Reconnaissance de la langue des signes

Rapport technique

Patricia, Eva, Erwan

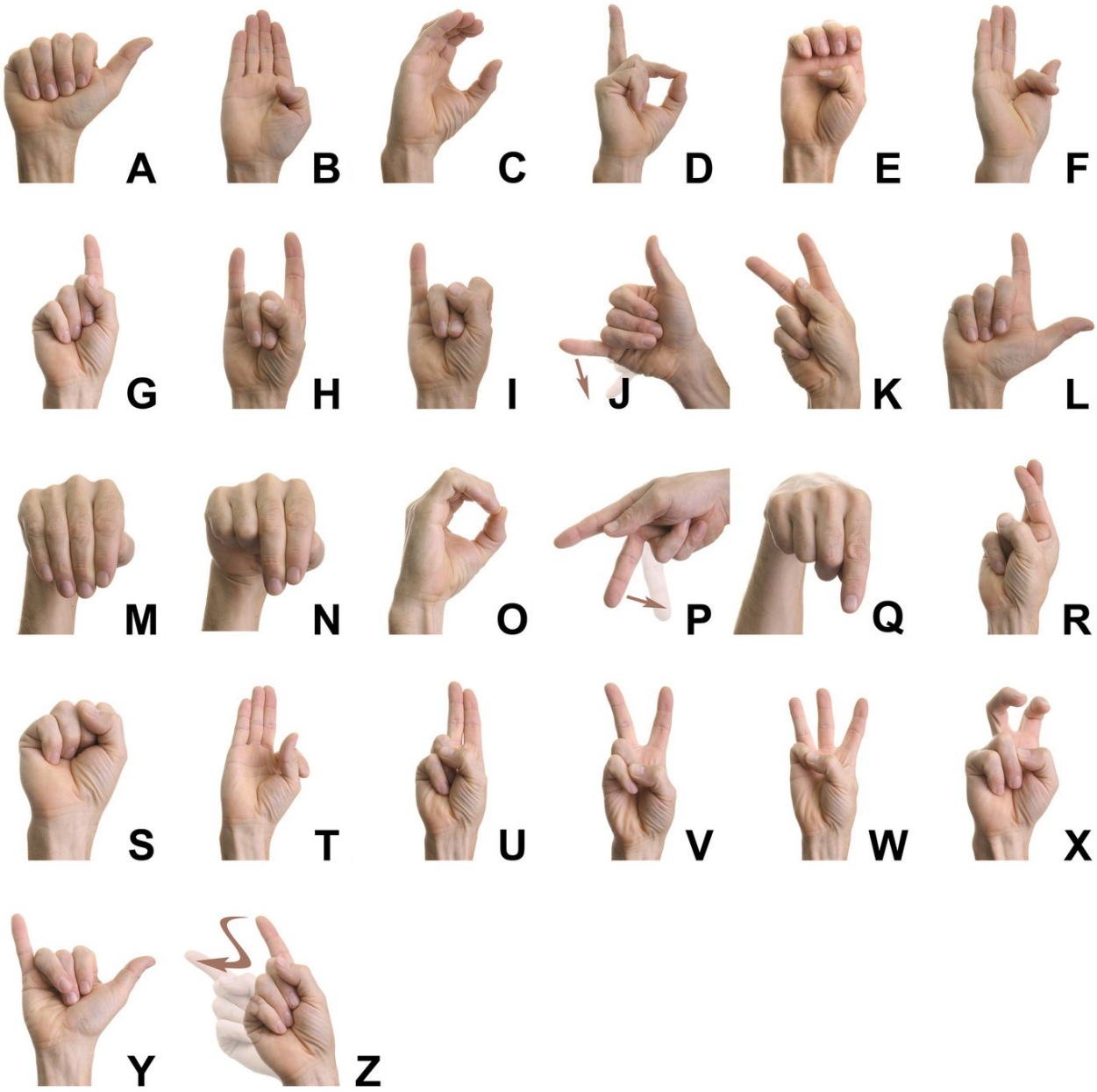


Table des matières

I. Le Dataset

Le dataset contient des photo de main faisant les lettre alphabétique du langage des signe.

