Code open source contre gros système

57 lignes de code et deux ou trois bidules électroniques feraient aussi bien voire mieux qu'un gros système coûteux. Telle est la démonstration que vient de faire un développeur australien.
L'expérience que relate ici Tait Brown relève du *proof of concept*, la démonstration de faisabilité. La spectaculaire économie de moyens numériques et financiers qu'il démontre avec 57 lignes de code *open source* et des appareils à la portée d'un bidouilleur ordinaire n'est peut-être pas une solution adaptable à grande échelle pour *remplacer* les puissants et massifs systèmes propriétaires mis en place par des entreprises. Pas plus que les services libres de Framasoft n'ambitionnent de *remplacer* les GAFAM, mais démontrent que des solutions alternatives libres et plus respectueuses sont possibles et viables, et de plus en plus disponibles.

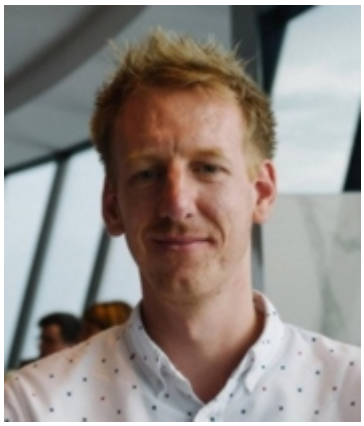
Outre le pied de nez réjouissant du hacker occasionnel aux institutions locales (ici, la police de l'état australien de Victoria) qui ont confié un traitement informatique à des sociétés privées, ce petit témoignage ouvre au moins une question : le code est mis au service de la police au bénéfice des citoyens (repérer les voitures volées, pister la délinquance…), mais peut fort bien ne faire qu'augmenter la surveillance de masse au détriment des mêmes citoyens, avec les conséquences pas du tout triviales qu'on connaît et dénonce régulièrement. Le fait que le code *open source* soit *auditable* est-il un garde-fou suffisant ?

Comment j'ai recréé un logiciel de 86 millions de dollars en 57 lignes de code

par Tait Brown

Publication originale : How I replicated an $86 million project in 57 lines of code
Traduction Framalang : xi, Lyn., goofy, framasky, Lumibd, Penguin

Quand un essai à base de technologie open source fait le boulot « suffisamment bien ».

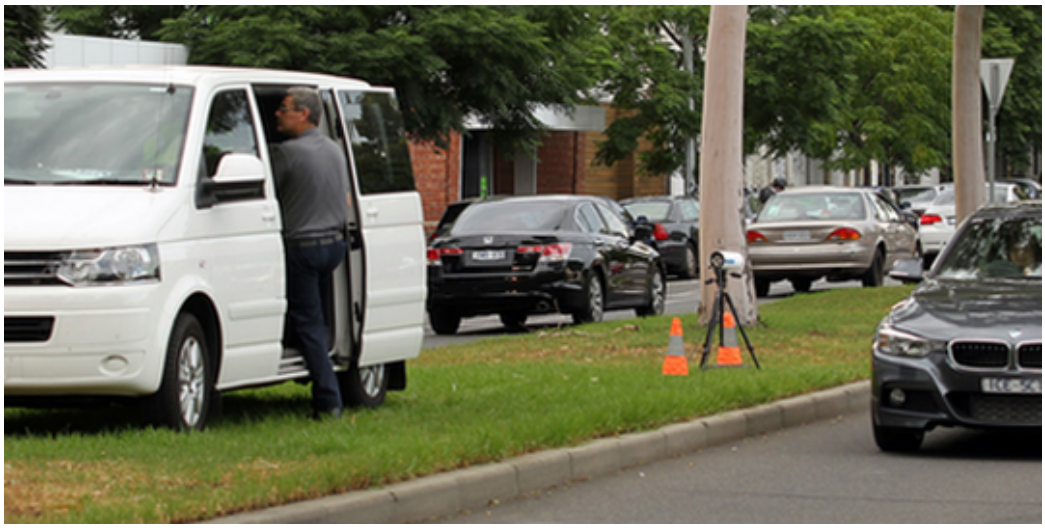


La police est le principal acteur du maintien de l'ordre dans l'État du Victoria, en Australie. Dans cet État, plus de 16 000 véhicules ont été volés l'an passé, pour un coût d'environ 170 millions de dollars. Afin de lutter contre le vol de voitures, la police teste différentes solutions technologiques.

