

TraffiCounter

Rapport

Promotion 2025

Groupe 103

Chef de projet : Ryan SEBBANE

Matthieu EVUORT - Marty EVUORT

Guillaume CLAVIER - Benjamin HOANG

Sommaire

I.	Introduction	2
II.	Réalisation du projet.	2
II.	Bilan rétrospective du projet	4
III.	Impact RSE	6
IV.	Planning prévisionnel	6
V.	Déroulement des tâches	7
VI.	Conclusion de la réalisation du projet.	8
	Sources	9

I. Introduction

Chaque année 50 000 accidents de la route ont lieu. A l'heure où vous lisez ces lignes, ces derniers sont peut-être même en train de se produire. Il est donc vital de remodeler les routes afin de les stopper. Que ce soit pour diminuer le nombre d'accident, ou pour créer/supprimer de nouvelles routes, TrafficCounter est une application née de ces besoins. Ce dernier est un logiciel permettant de recenser le nombre de véhicules. A l'aide de cette simple action, il sera possible d'analyser le trafic routier et donc d'agir en conséquence. Pour cela, nous réaliserons un programme en python et nous utiliserons OpenCV. Tout en codant, nous avons en tête tous les objectifs que devait réaliser notre logiciel, tels que :

- Concevoir un programme détectant des objets avec une caméra (et une vidéo)
- Concevoir un programme avec le deep learning différenciant les véhicules en circulation (compter une voiture et non un camion par exemple)
- Concevoir un programme incrémentant un compteur même lorsqu'il y a plusieurs voitures présentes dans notre vidéo
- Concevoir un programme triant et stockant les données obtenues afin de les analyser (ultérieurement par exemple)

Nous avons également pris en compte des objectifs secondaires tels que :

- Concevoir un programme permettant d'obtenir la vitesse de chaque véhicule
- Concevoir un programme permettant de recenser le nombre de véhicule par modèle et par marque

II. Réalisation du projet.

Tout d'abord, pour réaliser notre projet nous avons, premièrement, cherché de la documentation sur la bibliothèque open-cv afin d'en connaître en profondeur les fonctionnalités. Puis, nous avons réalisé un algorithme python permettant de reconnaître les voitures en temps réel ainsi que pour une vidéo

choisie. En effet, l'utilisateur a le choix entre choisir une vidéo qu'il a filmée ou utiliser sa caméra en direct.

```
turtle.clear()
write("Voulez-vous utiliser une caméra ou ouvrir une vidéo", font=("Comic sans ms", 20, ""))
choice = textinput("Choix", "Choisir votre option\n- caméra\n- vidéo\n")
while(choice != "caméra" and choice != "camera" and choice != "vidéo" and choice != "video"):
    turtle.clear()
    write("Voulez-vous utiliser une caméra ou ouvrir une vidéo", font=("Comic sans ms", 20, ""))
    choice = textinput("Choix", "Choisir votre option\n- caméra\n- vidéo\n")
```

Puis, nous avons créé une image binaire à partir de la caméra ou de la vidéo pour que l'ordinateur puisse l'analyser. Pour cela, nous avons transformé l'image en gris, puis nous avons utilisé un soustracteur pour isoler les voitures qui passent.

```
grey = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(grey, (3, 3), 5)
img_sub = subtract.apply(blur)
dilat = cv2.dilate(img_sub, np.ones((5, 5)))
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
dilated = cv2.morphologyEx(dilat, cv2.MORPH_CLOSE, kernel)
dilated = cv2.morphologyEx(dilated, cv2.MORPH_CLOSE, kernel)
```

Après, cela, nous avons créé les contours de chaque objet afin de rendre la vidéo analysable par l'ordinateur.

```
contour, h = cv2.findContours(dilated, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

À partir de ces contours, nous avons détecté les rectangles les plus petits englobant ces contours en vue de dessiner les rectangles verts et les points rouges afin d'afficher les véhicules à l'écran.