I.1. Creation

Pour la création du dataset nous avons pris nos main en photo avec un téléphone portable.

I.2. Contenu

Le dataset est formé d'environ 254 photos de signes qui correspondent aux lettres de l’alphabet de A-Z sans J et Z, car ses pour faire ces deux lettre il faut bouger la main ce qui est compliqué a prendre en photo.

Voici une image du dataset :

II. Setup du modèle

Pour ce projet nous avons utilisé l'*object detection API* de Tensorflow avec le modèle *ssd mobilnet v2 320x320*. L'*object detection API* a besoin d'une *label map* et de fichier *tfrecord*, ce que nous allons créer avant d'utiliser le modèle.

Nous avons séparé les images, en données d'entraînements et en données de test, a la main.

II.1. Label Map

Pour pouvoir classifier nous avons besoin d'associer un nombre a chaque classes. Ici nous avons associer un nombre a chaque lettre : 1 pour A, 2 pour B, etc... Pour ce faire nous avons créer un fichier `.pbtxt` qui contient un dictionnaire avec les association nécessaire pour la classification. Elle sont sous la forme :

```
item {
     name:'A'
     id:1
}
```

II.2. TF records

Ensuite nous avons besoin des tf records, un pour les données d'entraînements et l'autre pour les données de test. Nous les créons grâce au script `generate_tfrecord.py` fournit.

II.3. Le modèle

Pour le modelé nous avons décidé de faire un modèle qui détectera la main sur l'image et qui la classifiera dans une des lettre, pour ce faire nous allons faire du *transfer learning* avec un modèle déjà existant de *tensorflow zoo* et de l'*object detection API.*

II.3.A. Téléchargement

Nous allons ensuite télécharger le *github* de *tensorflow/models*, nous aurons besoin de scripts se trouvant dans ce *github*.

Pour ce faire on peut soit `git clone` soit le télécharger à la main : Tensorflow/models github
et a cette adresse pour le modèle en lui mème : Liste des modèle Tensorflow

II.3.B. Configuration

Pour configurer la modèle il faut modifier le fichier `pipeline.config` qui se trouve dans le dossier du modèle télécharger.

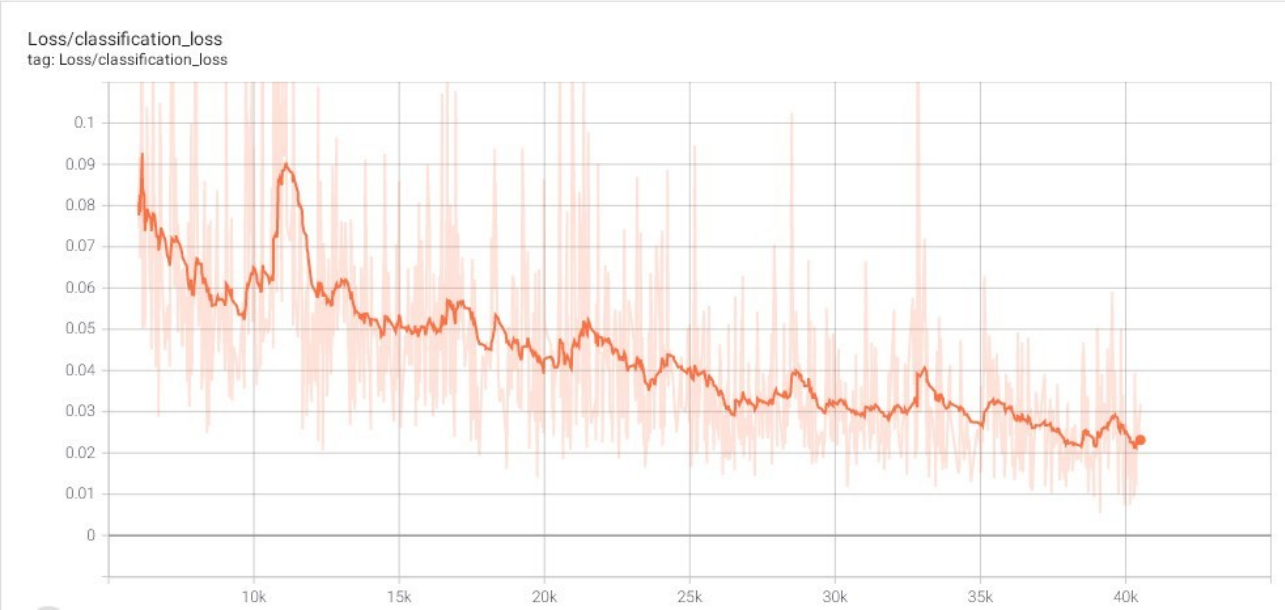
Dans notre cas nous avons fait les modification suivante :