Pour aider à prévenir les ventes frauduleuses de véhicules volés, VicRoads propose déjà un service en ligne qui permet de vérifier le statut d'un véhicule en saisissant son numéro d'immatriculation. L'État a également investi dans un scanner de plaque minéralogique : une caméra fixe sur trépied qui analyse la circulation pour identifier automatiquement les véhicules volés.

Ne me demandez pas pourquoi, mais un après-midi, j'ai eu envie de réaliser un prototype de scanner de plaques minéralogiques embarqué dans une voiture, qui signalerait automatiquement tout véhicule volé ou non immatriculé. Je savais que tous les composants nécessaires existaient et je me suis demandé à quel point il serait compliqué de les relier entre eux.

Mais c’est après quelques recherches sur Google que j’ai découvert que la Police de l’État du Victoria avait récemment testé un appareil similaire dont le coût de déploiement était estimé à 86 millions de dollars australiens. Un commentateur futé a fait remarquer que 86 millions de dollars pour équiper 220 véhicules, cela représentait 390 909 AUSD par véhicule.
On devait pouvoir faire mieux que ça.



Le système existant qui scanne les plaques minéralogiques avec une caméra fixe

Les critères de réussite

Avant de commencer, j’ai défini à quelles exigences clés devait répondre la conception de ce produit.

Le traitement de l’image doit être effectué localement
Transmettre en continu le flux vidéo vers un site de traitement centralisé semblait l’approche la moins efficace pour répondre au problème. La facture pour la transmission des données serait énorme, de plus le temps de réponse du réseau ne ferait que ralentir un processus potentiellement assez long.
Bien qu’un algorithme d’apprentissage automatique centralisé ne puisse que gagner en précision au fil du temps, je voulais savoir si une mise en œuvre locale sur un périphérique serait

« suffisamment bonne ».

Cela doit fonctionner avec des images de basse qualité
Je n'avais ni caméra compatible avec un Raspberry Pi, ni webcam USB, j'ai donc utilisé des séquences vidéo issues de dashcam [NdT : caméra installée dans un véhicule pour enregistrer ce que voit le conducteur], c'était immédiatement disponible et une source idéale de données d'échantillonnage. En prime, les vidéos dashcam ont, en général, la même qualité que les images des caméras embarquées sur les véhicules.

Cela doit reposer sur une technologie *open source*
En utilisant un logiciel propriétaire, vous vous ferez arnaquer chaque fois que vous demanderez un changement ou une amélioration, et l'arnaque se poursuivra pour chaque demande ultérieure. Utiliser une technologie *open source* évite ce genre de prise de tête.

Solution

Pour l'expliquer simplement, avec ma solution, le logiciel prend une image à partir d'une vidéo *dashcam*, puis l'envoie vers un système de reconnaissance des plaques minéralogiques *open source* installé localement dans l'appareil, il interroge ensuite le service de contrôle des plaques d'immatriculation et renvoie le résultat pour affichage.
Les données renvoyées à l'appareil installé dans le véhicule de police comprennent : la marque et le modèle du véhicule (pour vérifier si seules les plaques ont été volées), le statut de l'immatriculation et la notification d'un éventuel vol du véhicule.
Si cela semble plutôt simple, c'est parce que c'est vraiment le cas. Le traitement de l'image, par exemple, peut être opéré par la bibliothèque openalpr. Voici vraiment tout ce qu'il faut pour reconnaître les caractères sur les plaques minéralogiques :

```
 openalpr.IdentifyLicense(imagePath, function (error, output)
{
```

```
// handle result
});
```
(le code est sur Github)

Mise en garde mineure

L’accès public aux API de VicRoads n’étant pas disponible, les vérifications de plaques d’immatriculation se font par le biais du *web scraping* (NdT : une technique d’extraction automatisée du contenu de sites web) pour ce prototype. C’est une pratique généralement désapprouvée, mais il ne s’agit ici que d’un test de faisabilité et je ne surcharge pas les serveurs de quiconque.