```
for (i, c) in enumerate(contour):
    (x, y, w, h) = cv2.boundingRect(c)
    valid_contour = (w >= min_length) and (h >= min_height)
    if not valid_contour:
        continue

    cv2.rectangle(frame1, (x, y), (x + w, y + h), (0, 255, 0), 2)
    center = centering(x, y, w, h)
    detect.append(center)
    cv2.circle(frame1, center, 4, (0, 0, 255), -1)
```

À la fin, nous avons incrémenter le compteur dès que la voiture passait la ligne.

```
for (x, y) in detect:
    if (pos_count + offset) > y > (pos_count - offset):
        cars += 1
        if (choice == "camera" or choice == "caméra"):
            cv2.line(frame1, (4, pos_count), (636, pos_count), (0, 127, 255), 3)
        else:
            cv2.line(frame1, (5, pos_count), (1275, pos_count), (0, 127, 255), 3)
        detect.remove((x, y))
        turtle.clear()
        write("Car is detected : " + str(cars), font=("Comic sans ms", 20, ""))
```

Les principales difficultés rencontrées étaient de rendre l'image analysable par l'ordinateur car il fallait assimiler de nombreuses fonctions d'opencv afin de parvenir à ce résultat. Une autre difficulté était de pouvoir analyser le trafic routier en temps réel.

Les améliorations apportées sont l'ajout du texte à l'écran ainsi que la création d'un menu interactif avec l'utilisateur le permettant de switcher entre analyse en temps réel et analyse différée.

II. Bilan rétrospective du projet

Au début, nous avons eu du mal à travailler car même si nous avons tous les outils adéquats pour être bien organisé, c'était notre 2^e gros projet en groupe. De plus, nous n'étions pas habitués à travailler comme une équipe de développeur. Ainsi, la 1^{re} grande difficulté était d'estimer la durée qu'allait nous prendre le projet. En effet, n'ayant jamais été confronté à ce genre de projet, il a été difficile d'évaluer les compétences nécessaires à la bonne réalisation de ce dernier. En revanche, pour pallier ce problème, nous nous documentons beaucoup à l'aide d'internet (recherche sur les différents d'outils pour bien prendre conscience des enjeux de ce genre de projet). De plus, nous organisons des réunions fréquemment pour échanger nos informations. Ces dernières avaient également pour but de réfléchir aux fonctionnalités de notre application. A l'instar du Daily scrum où tous les jours nous avons indiqué les activités que nous avons

effectuées et ce qu'on pouvait améliorer. Ensuite toutes les semaines et tous les mois nous avons effectué une rétrospection des tâches effectuées et on ajoutait les nouvelles tâches à effectuer.

Nous avons eu énormément de difficulté à obtenir le raspberry Pi adéquat pour la réalisation de notre projet. Par conséquent, nous avons seulement utilisé la caméra de notre ordinateur ou bien celle de notre téléphone relié à l'ordinateur. De plus nous avons eu des difficultés à utiliser Matlab pour le projet alors on s'est tourné sur python qui est un outil familier pour nous tous. Le deep Learning et le machine Learning étaient énormément plus accessible sur python que sur Matlab alors nous avons totalement délaissé Matlab pour python. Comprendre les concepts du deep Learning et du machine Learning étaient difficile pour nous tous et nous avons passé la majorité du temps à comprendre ses enjeux et ses principes. De plus, exprimer nos idées dans le code sur un sujet qui était nouveau était laborieux mais au fur à et mesure que nous nous plongeons et que nous nous documentions, nous avons eu plus d'aisance à concrétiser nos idées. Même si nous avons réussi à coder notre programme, nous avons rencontrées des difficultés lors de l'ajout d'une caméra filmant en temps réelle une route. Nous avons donc opté pour une simple analyse vidéo. Cependant, après un travail acharné, nous avons choisi et réussi à appliquer ces deux méthodes à notre application (choix effectué par l'utilisateur). Après s'être mieux documenté et après plusieurs essais nous