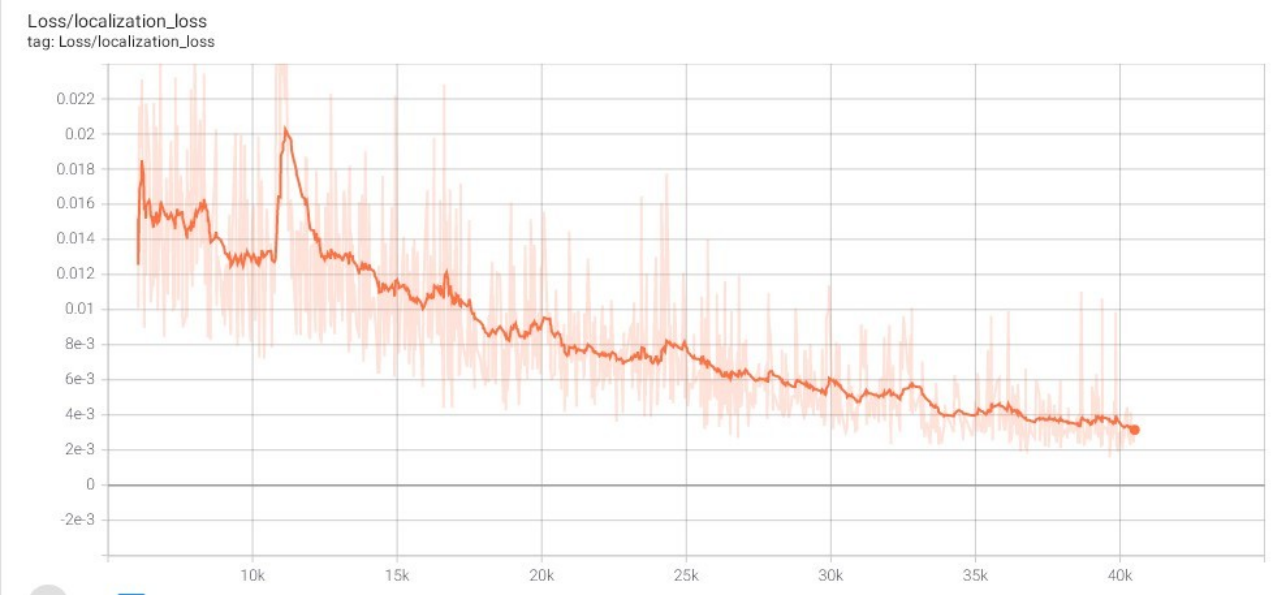
- nombre de classe a 26
- *batche_size* a 6
- chagement du *chekpoint*
- *chekpoint type* a detection
- le chemin vers la *label map*
- les chemins vers les *tfrecords*

