'adresse suivante : <https://2par.be/>

La seconde partie concerne la gestion et l'administration des parkings. C'est le travail donnant les résultats, tout ce que le client ne voit pas et qui permettra d'avoir toutes les informations dont le client aurait besoin. A cet endroit, nous retrouverons, le script de traitement d'images mais également le drone et ses composants, les images récupérées par celui-ci et les recherches liées au projet.

Application Web

Cette application est responsive afin de permettre aux utilisateurs de pouvoir l'utiliser aussi bien sur un ordinateur qu'un téléphone portable. Le but de cette démarche permettra de vérifier les places soit en partant de chez soi (sur un ordinateur) ou quand on est à l'extérieur et qu'on souhaite vérifier sa future place (sur son smartphone).

En arrière-plan, nous retrouvons une vidéo afin d'embellir le site web. La vidéo met en valeur la Voie lactée afin de faire ressortir les valeurs du projet : pour être plus intéressant au niveau du taux de CO₂, on souhaite s'étendre le plus possible et quoi de plus grand que la Voie lactée à notre échelle ? Une autre façon de voir les choses est l'utilisation de drone dans le projet : on se trouve dans le ciel afin de revenir sur la terre. Si on vise encore plus haut, on se retrouve rapidement dans l'espace, d'où la raison de ce choix. Cependant, pour des raisons d'accessibilité du web, j'ai opté pour un bouton qui met la vidéo en pause. Ainsi, les personnes en situation de handicap (exemple : la dyslexie) peuvent percevoir, comprendre, naviguer et interagir sans que l'œil ne soit attiré vers le non-essentiel. La vidéo est libre de droits. (cfr Annexe : Background vidéo)

Pour que le projet soit attrayant, il fallait choisir un nom et un logo. Pour ce projet, j'ai opté pour le nom 2par. Ce nom est survenu suite à l'idée que l'on travaillait sur des PARKing avec un PARrot. Les deux mots commencent par « PAR » et comme il y en a deux : 2PAR. Ce mot est également le verlan, langage de jeune consistant à inverser les syllabes de certains mots (cfr Bibliographie : Verlan), de « partout » étant donné que le but du projet est de se retrouver partout dans le monde.

En ce qui concerne le logo, nous retrouvons une forme de cube où l'on distingue en rouge un P, en jaune un A et en vert un R pour former le mot PAR. Avec un peu d'imagination, le P deviendra un drapeau qui, comme dans les cartes, permet de pointer un endroit sur celle-ci. L'addition entre le A et le R permettra de former des ailes pour schématiser le drone et la barre entre les deux sera la caméra.



En bas de la page, nous trouverons un message pour le copyright ainsi que la personne qui a fait le design du site web. En cliquant sur mon nom, nous sommes redirigés vers mon portfolio.

Dans l'application web, nous allons retrouver quatre onglets distincts : Accueil, Parking, A propos et Contact.

Onglet « Accueil »

Dans cet onglet, nous entrons dans l'accueil du site web où les utilisateurs sont redirigés lorsqu'ils entrent l'URL. Elle se compose d'un petit texte d'accroche et des images du projet. Dès l'arrivée, une barre de navigation se situe en haut à droite sous deux formes : sur les grands écrans, les onglets l'un à côté de l'autre et sur les plus petits un menu « hamburger » regroupant tous les onglets. Cette barre permettra une navigation fluide sur le site web.

Onglet « Parking »

Nous retrouvons une carte où se situe actuellement un parking afin de montrer que le projet fonctionne. Afin d'afficher une map interactive et qu'on peut y inclure des données sur un point, j'ai utilisé l'API Leaflet. C'est une bibliothèque Open-Source basée sur le JavaScript. Cette bibliothèque est riche et possède plusieurs fonctionnalités de cartographie utile pour le développement. Leaflet apporte une simplicité tout en étant performant et fonctionne sur la plupart des plateformes. Cet API possède un site web bien documenté et assez d'informations afin de créer ses propres cartes.

Nous pouvons donc remarquer, grâce à cette API, que j'ai pu afficher le nom du parking, le nombre de places libres ainsi que la date de la capture d'images. Ces informations sont récupérées dans la base de données et affichées via du PHP.

Onglet « A Propos »

Sur cet onglet, nous retrouvons un texte renseignant sur 2Par et d'où vient l'idée. Elle donne également quelques informations sur les problèmes liés à la voiture et comment l'entreprise permettra de les résoudre.

Onglet « Contact »

L'onglet possède deux éléments. Le premier, à droite, est un petit texte invitant l'utilisateur à envoyer un mail, appeler ou répondre au formulaire s'il possède des questions, suggestions.... La seconde partie est un formulaire où l'utilisateur envoie plusieurs informations et qui seront envoyées à une adresse mail propre au site web. L'utilisateur est dans l'obligation de mettre une adresse mail, un nom et un message. Si ces informations ne sont pas correctement introduites, un message d'erreur est affiché et quand elles sont correctes, il peut envoyer ses réponses via le bouton « Envoyer ». Celle-ci se retrouve alors sur la boîte mail de 2par, answer@2par.be, sous cette forme :

Nouvelle réponse au formulaire



De c1641721c@web49.lws-hosting.com, le 2021-05-19 15:11

[Détails](#)

Vous avez reçu une réponse de: Souhaib Azzouz

Email: sou_sou_azzouz@hotmail.com

Message:

Voici le message du formulaire

Frontend

Au niveau du design, j'utilise la collection proposée par Bootstrap. Il est installé dans l'application grâce à des liens vers les fichiers CSS et JS. Ces liens sont faits au niveau du <head> afin de les inclure dès le chargement de la page, d'autre mis à la fin du code car la page devait être générée complètement avant d'utiliser ces fichiers.

La disposition de l'application est faite afin de répondre à des besoins d'accessibilité du web. Les éléments ajoutés dans celle-ci sont réfléchis pour répondre aux différents besoins et attentes des utilisateurs en situation de handicap mais en gardant en tête qu'il doit rester beau à voir.

Bootstrap permet de faciliter également la mise en place de la responsivité du site sur les appareils mobiles. Il suffit simplement d'inclure dans l'attribut « class » une annotation pour changer la disposition en fonction de la taille de l'écran.

Backend

Le formulaire permettant d'envoyer un mail est vérifié grâce au Javascript. Il permet de simplement vérifier que toutes les valeurs entrées dans celui-ci sont correctes. Quand la vérification est correcte, j'utilise la fonction mail de PHP afin d'envoyer les mails. Ce service, assez simple d'utilisation, permet d'automatiser l'envoi de mail au format HTML en récupérant les variables du formulaire.

Les mails envoyés ne sont pas reçus à l'utilisateur, en outre, il pourra voir afficher sur le site web un accusé de réception afin de confirmer qu'il a bien été dans la boîte mail.

Sécurité

En ce qui concerne la sécurité du site, il y a une mise en place d'un certificat SSL afin de protéger les données en chiffrant les échanges et une redirection obligatoire vers l'HTTPS.

Base de données

(cfr Annexe : Schéma de la base de données)

La première chose que j'ai faite lorsque j'ai commencé à vouloir faire le site web, c'est de m'intéresser à la base de données. Cette base de données ne devait pas avoir un schéma complexe mais il fallait y consacrer du temps et connaître les infos que je souhaitais utiliser et celles que je n'utiliserais pas actuellement mais qui peuvent être intéressantes pour le futur.