avons donc réussi à implémenter le traitement en temps réel de la détection des mouvements dans notre code. On a aussi remarqué que lors du passage des camions, le compteur comptait plusieurs voitures au lieu d'une. Par conséquent, nous nous sommes renseignés à propos des manières permettant de régler ce problème. Nous avons par exemple envisagé de modifier la vitesse du clip vidéo ou reconnaître la taille du rectangle (sélection faite sur les différents véhicules par l'algorithme) pour différencier un camion d'une voiture (sélection plus grande et plus large dans le cas d'un camion). Cependant, par manque de temps, nous n'avons pas pu appliquer nos idées.

Nous avons fait de nombreuses et régulières mises au point des objectifs que nous devions atteindre. De plus, nous sommes très contents du résultat final mais nous pouvons pousser plus loin notre projet. En effet, nous aurions pu calculer la vitesse de chaque voiture détectée par la caméra, régler le problème des camions comptés plusieurs fois, différencier les différents types de voitures, compter les véhicules

en fonction des modèles/marques. Dans l'ensemble, hormis quelques objectifs, la plupart ont quand même été réalisés.

III. Impact RSE

Notre projet a un impact RSE environnemental très faible. En effet, notre application a été réalisée en intégralité en Python. De plus, la seule source possible de consommation serait l'usage d'un ordinateur ainsi que d'une caméra ou d'un téléphone. En revanche, ne nous pouvons pas réellement agir face à ces sources. En effet, nous ne sommes pas à l'origine de la réalisation de ces derniers. Ce ne sont que des supports extérieurs réalisés par d'autres concepteurs que nous incluons dans notre projet. A noter que ces sources laissent quand même un impact environnemental. On retrouve notamment la pollution générée par la construction de ces derniers. C'est le cas des usines mais aussi des humains ravageant les paysages afin d'acheminer des matériaux nécessaires à la conception des produits. On peut prendre l'exemple du téléphone où il est nécessaire d'extraire des minerais. Cela serait responsable de la destruction des écosystèmes et serait également à l'origine de la pollution dans la nature (air, eau, etc.)

Voici un tableau résumant l'impact environnemental :

	Identification	Pourquoi?	Niveau d'impact estimé	Mesure	Comment?	Axes d'amélioration pour réduire impacts	Implémenté
Points d'analyse / Points de vigilance							
1. Conception fonctionnelle (liste de fonctionnalités, nombre de cas d'usage...)	probablement pas	La réalisation de notre projet a été faite qu'en Python donc il n'y a pas vraiment de source de pollution	aucun	quantifiable	Notre conception fonctionnelle n'a pas d'impact énergétique	aucun	non
2. Conception technique (type de langage, framework, APIs, bases, architecture...)	probablement pas			quantifiable	Notre conception technique n'a pas d'impact énergétique	aucun	oui
3. Développement (structure du code, base de données, green patterns...)	probablement pas			quantifiable	Notre structure de code n'a aucun impact énergétique	aucun	oui
4. Conception graphique (UI/UX, Ergonomie...)	probablement pas			quantifiable	Notre conception graphique n'a aucun impact	aucun	non
5. Hébergement (distance de transport, cloud-local, sources d'énergie...)	probablement pas	Nous ne stockons pas notre code dans des serveurs	aucun	quantifiable	Nous n'utilisons pas de cloud pour notre projet	aucun	non
6. Contenu (textes, images, vidéos...)	certainement	Il n'y a pas de consommation énergétique	aucun	quantifiable	Notre contenu n'a aucun impact énergétique	aucun	oui
8. User device (PC, tablet, smart phone...) et/ou éléments numériques d'un produit							
1. Consommation énergétique directe	certainement	Nos appareils nécessaires à notre projet consomment de l'énergie (caméra ou téléphone...)	modéré	quantifiable	Consommation lors des chargements des ordinateurs/téléphones, lors de l'utilisation de la caméra aussi	Mode économie d'énergie, éviter de laisser nos ordinateurs portables branchés s'il est chargé	oui