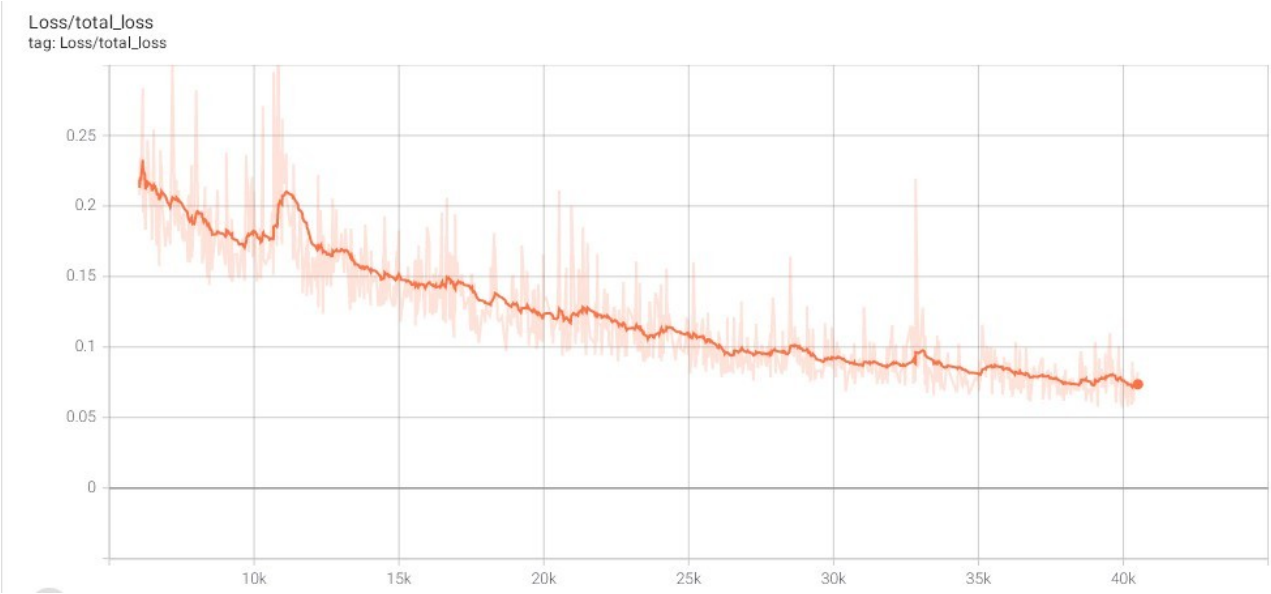
III. Évaluation

Pour évaluer notre modèle nous avons utiliser *Tensorboard*, un outils de Tensorflow.

III.1. Graphiques







Grace a ses graphique nous pouvoir voire que la loss de classification et de détection diminue au fur des *epochs*, mais que nous pouvons faire mieux.

IV. Tests

Nous avons aussi fait des tests sur des images ou en temps réel avec une camera.
Nous allons ici montre que les test sur image, mais le test sur la vidéo ce trouve dans le notebook *`signe_detection.`*

IV.1. Test sur image

Pour faire des test sur une image il faut d'abord que le modèle soit charger, puis que l'image soit sous la forme d'un tensor.

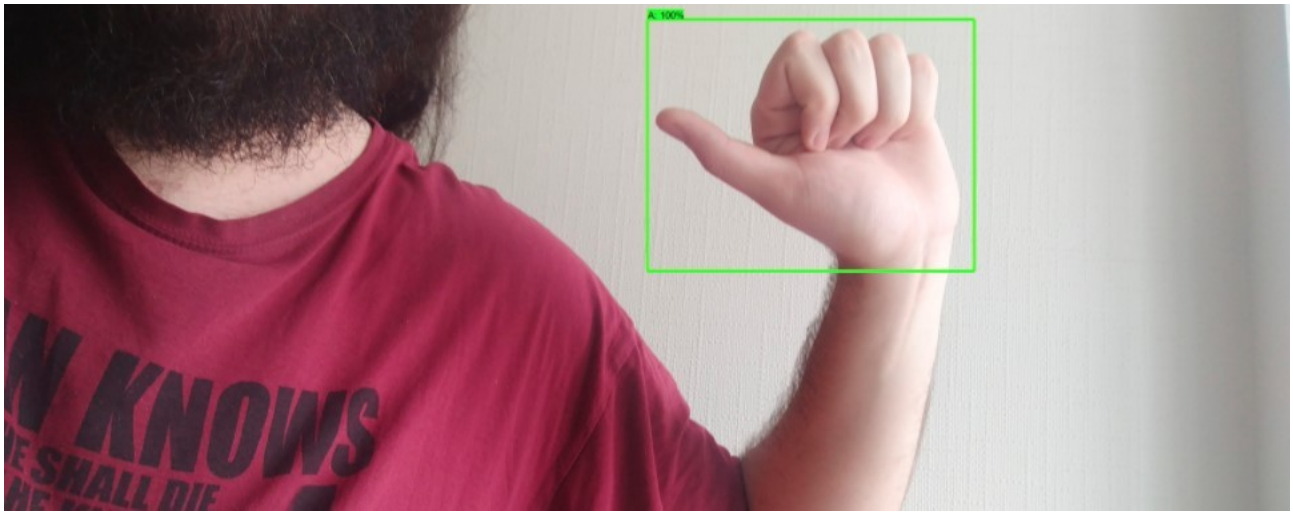
Pour ce faire nous allons utiliser les scripts fournit par Tensorflow :