Au sein de la base de données, je n'ai utilisé qu'une seule entité afin d'avoir un bon fonctionnement du site web. L'entité n'est utile que pour l'onglet Parking et permettra de récupérer les informations.

Entité « Parking »

Cette entité regroupe toutes les informations qui seront utiles pour le projet. Les différents attributs de celle-ci sont un **id** qui est la clé primaire et qui s'auto-incrémente, un **name** pour le nom du parking qui a une limite de caractères de 45 et par défaut 'sans nom', une **adress** pour le lieu du parking qui a également une limite de caractères de 45 et qui ne peut pas être vide, un **tot_slot** pour le nombre de places total du parking avec une limite d'un nombre à 11 chiffres et qui ne peut pas être vide, un **empty_slot** pour le nombre de places libres dans le parking avec une limite d'un nombre à 11 chiffres et par défaut 'null' et la **date** qui est par défaut 'null'.

Récupération de données

Pour la récupération des valeurs, nous utilisons le PHP. Grâce au PHP, nous entrons dans la base de données, nous prenons l'id le plus grand dont le nom est égal à une valeur. De cette façon, si dans le futur nous utilisons plusieurs parkings, nous serons toujours dans la possibilité de récupérer les valeurs les plus récentes. Par après, on récupère chaque valeur et on les inclut sur la map Leaflet en attribuant les données à des variables en sachant que les positions géographiques du point sont connues à l'avance.

Tests

Pour arriver à un résultat et avoir un script qui fonctionne, j'ai dû faire plusieurs tests. Ce projet étant un code assez complexe nécessitant pas mal de recherches et investissement et ne consiste pas uniquement à la création d'un site web. Il fallait en premier lieu découper la tâche en plusieurs morceaux, prendre chacune de celles-ci au cas par cas et trouver une solution qui fonctionne.

Contour

Le premier test consistait à de la détection de formes sur une image. Grâce à Photoshop, j'ai pu créer une image où se trouvaient des triangles, un carré, un rectangle et un cercle. Ce test m'a permis de comprendre comment faire de la reconnaissance sur une image et de modifier celle-ci afin d'ajouter des annotations.

(cfr Annexe : test.py)

Le fonctionnement de ce script est simple :

- 1) On récupère l'image via OpenCV.
- 2) Transformation de l'image en Gris. La raison ? Sur le code, on travaille sur un seul niveau d'image et non pas sur les niveaux Rouge, Vert et Bleu. Cela permettra de se baser que sur les changements forts de pixel et réduira les ressources nécessaires lorsque l'image sera traitée via `cvtColor` et l'attribut `COLOR_BGR2GRAY`. (cfr Bibliographie : `cvtColor`)
- 3) On rajoute un filtre de Flou Gaussien afin de réduire le bruit dans l'image et enlever certains détails via `GaussianBlur` (cfr Bibliographie : `GaussianBlur`)
- 4) On fait par la suite un seuillage afin de séparer les formes de l'arrière-plan via `threshold`. (cfr Bibliographie : `Threshold`)
- 5) On récupère finalement les contours via `findContours` afin de traiter la variable. (cfr Bibliographie : `findContours`)
- 6) Pour finir, avec un jeu de conditions, on compte le nombre de coins de la forme. En fonction du nombre, on note le nom de la forme et dessine le contour. Si deux formes possèdent le même nombre de côtés (le carré et le rectangle dans ce cas), on divise deux côtés qui se suivent ensemble et si la moyenne récupérée entre les deux se situe entre 0,95 et 1,05, on a un carré, dans le cas contraire, c'est un rectangle.

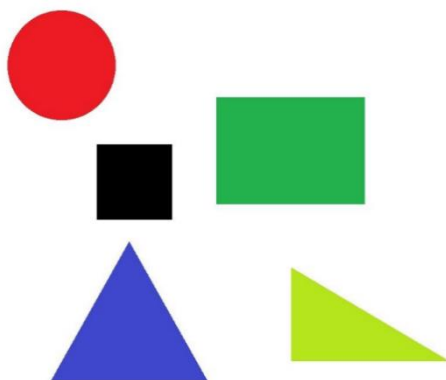


Figure 1 : Formes non travaillées

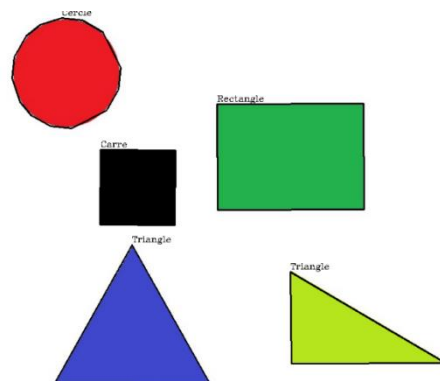


Figure 2: Formes travaillées avec contour et description

Perspective

Le second test qui a été fait, est de modifier les images qui sont présent sur une certaine inclinaison et de modifier afin que l'image soit « plate ». Pour arriver à cette fin, on récupère quatre points sur une image et on remplace la position de ceux-ci par des nouveaux à un meilleur emplacement. On récupère une matrice via les anciens points et les nouveaux ainsi que `getPerspectiveTransform` de `openCV` et on transforme toute l'image grâce à cette matrice via `warpPerspective`. (cfr Bibliographie : Geometric Image Transformations)

Cette solution a finalement été abandonnée par la suite.

(cfr Annexe : `perspective_transformation.py`)

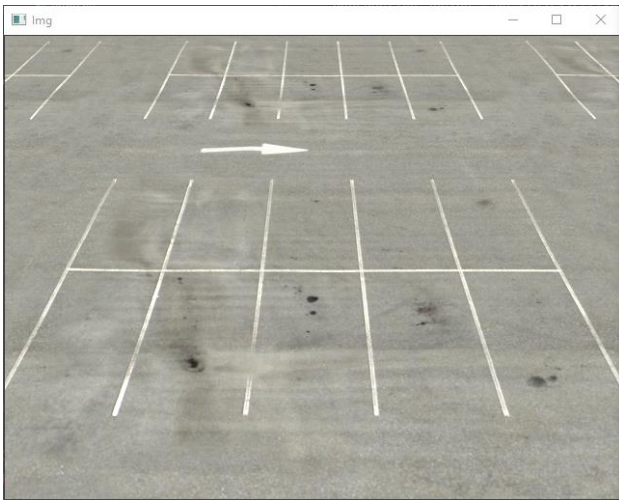


Figure 3 : Parking vide

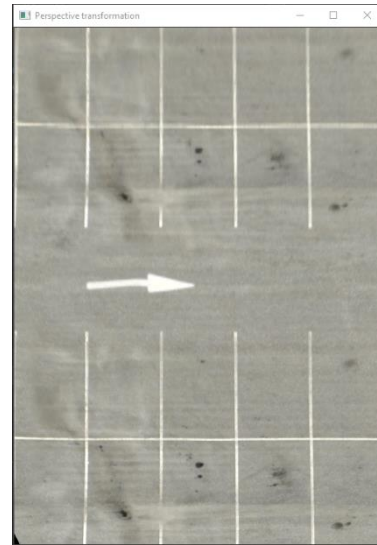


Figure 4 : Parking vide travaillé avec perspective

Détection de slot

Une autre procédure était de faire la détection de ligne. Comme a pu être fait pour les contours sur les formes, en partant de ce principe, on essaye de détecter uniquement les lignes des places de parkings. On utilise toujours la procédure niveau de gris, flou gaussien mais par après on fait de la détection de bord via Canny (cfr Bibliographie : `findContours`). Suite à celles-ci, on peut récupérer les lignes de cette dernière et les dessiner sur la nouvelle image via `HoughLinesP`. (cfr Bibliographie : `HoughLinesP`)