2. Durée de vie d'équipements existants (remplacements nécessaires)	probablement pas	Nous n'avons pas remplacé d'appareils	non-déterminé	quantifiable	nous n'avons effectué aucun remplacement	aucun	non
3. Besoin d'équipements neufs spécifiques ou complémentaires	certainement	On a eu besoin d'une caméra pour l'une des fonctionnalités de notre application	faible	quantifiable	Nous avons acheté une nouvelle caméra. Ici, il y a 2 sources de pollution. Le déplacement de l'individu achetant notre objet. La fabrication de l'objet (pollution de l'usine, dû à l'acheminement de matériaux, etc.)	Acheter notre appareil en magasin en veillant à se déplacer de manière écologique (vélo) et pas sur Amazon	oui
4. Consommables (cables, cartes, batteries, cartes mémoire, matériaux, boîtiers...)	probablement pas	L'utilisation de notre caméra, ordinateur, téléphone a fait intervenir des batteries (batteries externes)	faible	quantifiable	On a chargé les batteries de nos ordinateurs	On aurait pu mettre en mode économie d'énergie nos ordinateurs pour éviter la consommation électrique	non
5. Consommation énergie grise	probablement pas	Notre produit est virtuel	non-déterminé	quantifiable	On n'utilise pas d'énergie grise	aucun	non
6. Consommation eau virtuelle	probablement pas	Notre produit n'utilise pas d'eau	aucun	quantifiable	On n'utilise pas d'eau	aucun	non
7. Déchets, recyclage	probablement pas	Nous n'avons pas eu de déchets ni de recyclage. Cependant, cette source de consommation sera valable tôt ou tard lors du remplacement de nos ordinateurs / téléphones / caméras par exemple (lorsqu'ils seront obsolètes)	aucun	quantifiable	Notre produit étant virtuel nous n'avons pas de déchets provenant directement de notre application. Ils ne peuvent provenir que des supports extérieurs (ordinateurs, etc.)	aucun	non

IV. Planning prévisionnel

On a utilisé la méthode SCRUM pour travailler sur ce projet ainsi que l'outil Trello pour mieux s'organiser sur nos tâches respectives, les deadlines et les objectifs premiers à réaliser avant un certain temps.

Pour les rôles, nous nous sommes réparti les tâches lorsque nous avons mieux analysé les attentes techniques requises pour réaliser le projet. En revanche, nous pensions déjà tous travailler sur le code. De plus, nous avons affecté la partie « test de l'application » à Matthieu et Benjamin, la partie « fournir le matériel » à Marty et Guillaume. Quant à Ryan, il a dû surveiller notre avancement et nous aider, en cas de retard ou de difficultés (SCRUM Master).

Afin de bien organiser le code mais aussi l'avancement du projet, nous avons utilisé GitHub et nous avons créé une branche pour chaque membre. L'utilisation d'un Trello a également été un élément d'aide décisif dans notre organisation.

V. Déroulement des tâches

Découpage des tâches à réaliser :

Nom de la tâche	Précision
Code	Tout le monde
Poster	Tout le monde
Vidéo	Tout le monde

Calendrier :

Nom de l'étudiant	Tâche	Echéance	Remarques
Marty EVUORT Matthieu EVUORT	Ajout de la vidéo et de la caméra	04/03/2022	Le code python arrive à charger une vidéo téléchargée ainsi qu'une vidéo enregistrée par une caméra en temps réelle. Il est donc possible d'analyser des images en temps réel ainsi que différentes vidéos téléchargées et stockées dans notre ordinateur.
Tout le monde	Détection d'objet en mouvement	10/04/2022	Détection des objets filmés avec une caméra. Notre algorithme devait différencier les objets en mouvements de ceux immobiles.
Marty EVUORT Guillaume CLAVIER	Affichage du compteur	20/04/2022	A l'issue de cette action, un compteur est affiché sur l'écran de l'utilisateur

Guillaume Clavier Benjamin HOANG Marty EVUORT Matthieu EVUORT Ryan SEBBANE	Codage du compteur	05/05/2022	Incrémentation d'un compteur lorsqu'un véhicule est détecté. Pour cela, dès qu'un objet semblable à un véhicule (donc en mouvement) traversait notre ligne bleue, notre compteur devait le compter.