Voici à quoi ressemble mon code, vraiment pas propre, utilisé pour tester la fiabilité de la récupération de données :

```
// Open form and submit enquire for `rego`
function getInfo(rego) {
        horseman
          .userAgent('Mozilla/5.0 (Windows NT 6.1; WOW64; rv:27.0) Gecko/20100101 Firefox/27.0')
          .open(url)
          .type('#registration-number-ctrl input[type=text]', rego)
          .click('.btn-holder input')
          .waitForSelector('.ctrl-holder.ctrl-readonly')
          .html()
          .then(function(body) {
                console.log(processInfo(body, rego));
            return horseman.close();
          });
}

// Scrape the results for key info
function processInfo(html, rego) {
        var $ = cheerio.load(html);
        var vehicle = $('label.label').filter(function() {
          return $(this).text().trim() === 'Vehicle:';
        }).next().text().trim();

        var stolen = $('label.label').filter(function() {
          return $(this).text().trim() === 'Stolen status:';
        }).next().text().trim();

        var registration = $('label.label').filter(function() {
          return $(this).text().trim() === 'Registration status & expiry date:';
        }).next().text().trim();

        return {
                rego,
                vehicle,
                stolen,
                registration
        };
}
```

(le code est sur Github)

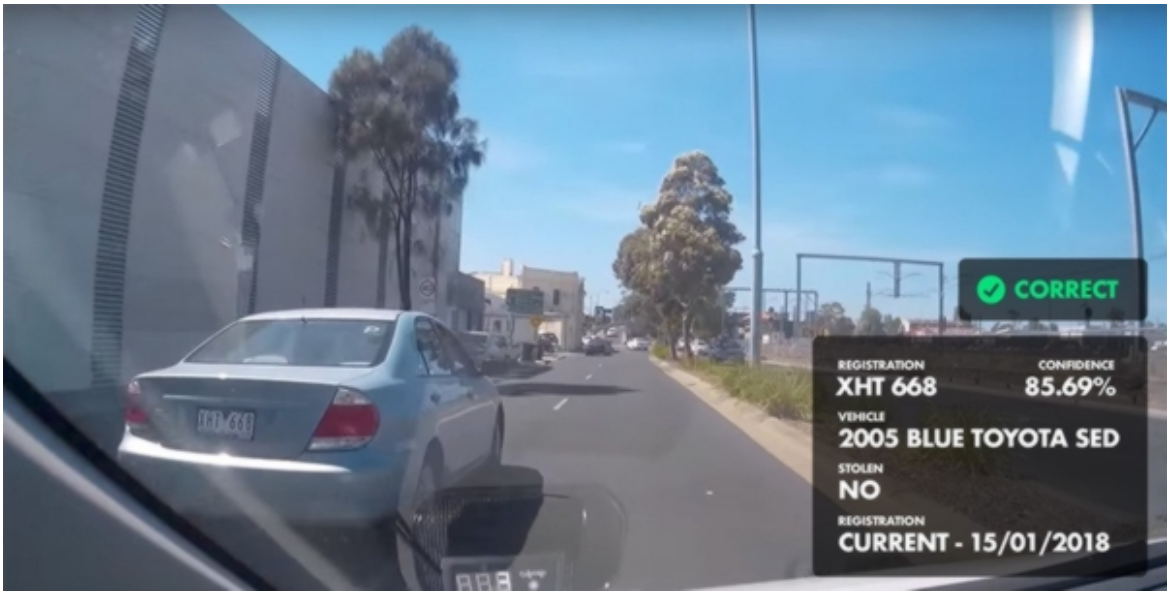
Résultats

Je dois dire que j'ai été agréablement surpris.