(cfr Annexe : `slot_detection.py`)

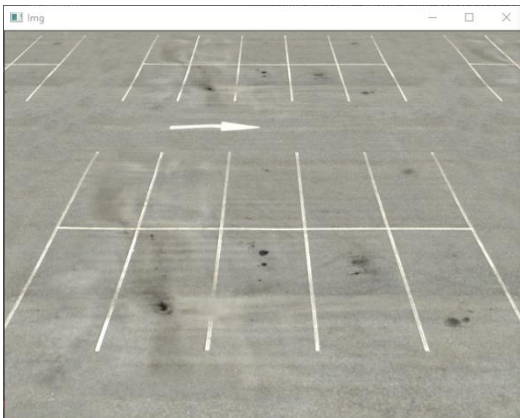


Figure 3 : Parking vide

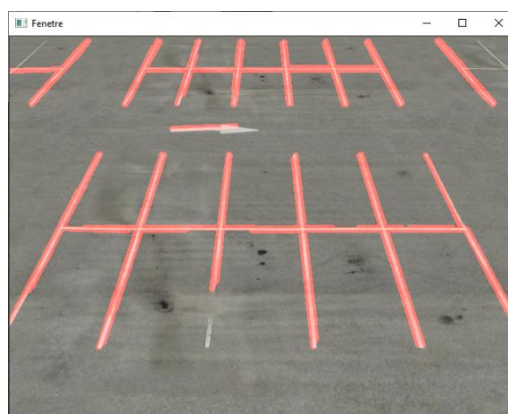


Figure 5 : Parking vide avec détection de lignes

Détection de coin

Comme j'ai pu le faire pour la détection de lignes pour les slots de parkings, j'ai voulu essayer de récupérer juste les points. Pour arriver aux résultats ci-dessous, au lieu d'utiliser le HoughLinesP, j'ai utilisé cornerHarris. La différence entre ces deux-là, le premier doit travailler sur une image où il y a eu une détection de bords alors que pour le cornerHarris, on récupère les données directement sur l'image originale. (cfr Bibliographie : CornerHarris)

(cfr Annexe : corner_detection.py)

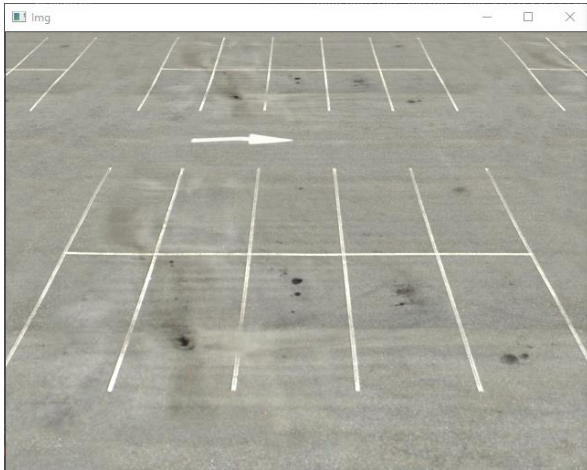


Figure 3 : Parking vide

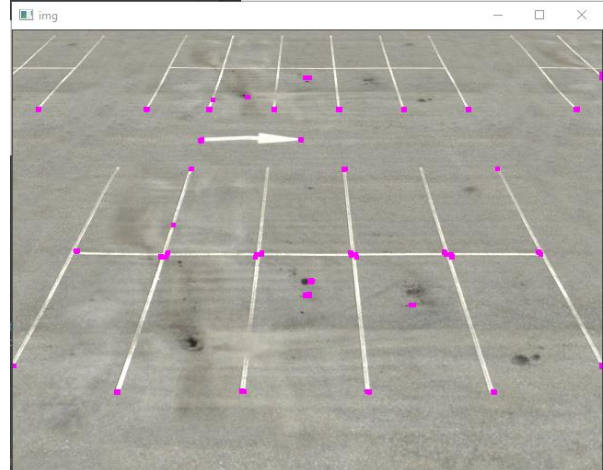


Figure 6 : Parking vide avec détection de coins

Petits tests

Pour terminer les tests, je me suis intéressé à une image où j'ai fait quelques fonctionnalités :

- Emplacement de parking : le fait de dessiner des points sur les quatre coins des slots de parking
- Centre des places : le fait de calculer la moitié entre deux paires de coins opposées et dessiner un cercle dans le centre des places.
- Compteur de pixel : le fait de faire une boucle qui récupère la valeur de chaque pixel et compte les noirs et les blancs. Le script envoie un message à la fin pour dire la majorité des pixels dans l'image en fonction du nombre le plus grand.

J'ai réutilisé la majorité des fonctions déjà utilisées. Les ajouts ont été faits avec des conditions et des prints.

(cfr Annexe : cas_isole.py)

Script principal

Pour pouvoir faire la détection de places, nous allons utiliser plusieurs fonctions vues lors des tests et les ajouter les unes aux autres afin de faire un script complet. Pour pouvoir travailler, nous allons faire la présentation sur un seul cas de parking. Si le temps me l'avait permis, je me serais intéressé à d'autres sortes de parkings. Pour montrer que le script fonctionne correctement, j'ai pu, grâce à Photoshop, modifier légèrement l'image afin d'avoir différents cas. (cfr Annexe : main.py)

Les quatre images sont :

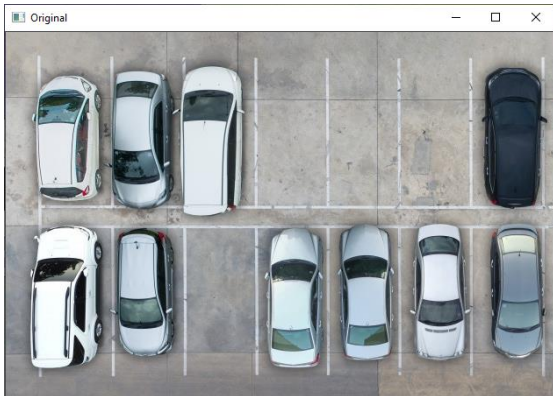


Figure 7 : Parking Cas 1



Figure 8 : Parking Cas 2



Figure 9 : Parking Cas 3



Figure 10 : Parking Cas 4

- Cas 1 : un parking avec 4 places libres
- Cas 2 : le même parking avec une voiture en plus donc 3 places libres
- Cas 3 : le même parking mais la voiture s'est mal garée et prend 2 places donc 2 places libres
- Cas 4 : identique au cas 1 mais une personne est en plein milieu d'une place et ne doit pas être considérée comme une voiture donc 4 places libres

Pour les explications, nous allons nous consacrer au cas1 et faudra partir sur deux optiques : mettre en valeur le parking et mettre en valeur les slots (lignes). Pour mettre en valeur le parking, on doit jouer sur le contraste et la luminosité et pour les lignes, on traite l'image originale. Dans les pages suivantes, nous allons voir le traitement fait dans les deux cas et le résultat obtenu.