Marty EVUORT Matthieu EVUORT	Interface graphique	10/05/2022	Réalisation d'une interface graphique. C'est dans ce menu que l'utilisateur saisit son choix en fonction de son besoin.
Tout le monde	Perfection du code	13/05/2022	Rendre le code moins lourd, plus joli et plus adaptif à notre projet pour que le code ne consomme moins de mémoire.
Marty EVUORT Matthieu EVUORT Guillaume CLAVIER Ryan SEBBANE Benjamin HOANG	Réalisation du poster	25/05/2022	Ayant appris à promouvoir des objets, avoir un rendu joli et bien contrasté a été rapide. Pour cela, nous nous sommes servis de divers logiciels tels que Photoshop.
Tout le monde	Réalisation de la vidéo	29/05/2022	Grâce à nos logiciels de montage, nous avons pu réaliser la vidéo plus facilement.

VI. Conclusion de la réalisation du projet.

Nous avons trouvé ce projet très enrichissant et très important pour nous tous pour notre future carrière d'ingénieur. Cela nous a permis d'appréhender les mécaniques à avoir en entreprise afin de réaliser ce type de projet. Nous aurons travaillé de la même manière ou presque qu'un ingénieur. Nous avons également pu appliquer la méthode agile qui est nécessaire pour la bonne réalisation de ce genre de projet. Cela a aussi été une bonne occasion pour appliquer les notions étudiées durant le cours d'introduction au Delivery Project. L'utilisation du GitHub et du Trello a également été un élément décisif dans la réalisation de notre application.

Les notions abordées pour réaliser le projet est tout aussi captivante et intéressante. De nos jours nous entendons énormément les notions telles que le Deep Learning, Machine Learning, l'Intelligence Artificielle mais ne nous savons strictement pas comment cela fonctionne et quels sont les enjeux pour pratiquer ses notions. Avoir un petit aperçu de cela grâce au projet Transverse nous sera salutaire dans notre culture générale et nous ouvrira certainement des portes. Cela

nous a également apporté des informations quant aux spécialisations proposées par notre école. Il sera donc plus aisé de s'orienter plus tard.

Quant aux imprévus rencontrés, même si nous n'avons pas réussi à remplir tous nos objectifs (utilisation de la raspberry, différence entre les différents types de véhicules, reconnaissance des marques/modèles), nous avons beaucoup apprécié notre projet. D'autant plus que la plupart de nos objectifs ont bien été remplis (comptage des véhicules, utilisation d'une caméra en temps réel et en bonus d'une vidéo au format mp4, détection de plusieurs véhicules en même temps)

Malgré les imprévus survenus durant la réalisation de notre projet à savoir les problèmes d'outil ou bien l'existence de nouvelles alternatives plus simples, toutes les attentes du cahier des charges ont donc bien été respectées.

Finalement, il aurait été intéressant de posséder davantage de temps pour qu'on puisse améliorer notre application. Des mises à jour auraient été envisageables dans notre cas. Par exemple, nous aurions pu prendre en considération la vitesse, rajouter un système de prise de photos, classer les véhicules par modèle/marque, différencier les différents véhicules (moto, camion, voiture), proposer un système de reconnaissance faciale (donc du conducteur), etc. Il n'y a donc aucune limite à notre projet !

Sources

Tutoriel OpenCV Python - Traitement d'images - Vision par ordinateur :

https://www.kongakura.fr/article/OpenCV_Python_Tutoriel

<https://www.cours-gratuit.com/tutoriel-python/tutoriel-python-les-bases-de-traitement-dimages-en-python-opencv>

Détecter un objet avec OpenCV-Python :

<https://fr.acervolima.com/detecter-un-objet-avec-opencv-python/>

Loading Video Source – OpenCV with Python :

<https://www.youtube.com/watch?v=aUAVLhfVE5U>

LEARN OPENCV in 3 HOURS with Python :

<https://www.youtube.com/watch?v=WQeoO7MI0Bs>