Je m'attendais à ce que la reconnaissance des plaques minéralogiques *open source* soit plutôt mauvaise. De plus, les algorithmes de reconnaissance d'images ne sont probablement pas optimisés pour les plaques d'immatriculation

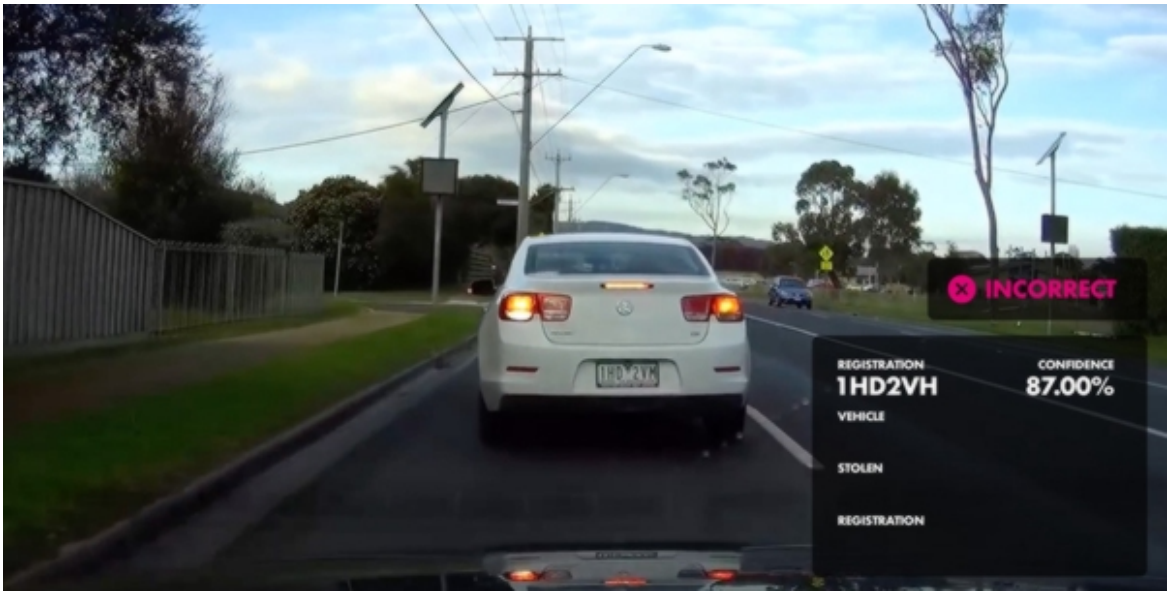
australiennes.

Le logiciel a été capable de reconnaître les plaques d’immatriculation dans un champ de vision large.



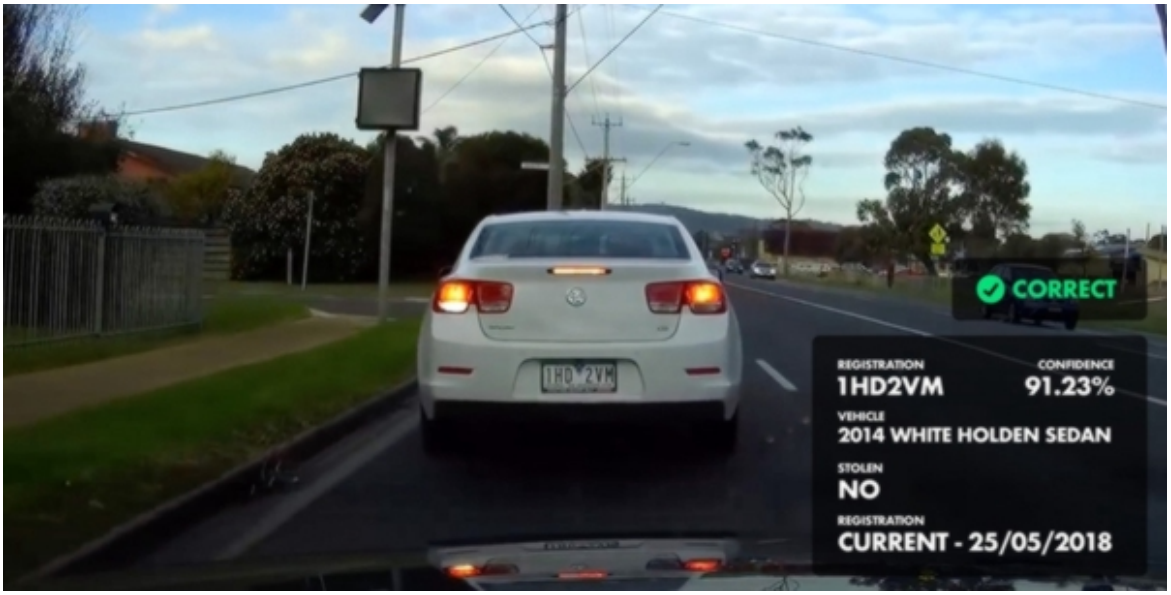
Annotations ajoutées sur l’image. Plaque minéralogique identifiée malgré les reflets et l’axe de prise de vue