Comme expliqué précédemment, nous allons modifier l'image originale et ajouter un contraste,

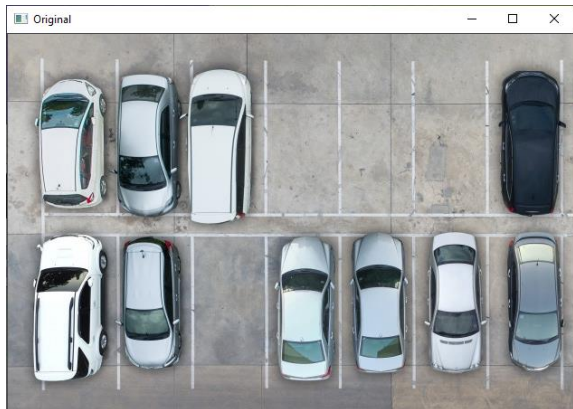


Figure 7 : Parking Cas 1

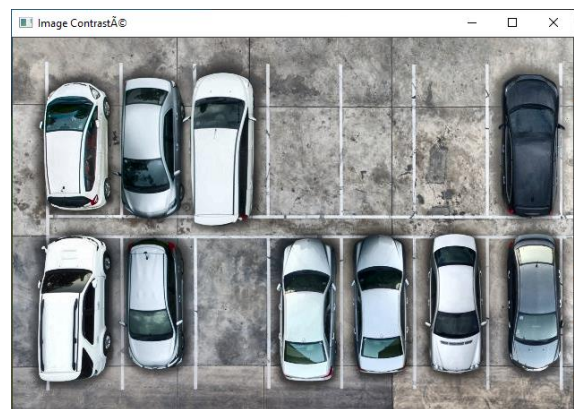


Figure 11 : Figure 7 plus contrastée

et par la suite augmenter la luminosité de l'image contrastée.

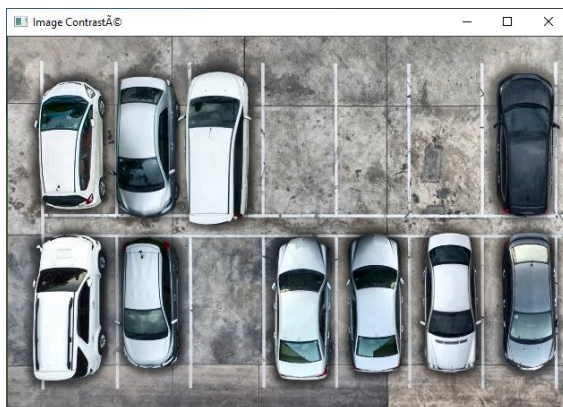


Figure 11 : Figure 7 plus contrastée

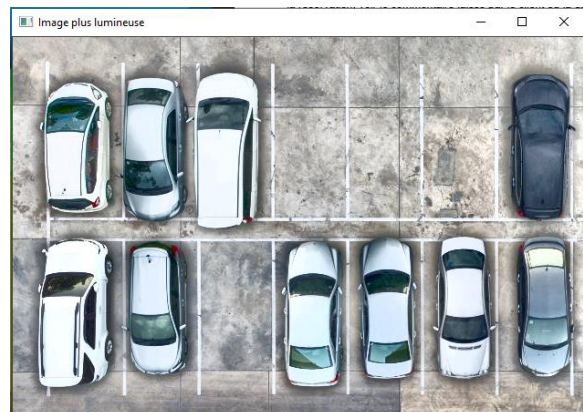


Figure 12 : Figure 11 plus lumineuse

Grâce à ce procédé, nous nous rapprochons de l'image originale mais en mettant le gris du parking en valeur. Dans les prochaines figures, les traitements faits via l'image originale se trouveront à gauche et ceux sur l'image contrastée-lumineuse à droite.

Ensuite, nous allons appliquer un filtre jaune et un filtre blanc sur les images original et contrastée-lumineuse, afin de faire ressortir les voitures de l'arrière-plan. Comme on peut le voir ci-dessous, on a une impression que les voitures sont séparées du parking et qu'elles possèdent toutes un contour noir.

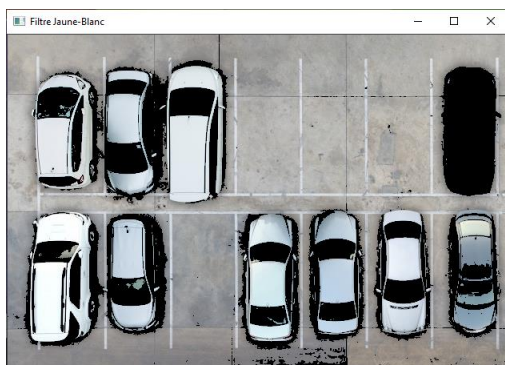


Figure 13 : Figure 7 avec Filtre Jaune-Blanc

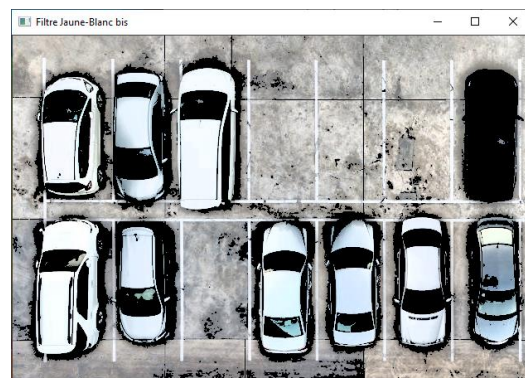


Figure 14 : Figure 12 avec Filtre Jaune-Blanc

Dans les tests, on a pu voir que pour gagner en performance, on devait passer l'image RGB en Niveau de Gris. De cette manière, on augmente les performances et réduit les trois couches en une couche qui représentera une luminosité du pixel.

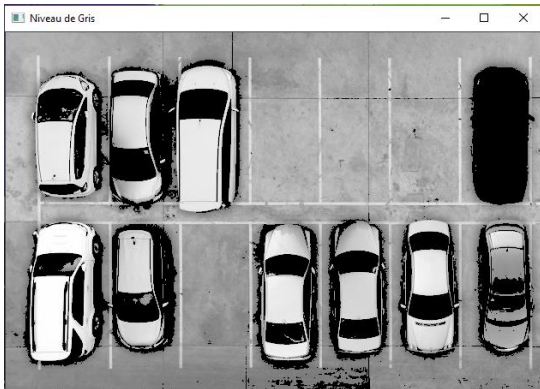


Figure 15 : Figure 13 en niveau de Gris

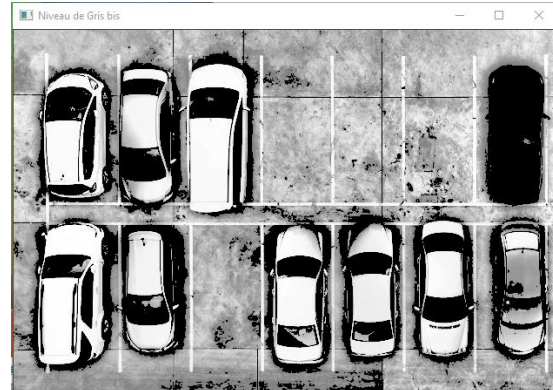


Figure 16 : Figure 14 en niveau de Gris

Nous prenons par la suite les contours des images via la fonction Canny. Comme on peut le voir, à gauche nous observons seulement les lignes du parking ainsi que les voitures alors qu'à droite, il y a également des contours dans les places vides. C'est ce qui nous permettra de faire une différence entre les deux et récupérer seulement les places libres.



