

Bilan

Realtime keynote feedback

Projet Master 2 MIAGE Université Paris Nanterre

Rémy Pocquerusse

Ce projet m'a permis de découvrir de nombreuses technologies qui sont de plus en plus utilisées depuis quelques années, notamment avec l'apparition de Tensorflow qui a boosté le domaine du deep learning (c'est le framework qui est le plus utilisé aujourd'hui). J'ai travaillé avec OpenCV qui offre les outils nécessaires pour faire du traitement vidéo poussé et permet de travailler sur la construction d'algorithmes de type machine learning.

Durant ce projet j'ai principalement effectué les tâches suivantes : la mise en place du serveur REST avec le framework Django REST, la gestion du backend, notamment assurer le routage des requêtes. J'ai pu aussi travailler sur les algorithmes de détection de visages fournis par OpenCV. J'ai pu étudier leur fonctionnement et les méthodes utilisées pour les concevoir et les entraîner. J'ai essayé d'entraîner mes propres algorithmes et de ré-entraîner les algorithmes fournis par OpenCV mais la procédure est chronophage et la puissance de calcul nécessaire pour arriver à des résultats probants est trop importante. En effet la procédure est comme suit : il faut tout d'abord trouver les ressources pour "nourrir" l'algorithme (milliers d'images positives et négatives à analyser), puis entrer dans un fichier texte les coordonnées de chaque objet à détecter dans les images pour enfin implémenter les méthodes nécessaires pour entraîner l'algorithme. J'ai pu effectuer ce travail sur quelques images mais j'ai rapidement abandonné.

J'ai aussi eu pour responsabilité d'assurer l'intégration des différentes fonctionnalités (détection visages et émotions, etc.) côté backend. Un des problèmes que nous avons rencontré c'est la puissance de calcul nécessaire pour faire tourner les algorithmes. Il a fallu réduire considérablement le nombre d'images analysées toutes les secondes.

Choses faites : intégration des fonctionnalités (détection de l'émotion), correction d'un bug bloquant entre la gestion de l'émotion et le flux vidéo.

Avec ce projet j'ai donc pu travailler sur des technologies innovantes et sur des cas d'applications nouveaux. J'aurais aimé accorder autant de temps à ce projet qu'au projet que nous avons eu en M1 car je pense qu'il y avait encore beaucoup de choses intéressantes à traiter. Mais malgré le manque de temps je suis tout de même fier de ce que nous avons fait.

Oussema Oulhaci

Ce projet a été l'opportunité pour moi de découvrir et de mettre en pratique les nouvelles technologies concernant le deep learning avec les APIs de tensorflow et keras.

En effet, j'ai pu apprendre auparavant qu'est-ce que le machine learning et les réseaux de neurones à travers le cours donné par Andrew NG sur Coursera et c'est ce que j'ai pu faire tout le long de ce projet à travers la reconnaissance faciale et des émotions.

Mon travail durant ce projet a consisté à faire des recherches et implémenter le code qui permet de reconnaître un visage humain puis de classifier son émotion. J'ai pu donc découvrir la librairie OpenCV que j'ai dû télécharger pour re-builder toute la librairie sur mon Ubuntu afin de donner les commandes nécessaires pour pouvoir installer la librairie et ses dépendances sur notre environnement de développement Vagrant. Après cela j'ai expérimenté pas mal de code et forker des projets depuis github qui servent à faire de la reconnaissance faciale.

Après la reconnaissance faciale, j'ai pu m'occuper de la partie reconnaissance émotionnelle des personnes. Pour cette partie, j'ai pu découvrir le réentraînement et les algos cnn de tensorflow à travers de tutos. Ces tutos permettaient de ré entraîner des models sans pour autant avoir une puissance de calcul. Ce ré entraînement, permet de créer de nouvelle classe dans les outputs du cnn ce qui permet d'adapter en fonction du besoin. Malgré que cette librairie soit très utile, il faut tout de même un certain temps pour ré-entraîner les models. Je me suis donc dirigé vers les APIs Keras car il y avait déjà beaucoup de travail concernant la reconnaissance émotionnelle implémenter par des thésard et chercheurs. J'ai travaillé surtout avec Remy qui assurer l'intégration coté serveur pour savoir par exemple dans quelle forme les données renvoyés par la détection émotionnelle devaient être.

Ce projet était très intéressant mais malgré le manque temps nous avons pas le temps d'approfondir et de s'y investir à 100% ajouter de nouvelles fonctionnalités tels que la corrélation des émotions avec l'intensité de la voix etc.

Valentin Bouquet

Ce projet a été l'occasion pour moi de découvrir de nouvelles technologies et de sortir un peu des carcans habituels de la formation MIAGE.

Les technologies javascript sont de plus en plus populaires et les mentions des nouveaux framework comme React ainsi que les outils npm et webpack sont partout. Je souhaitais découvrir de nouvelles choses que nous n'avons pas la chance de découvrir en cours. J'ai redécouvert un langage, le javascript qui ne me plait toujours pas à cause d'une syntaxe mal défini et surtout de la prolifération d'outils et de bibliothèques pour contourner les limitations du langage comme l'absence de typage pouvant être contourné avec Typescript (langage transpilé à partir du javascript) ou l'absence de classe avant la spécification ECMAScript 2015 (assez récent donc). Parcourir les documentations en ligne est tout sauf agréable, les bibliothèques et les évolutions du langage avancent à une vitesse folle et il n'est pas rare de voir du code déprécié un peu partout et des syntaxes complètement différente à plusieurs endroits pour une même fonctionnalité.

Découvrir React par contre a été une expérience assez enrichissante. Le framework est assez proche des architectures mobiles (Android et IOS), il adopte par exemple le cycle de vie des conteneurs visuels et la programmation asynchrone pour fluidifier la navigation. C'est d'ailleurs pour cela qu'il est possible de développer des applications en javascript avec React sur Android et IOS. Bien qu'étant une bibliothèque pour faciliter le développement d'interface, il ne faut sous-estimer la bibliothèque. Suivre la philosophie de celle-ci et adapter de nouveaux patterns de programmation n'est pas toujours facile.

Du côté des outils de développement, je n'ai rien découvert mais j'ai pu consolider mes bases avec l'utilisation de vagrant et git que je trouve toujours très efficace ainsi que la facilité d'utilisation de TravisCI pour vérifier que nos builds passent même si j'aurais aimé avoir le temps de diminuer la durée de vérification de dépendances (~30minutes actuellement).

J'ai aussi eu l'occasion de gérer la synchronisation entre l'interface (front-end) et le serveur (back-end) pour l'échange de données. J'ai assez vite compris, que nous devions nous entendre sur le travail réalisé et les données échangées pour que l'on puisse développer de manière asynchrone pour ensuite scinder ce que nous avons réalisé.

Je n'ai malheureusement pas mis autant les mains dans le cambouis sur les outils de machine learning, deep learning et de reconnaissance d'images. Mes collègues ont fait la recherche et ont développés les outils, j'ai tout de même pu discuter avec eux de leurs choix et intégrer ceux-ci au projet.

Je regrette tout de même que nous n'ayons pas eu le temps d'avancer plus sur le projet. A cause de la charge de travail cette année, j'ai n'ai pas eu la sensation de m'être suffisamment investi dans le projet. Je pense que nous aurions pu aller plus loin: croisement des données, analyse audio etc.