Toutefois, le logiciel a parfois des problèmes avec des lettres particulières.



Mauvaise lecture de la plaque, le logiciel a confondu le M et le H

Mais… il finit par les corriger :



Quelques images plus tard, le M est correctement identifié à un niveau de confiance plus élevé

Comme vous pouvez le voir dans les deux images ci-dessus, le traitement de l'image quelques images plus tard a bondi d'un indice de confiance de 87% à un petit peu plus de 91%.

Il s'agit de solutions très simples au niveau de la programmation, qui n'excluent pas l'entraînement du logiciel de reconnaissance des plaques d'immatriculation avec un ensemble de données locales.
Je suis certain que la précision pourrait être améliorée en augmentant le taux d'échantillonnage, puis en triant suivant le niveau de confiance le plus élevé. On pourrait aussi fixer un seuil qui n'accepterait qu'une confiance supérieure à 90% avant de valider le numéro d'enregistrement.
Il s'agit de choses très simples au niveau de la programmation, qui n'excluent pas l'entraînement du logiciel de reconnaissance des plaques d'immatriculation avec un jeu de données locales.

La question à 86 000 000 dollars

Pour être honnête, je n'ai absolument aucune idée de ce que le

chiffre de 86 millions de dollars inclut – et je ne peux pas non plus parler de la précision d'un outil *open source* sans entraînement spécifique adapté au pays par rapport au système pilote BlueNet.
Je m'attendrais à ce qu'une partie de ce budget comprenne le remplacement de plusieurs bases de données et applications logicielles existantes pour répondre à des demandes de renseignements sur les plaques d'immatriculation à haute fréquence et à faible latence plusieurs fois par seconde par véhicule.
D'un autre côté, le coût de 391 000 dollars par véhicule semble assez élevé, surtout si le BlueNet n'est pas particulièrement précis et qu'il n' existe pas de projets informatiques à grande échelle pour la mise hors service ou la mise à niveau des systèmes dépendants.

Applications futures

Bien qu'on puisse aisément être soucieux de la nature orwellienne d'un réseau qui fonctionne en continu de mouchards à plaques minéralogiques, cette technologie a de nombreuses applications positives. Imaginez un système passif qui analyse les autres automobilistes à la recherche d'une voiture de ravisseurs et qui avertit automatiquement et en temps réel les autorités et les membres de la famille de leur emplacement et de leur direction.

Les véhicules Tesla regorgent déjà de caméras et de capteurs capables de recevoir des mises à jour OTA (NdT : Over The Air, c'est-à-dire des mises à jour à distance) – imaginez qu'on puisse en faire une flotte virtuelle de bons Samaritains. Les conducteurs Uber et Lyft pourraient également être équipés de ces dispositifs pour augmenter considérablement leur zone de couverture.

En utilisant la technologie *open source* et les composants existants, il semble possible d'offrir une solution qui offre un taux de rendement beaucoup plus élevé – pour un

investissement bien inférieur à 86 millions de dollars.