Figure 17 : Figure 15 en détection de bord



Figure 18 : Figure 16 en détection de bord

Pour éviter de prendre en compte l'extérieur du parking, nous donnons une région d'intérêt ou Region of Interest (ROI) afin de n'avoir que dans ce cas les 14 places de parking possible via la fonction fillPoly. Difficilement visible, mais nous voyons, par exemple dans l'image de droite, que nous avons perdu des contours qui ont disparu en haut de l'image. (cfr Bibliographie : fillPoly)



Figure 19 : Région d'intérêt de la Figure 17



Figure 20 : Région d'intérêt de la Figure 18

Un des tests consistait à la détection de ligne et nous avons pu observer qu'avec HoughLinesP, on pouvait dessiner les lignes sur la photo originale. On crée alors une copie de l'image et on inclut d'abord les lignes des contours de l'image contrastée-lumineuse et ensuite de l'image originale, afin de retrouver les lignes vertes (image originale) au-dessus des lignes blanches (images contrastées-lumineuses).

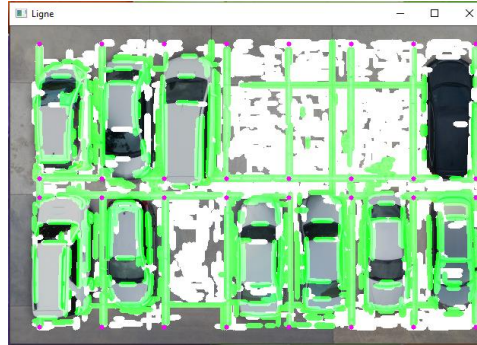


Figure 21 : Assemblage des valeurs

C'est avec cette image que nous allons créer le programme permettant de compter chaque élément. Les différentes étapes pour arriver aux résultats sont les suivantes.

- 1) On crée une boucle qui va aller de slot en slot.
- 2) Pour chaque slot, on fait une moyenne pour chaque couche (RGB) et on ajoute la valeur moyenne dans un array.
- 3) On compare les trois valeurs de l'array et on vérifie qu'elles soient toutes les trois au-dessus d'un numéro correspondant aux couleurs RGB des lignes ajoutées à l'image originale.
- 4) On rajoute ensuite un cercle au centre du slot grâce à une condition : si la condition est vraie, la place est libre et on met un cercle vert. Dans le cas contraire, si elle est fausse, la place est occupée et donc on met un cercle rouge.
- 5) On récupère les valeurs reçues (la date, le total de place, le nombre de places libres, l'adresse et le nom du parking).
- 6) Le script se connecte à la base de données et crée une table si elle n'est pas encore existante.
- 7) Il injecte enfin les données reçues dans la base de données.

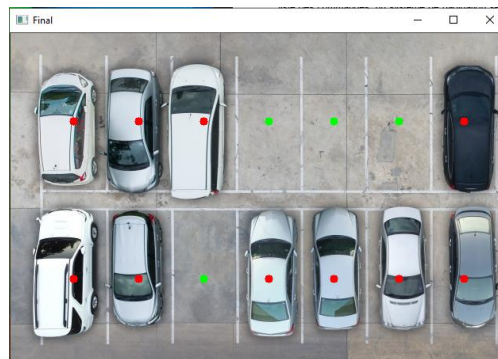


Figure 22 : Résultat final

Le script a fini la tâche qu'il lui était demandé et renvoi une image plus propre avec simplement des cercles en fonction de l'occupation du parking sur celle-ci.

Drone et composants

Quand il fallait trouver un TFE, j'ai décidé dès le départ de partir sur un drone aérien. C'est la première chose à laquelle j'ai pensé. Ces petits avions offrent énormément de possibilités que ce soit pour les forces armées, la sécurité ou encore un usage ludique et du fait qu'ils peuvent s'élever en altitude et être contrôlés à distance.

L'objectif du projet est d'apprendre à utiliser un nouvel outil, sa conception matérielle et logicielle, ses composants et d'en apprendre plus sur les objets connectés et la relation entre deux machines.

Le but de ce drone est de pouvoir visualiser des places de parkings, trouver une utilité à tout type de personnes et optimiser la circulation. Un autre but, plus personnel, est que je souhaitais avoir la possibilité d'utiliser cet outil. Je suis fasciné par cet objet et je suis sûr qu'on peut exploiter énormément de fonctionnalités propices à une vie meilleure. C'est une technologie très intéressante.

Premier choix

L'idée de départ était de partir sur le CrazyFlie de BitCraze.io. Ce drone est un quadcopter en pièces détachées et sa conception est simple : une carte électronique (PCB) rattachée à quatre moteurs afin de lui donner la possibilité de voler.

En ce qui concerne l'Hardware, le CrazyFlie a un faible coût et facile d'accès. Le CrazyFlie étant vendu en pièces détachées, c'est à l'acheteur d'assembler les différents composants. Il y a un kit de base vendu à 195 \$ et si un souci survient lors des tests ou à la réception de celui-ci, il y a un service client et un wiki afin de résoudre la défaillance. Dans le cas où les pièces ne peuvent pas être réparées, chacune d'elles peut être achetée séparément.

Le CrazyFlie est vendu en tant que plateforme de développement open-source. Il est configurable en C et Python.

L'avantage, si j'optais sur cette idée, était de mettre en avant mes atouts en électronique afin d'assembler, configurer et tester. Par la suite, utiliser ce produit créé pour le projet en lui donnant la possibilité d'analyser les parkings.

Les inconvénients, en revanche, m'ont fait opter pour un nouveau choix :

Premièrement, le circuit n'est pas protégé et donc pas étanche. Il fallait réfléchir et modéliser un boîtier pour le quadricoptère.

Deuxièmement, n'étant pas doté de caméra, il fallait utiliser une solution avec des capteurs infrarouges. Soit deux capteurs qui se relient et donc travailler sur chaque parking afin d'ajouter des capteurs à chaque place. Le temps pour implémenter le projet sur plusieurs endroits serait intolérable. Soit un seul capteur infrarouge sur le drone qui mesure la distance du rayon. Il fallait donc initialiser la hauteur de vol du drone et un coup de vent tous les calculs sont mauvais et plusieurs erreurs seraient survenues.

Pour finir, le temps et la quantité de travail donné pour ce drone ne seront pas proportionnels à la valeur qu'il apportera. Il fallait trouver autre chose.

Choix final

Le drone utilisé pour les tests est l'AR Drone Parrot 2.0. C'est un drone qui m'a été prêté par l'UCLouvain. Le fait d'avoir eu ce drone m'a permis de me concentrer sur le script PHP.

Cet aéronef est accessible à tout public, robuste et facilement contrôlable. Il possède plusieurs atouts qui seront utiles au projet : deux caméras, une stabilité haute altitude, le mode Absolute Contrôle de Parrot et une résistance à l'eau avec une structure renforcée ainsi qu'une application gratuite et complète. Une explication de chacun de ces composants ci-dessous.

Les Caméras

Le drone possède deux caméras. Une caméra HD de 720p frontale et filmant l'avant ainsi qu'une caméra ventrale de 320p dirigée vers le sol. Il y a un retour vidéo en direct et avec une latence faible, presque invisible à l'œil nu. Il y a une possibilité d'enregistrer les images, que ce soit vidéos ou photos. Un port USB est disponible pour stocker les données mais par défaut, les données sont enregistrées sur l'ordinateur ou le téléphone. Des images du parking de l'UCLouvain en 320p se retrouvent dans les annexes.

(cfr Annexe : Différences Caméra) (cfr Dossier 'images' : parking1-4.jpg)



Figure 23 : Caméra ventrale de 320p



Figure 24 : Caméra frontale de 720p

Cependant, un problème survient : la caméra avec la plus haute résolution n'est pas dirigée vers le sol et celle qui l'est n'a pas une assez bonne résolution, donc les traitements d'images ne sont pas optimal, voir ne fonctionne pas. La solution qui est donc proposée pour le projet est de diriger la caméra frontale vers le bas. De cette manière, nous utiliserons la caméra la plus optimale et pourrons par la même occasion traiter les images reçues. Dans la figure 25, nous trouverons à gauche le drone modifié et à droite un drone normal.



Figure 25 : Modification de la position de la caméra

La Stabilité

Le drone possède une stabilisation automatique à haute altitude. Cette fonctionnalité permet de prendre des photos sans être flou. Ce paramètre est possible grâce à un capteur de pression intégré. Lors des déplacements, ce capteur permettra de mesurer la vitesse verticale et d'améliorer la navigation. Il fera les calculs nécessaires afin de remettre le drone dans une bonne position malgré les coups de vent. Il facilitera également les acrobaties du drone, mais non-essentiels à ce projet.

La Batterie

Le drone peut contenir une batterie. L'UCLouvain m'a mis à disposition plusieurs batteries de capacités et poids différents. J'ai, après quelques essais, opté pour la batterie de 1500maH : elle est plus légère que celle de 2500maH et permet de faire tenir le drone légèrement plus longtemps dans le ciel. Cette batterie, lorsqu'elle est chargée à 100 %, fait voler le drone pendant 9 minutes 53 secondes 68 centièmes. Certains calculs ont été faits pour connaître la taille maximale du parking que le drone peut survoler en prenant en considération la capacité de la batterie. (cfr Parking Maximal)



Figure 26 : Batterie

Le Mode Absolute Control

Ce mode de pilotage, développé par Parrot et exclusif à ce drone, permet de se positionner dans n'importe quelle direction face à l'AR Drone et que ce dernier aille dans la position indiquer. Pour faire simple, le drone s'adapte à l'endroit où est positionné l'utilisateur de celui-ci. Si l'utilisateur avance le joystick vers l'avant, le drone va s'éloigner de lui, qu'il soit en avant ou en arrière.

La Résistance

Ce drone est résistant à l'eau grâce à une fine pellicule étanche qui recouvre les parties électroniques. Il peut donc voler au-dessus de la neige ou d'une faible pluie. La caméra est également protégée par une petite coque et le drone possède une carène renforcée afin de le protéger en vol. Dans la figure 27, nous retrouvons la carène pour l'extérieur à gauche, pour l'intérieur à droite et le drone au centre.



Figure 27 : Drone et Carènes

L'Application

Le drone possède une application gratuite : AR.FreeFlight. Cette application possède une interface ergonomique et une facilité au niveau des réglages permettant de tout régler lors de l'utilisation du drone. On peut y retrouver les contrôles de vols, la galerie photo, les paramètres permettent de changer les préférences de vol...



Contact avec le client

Pour ce qui est du contact avec le client, nous nous sommes vu peu de fois mais on a réussi à mettre sur papier les critères minimum et pour définir ses besoins. Durant la réalisation du TFE, nous avons accès à un Trello afin de voir les avancées du projet.

Comme c'est un ami qui jouait le rôle du client, on a opté pour un choix. Il me donne tous ses souhaits et on discute du projet pour voir ce qui est possible ou pas. Sa demande consistait à trouver un moyen afin de se garer rapidement. Connaître le nombre de places libres dans un parking et pouvoir y avoir accès. Le choix d'un drone est une option qui a été proposée et j'ai accepté de le faire dans cette optique.

Nous avons souvent été d'accord dans les choix de chacun et lorsque j'hésitais sur certaines solutions, il me conseillait sur celle qui était la plus intéressante.

Méthode de travail

Ma méthode consistait à donner de l'importance aux résultats et pas au code. J'implémentais les fonctionnalités et ne me souciais pas de si le code était propre ou non. Ayant fait plusieurs tests avant de faire le code principal, je reprenais des bouts de code déjà fait et les implémentais dans le script de base. Dès lors où le résultat était à ma convenance, je reparcourais tout le code et prenais du temps à effectuer les corrections non pas pour moi, mais pour que le code soit compréhensible par un autre développeur.

Recommandations

Problèmes rencontrés

Le premier problème que j'ai pu rencontrer, résidait dans les captures d'images faites avec le drone : n'ayant au début pas pris en compte la résolution, je pensais que la caméra ventrale suffirait et que les tests seraient corrects avec les images prises. Quand j'ai commencé à essayer de travailler avec celles-ci, j'ai compris qu'il fallait que je trouve une solution. C'est pour cette raison que j'ai décidé d'inverser l'axe de la caméra frontale.

Un second problème venait du formulaire : étant donné qu'il faut un service mail, je ne pouvais pas faire les tests en local. J'ai donc hébergé le site web et travaillé directement via l'hébergement. C'était donc des tests qui prenaient un peu plus de temps mais qui fonctionnent correctement maintenant. Il y a une méthode pour ajouter un service mail à wamp mais n'est plus d'actualité et je n'ai pas pris le temps d'en trouver une autre. (cfr Bibliographie : Wamp Mail)

Un problème qui a pris un peu de temps à résoudre : dans la base de données de l'hébergement, le site pouvait y accéder mais pas le script Python. Au début, j'ai cru à un problème dans la base de données qui était local. J'ai donc essayé plusieurs SGBD gratuites et le problème était inversé. Le script envoyait les données mais le site ne les récupérait pas. Après plusieurs essais, j'ai compris qu'il fallait que j'autorise l'adresse IP de ma machine à accéder à la base de données de l'hébergeur.

Un dernier problème, mais pas majeur : la responsivité qui ne se faisait pas sur certains éléments du site (images, ligne de mise en page...). J'ai donc pris l'initiative de cacher les éléments sur certains écrans s'ils ne sont pas primordiaux afin d'avoir un site le plus responsive possible.

Améliorations futures

Plusieurs pistes d'améliorations peuvent être travaillées :

Premièrement, un outil nommé ROS permet de développer des logiciels pour la robotique. Avec un peu de bricolage et de recherches, ce langage peut permettre de faire un logiciel qui donne des instructions et le robot les exécute. ROS est disponible pour les AR Drone Parrot. L'idée serait donc que le drone soit relié à un wifi du parking, l'ordinateur également et ce dernier envoie des instructions avec les coordonnées GPS afin d'envoyer le drone faire des captures à des endroits précis et transfère les images à l'ordinateur qui les récupère et les traite. Cette solution permettra de réduire l'intervention humaine. (cfr Bibliographie : ROS)

La seconde solution est d'améliorer l'onglet Parking sur le site web afin de pouvoir cliquer sur un tableau et d'accéder au parking choisi. Pour l'instant, nous utilisons un seul parking pour l'exemple mais comme le but est de s'étendre, il faut déjà réfléchir à la solution.

Une autre amélioration est de travailler en deep learning ou voire même en intelligence artificielle. Le python est un langage vaste permettant d'obtenir énormément de résultats. Une des alternatives est le CNN. Le Réseau Neuronal Convolutif ou Convolutional Neural Network (CNN) réagit directement sur les images qui lui sont transmises. En outre, pour que le CNN fonctionne correctement, il faut l'entraîner afin qu'il sache différencier une place libre d'une place avec un moyen de transport. C'est une solution de deep learning.

La dernière amélioration concerne le site web. Lors de tests, j'ai pu remarquer que le site a reçu une note ecoIndex de C (note sur l'empreinte environnementale du site). Ce n'est pas une mauvaise note mais, ayant un projet qui consiste à réduire le taux de CO2, cela donne une mauvaise image à 2Par. (cfr Annexe : EcoIndex)

Parking Maximal

J'ai pu prendre plusieurs informations afin de faire mon calcul (ces calculs sont manuels, il y a donc un taux d'erreur à prendre en compte) :

- Le drone peut voler pendant 9 minutes 53 secondes 68 centièmes lorsque la batterie est pleine.
- La batterie se charge complètement en 1 h 30
- Il lui faut 0,5 seconde pour se stabiliser
- Il lui faut 3,7 secondes pour se déplacer dans une nouvelle zone de capture (places et éléments différents de la capture précédente).
- Il lui faut 26,69 secondes pour se lever et se déplacer jusqu'à la première zone de capture en s'orientant correctement.
- La hauteur maximale du drone est de 10m.
- Il lui faut 13,96 secondes pour atterrir en étant au-dessus de la zone d'atterrissage.
- Le drone prend une image de 12 m sur 10 m (taille approximative)

Avec ces informations, on peut calculer approximativement la taille maximale du parking.

Les calculs sont fait selon le parking idéal, à savoir une longue ligne droite.

Handwritten calculations for the maximum parking size:

- 9'53"68 = 100% batterie
- 533"68
- 0,5" = stabilisation
- 3,7" = déplacement $\Rightarrow A/R = 7,9"$
- prix de vue + A/R = 7,9"
- 1^{re} capture = 26"68
- hauteur max = 10m
- 10m
- 7,9 x 70 = 553 \Rightarrow
- 70 prises possible + 1^{re} capture = 71 images
- 12 x 71 = 852
- La taille maximale est de 8520 m²

Additional calculations on the right side of the page:

- Temps total = 533"68
- 1^{re} capture = 26"68
- Temps restants (TR) = 507"00
- atterissage = 13,96"
- 507,00"
- 7,9 x = 553,04
- x = 553,04
- 7,9
- x = 70,005
- 553,04 - 553 = 0,04
- 0,04 secondes restantes
- 533"68 - 100%
- 0,04 - 0,006%

Grâce aux calculs, nous pouvons conclure plusieurs choses :

- La taille maximum du parking idéal est de 8520 m²
- Lorsque le drone aura terminé tous ces clichés, il restera 0,04 seconde d'autonomie à savoir 0,006 % de batterie
- Le drone pourra faire 71 images.
- Si nous souhaitons faire voler le drone toutes les 30 minutes, il faudra trois batteries.

Confidentialité

De nos jours, quasiment tous les drones sont équipés d'une caméra, de ce fait, des photos et des vidéos sont capturées sans que l'on ne s'en aperçoive.

Les utilisations sont très variées, par exemple, ils peuvent être employés comme moyens de reconnaissance faciale, de surveillance, de livraison, de cartographie ou même pour jouer tout simplement. Mais le problème reste les utilisateurs, lorsqu'ils sont utilisés par des civils, ils sont impossibles à contrôler.

C'est pour cela que dans le projet, les images ne seront pas enregistrées. Le drone envoie les captures au script, celui-ci fait son traitement, ajoute les informations utiles à la base de données et supprime les photos reçues. Le drone ne capture également aucune plaque ou ne fait pas de la reconnaissance faciale.

Cybersécurité

Les drones sont des machines qui peuvent se connecter à Internet. De ce fait, ils sont vulnérables aux virus. Il faut aussi savoir qu'ils ne disposent pas de sécurité intégrée poussée ce qui les rend encore plus vulnérables que nos PC ou nos smartphones.

Avec l'évolution de ces avions radiocommandés et des domaines d'utilisations, cela présentera bien évidemment beaucoup plus d'intérêt financier pour les hackers et donc beaucoup plus de risques pour les utilisateurs et on pourrait observer une augmentation du nombre de malwares.

Cependant et heureusement pour nous, les données utilisées dans notre cas ne sont pas des données confidentielles. Il n'y a pas de transaction ni de mot de passe enregistré et donc aucun intérêt à infecter un drone. La principale préoccupation des pirates reste l'argent.

Sécurité physique

Bien que la cybersécurité et la confidentialité sont importantes, la sécurité physique l'est tout autant.

Certains de ces objets volants peuvent peser jusqu'à 150 kilos, sans compter qu'ils peuvent voler à 150 mètres d'altitude (cfr Bibliographie : Législation belge), un accident peut vite arriver, il suffit d'une erreur de calcul ou même d'une mauvaise manipulation et les conséquences pourraient être très graves.

Heureusement, sur certains drones, la sécurité à ce niveau-là est aussi en développement et de nouvelles fonctionnalités sont incorporées, comme par exemple l'Obstacle Avoidance ou l'Obstacle Sensing qui permet de ralentir, d'arrêter et de bloquer l'avancer d'un drone lorsqu'il s'approche d'un obstacle, et cela grâce à des capteurs.

Contrainte législative

Comme toute activité comprenant des menaces, des contraintes doivent être appliquées. Dans notre cas, ce projet utilise le drone comme usage récréatif. Les mesures qui sont d'application sont :

- Poids du drone : Max 1 kg
- Hauteur maximum de vol : 10 m
- Au-dessus d'un terrain privé
- A plus de 3 kilomètres des aéroports et aérodromes

Le projet respecte donc toutes les conditions. Cependant, il faudra faire attention à deux choses. Premièrement, j'ai pu utiliser un parking privé (UCLouvain) car j'en ai eu l'autorisation et il faudra demander l'autorisation dans chaque parking (de préférence par écrit). Deuxièmement, les parkings sont loin des aéroports mais cela signifie également qu'on ne pourra pas implémenter le projet dans leurs parkings.

(cfr Annexe : Législation belge)

Assurances

Tous les appareils ne doivent pas obligatoirement être couverts par une assurance drone. Selon la catégorie de votre appareil, celui-ci sera soumis à une législation bien précise. D'ailleurs, une licence de télépilote, une formation adéquate et un examen pratique pourra vous être imposé avant d'utiliser votre drone.

Vous pouvez obtenir cette assurance via ce lien : <https://www.assurance-drone.be/>

Licences et Formation

Comme évoquée précédemment, une attestation ou licence de télépilote valable est obligatoire pour les drones de Classe 1 et 2. Dans notre cas, le drone peut être utilisé en usage récréatif. Des formations sont disponibles dans des clubs et associations d'aéromodélisme pour tous les âges. Cependant le télépilote doit respecter les mesures générales de sécurité.

Si l'on souhaite survoler avec un drone de classe 2, une attestation sera demandée. Pour l'obtenir, trois conditions sont requises : être âgé de 16 ans, avoir suivi une formation théorique et réussir un examen pratique. L'examen pratique doit être validé par un examinateur RPAS. (cfr Bibliographie : - Attestation de télépilote)

Par contre si notre choix se tourne vers les drones de Classe 1, nous devons passer une licence. L'avantage en ayant seulement la licence est de pouvoir survoler avec les Classe 1 et 2. Pour l'obtenir, cinq conditions sont requises : être âgé de 18 ans, suivre une formation théorique et suivre une formation pratique chez un instructeur reconnu, réussir l'examen théorique à la DGTA et réussir un test pratique devant un examinateur RPAS reconnu et désigné par la DGTA. La DGTA est la Direction générale Transport aérien. (cfr Annexe : Législation belge)

Aspect RSE

Durant le cursus à l'EPHEC, nous avons pu avoir un cours sur la déontologie et dans celui-ci, vu les différents aspects RSE d'une entreprise et les Objectifs de développement durable.

Mon projet nous permet de cibler et de toucher un maximum de personne étant donné que le but de ce projet est de partager des données et qu'elles soient accessibles à tout le monde. Je réponds à un problème actuel et qui ne cesse de croître de jour en jour. En effet, dans le monde d'aujourd'hui la pollution est de plus en plus présente et les campagnes sensibilisant sur celle-ci touchent les gens et les inquiètent.

Le projet répond à trois ODD :

- 11e ODD « Ville et communauté durable »
- 13e ODD « Mesures Relatives à la Lutte contre le Réchauffement Climatique »
- 15e ODD « Vie terrestre »

11^e ODD

Ce projet rentre dans cet ODD. Les villes et entreprise qui implémenterait le projet dans leurs parkings permettraient au public de s'informer sur les emplacements libres. Si un parking est rempli, les utilisateurs ne se déplaceront pas vers celui-ci et iront à un endroit où ils ont plus de chance de trouver une place. Cela réduira en conséquence leur consommation de CO₂. Les conducteurs deviendront donc plus responsable et produiront moins de CO₂ grâce à une simple action de l'habitant.

Les villes quant à elle en essayant de placer un maximum le projet dans leurs parkings, pourront réduire et fluidifier la circulation dans les heures de pointe. Elles sensibiliseront les conducteurs à prendre des transports en commun ou des moyen de locomotion vert tel que le vélo, la trottinette ou même la marche.

Toutes ces choses permettront aux villes de devenir des communautés durables grâce à 2Par afin de changer les habitudes des utilisateurs.

13^e ODD

Le projet vise également à intégrer cet ODD. 2Par espère pouvoir agir afin de ne pas arriver à un cas excessif de CO₂ en prenant une mesure de sécurité. Pourquoi changer dès maintenant les habitudes ? L'air contient en moyenne 0.04 % aujourd'hui. Le corps humain ne peut pas s'exposer plus de 15 minutes à 3 % et cette valeur ne doit jamais être dépassée. Nous ressentons des effets à partir de 2 % avec une amplitude respiratoire qui augmente. À 4 %, la fréquence respiratoire commence à s'accélérer. À 10 %, nous ressentons des symptômes plus concrets tels que des troubles visuels suivis d'une perte de connaissance brutale à 15 % et d'un arrêt respiratoire entraînant le décès à 25 %. Je souhaite éviter ces problèmes et prévenir des dangers qui peuvent survenir. C'est pour cela que 2Par est une des mesures possibles mais ne doit pas être la seule à prendre.

15^e ODD

Au-delà d'aider les villes à réduire leur taux de CO₂ dans l'air pour ses habitants, améliorer la qualité de l'air permet à tout être vivant d'avoir des meilleures conditions de vie. Si les villes se voient ainsi réduire leur production de CO₂ et avoir un air meilleur cela impactera aussi tous les écosystèmes au sein même de la ville et ceux se trouvant dans les alentours.

Le projet pourra alors à long terme impacter tout l'écosystème se trouvant près de milieux aménagés par les villes, pour leurs citoyens en premier lieux mais aussi la faune et la flore se trouvant tout autour. On pourra alors voir une augmentation d'espèces animales et végétales dans ces lieux ce qui permettra à tous les animaux de trouver plus d'espèces à chasser. La vie terrestre ne sera pas directement impactée par le projet mais le sera indirectement.

Conclusion

En conclusion, ce projet m'a permis d'en apprendre beaucoup sur le monde du drone aérien et des différentes restrictions lié à celui-ci. J'ai pu également trouver une passion pour les objets connectés, repousser mes limites pour fournir une application web la plus complète possible avec mes connaissances et celle que j'ai pu acquérir. Malgré le manque de temps et la situation actuelle, je suis satisfait du TFE que je présente.

L'avantage est que ce projet me plaisait et que j'avais décidé de l'idée et de comment obtenir un résultat. Je n'ai donc pas eu l'impression de travailler tout au long de ce projet et j'en apprenais davantage chaque jour, que ce soit dans le développement du site ou dans le script. J'ai pu utiliser les outils que je préférais ce qui m'a poussé à en connaître d'avantage et à ressortir mon envie de continuer sur le projet. J'ai compris également grâce à ce projet que je pouvais trouver des solutions au développement et travailler en autonomie. Ayant l'habitude de me reposer sur les idées des autres et de suivre le mouvement, c'était devenu un challenge qui m'a redonné confiance en moi.

Ce projet ne restera pas à ce stade et je continuerais à travailler dessus quand le temps me le permettra.

2Par, la solution de partout

Bibliographie

- AerosSpaceLab : <https://www.aerospacelab.be/>
- Attestation de télépilote:
https://mobilit.belgium.be/fr/transport_aerien/licences/attestation_ou_licence_de_telepilote/attestation_de_telepilote
- Canny: https://docs.opencv.org/3.4/da/d5c/tutorial_canny_detector.html
- CNN: <https://www.tensorflow.org/tutorials/images/cnn>
- CornerHarris : https://docs.opencv.org/master/dc/d0d/tutorial_py_features_harris.html
- cvtColor :
https://docs.opencv.org/3.4/d8/d01/group_imgproc_color_conversions.html
- fillPoly : https://docs.opencv.org/master/d6/d6e/group_imgproc_draw.html
- GaussianBlur : https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html
- Geometric Image Transformations :
https://docs.opencv.org/master/da/d54/group_imgproc_transform.html
- HoughLinesP : https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html
- Législation belge :
https://mobilit.belgium.be/fr/transport_aerien/drones/ancienne_legislation/jai_un_drone_et_maintenant#:~:text=Un%20drone%20est%20un%20a%C3%A9ronef,d%27un%20poste%20de%20t%C3%A9pilote.&text=Outre%20la%20s%C3%A9curit%C3%A9%2C%20l%27arr%C3%AAt%C3%A9,la%20vie%20priv%C3%A9e%20du%20citoyen
- Licence télépilote d'un RPA ou drone :
https://mobilit.belgium.be/fr/transport_aerien/drones/attestation_ou_licence_de_telepilote/licence_de_telepilote_dun_rpa_ou_drone
- Objectifs de développement durable :
<https://www.un.org/sustainabledevelopment/fr/objectifs-de-developpement-durable/>
- Péruwelz : https://www.rtbf.be/info/regions/hainaut/detail_peruwelz-teste-une-camera-intelligente-pour-le-parking-de-la-grand-place?id=10753490
- PHP : <https://www.php.net/>
- ROS : https://fr.wikipedia.org/wiki/Robot_Operating_System
- Threshold: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html
- Verlan : <https://dictionnaire.lerobert.com/definition/verlan>
- Wamp Mail : <https://grafikart.fr/blog/mail-local-wamp>