- *`tf.saved_model.load()`*
- *`tf.convert_to_tensor()`*

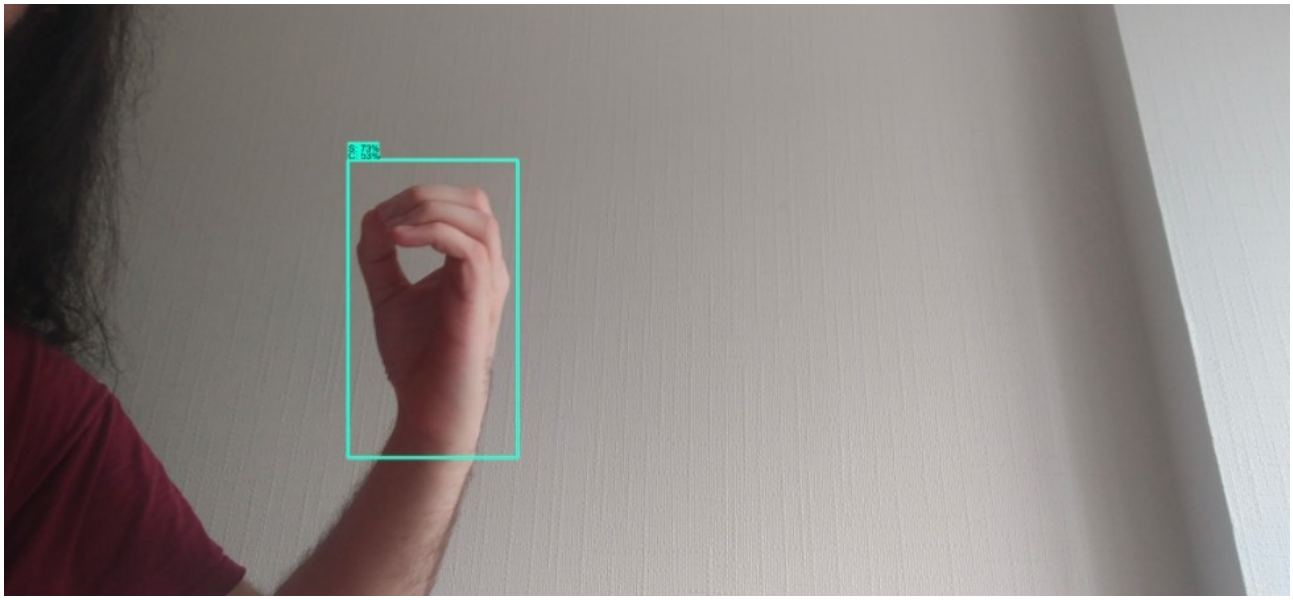
une fois l'image dans le forma souhaité il suffit juste de faire une inferance dessus et de récupérer la l'index de la classe prédite et d'utiliser la label map pour avoir le nom.

Test sur deux images :

A :



O :



Sur ces deux test on peu voir que pour le A il n’y a aucun problème, mais que pour le O le modele hésite entre deux lettre.

V. Conclusion et recommandations

Nous avons pu voir que le modèle est un peux compliqué a mettre en place mais demande moins de code que un modèle fait soit même. Nous avons utiliser du *transfer leraning* c’est pour ça que en quelques étapes nous avons pu avoir un modèle correcte mais améliorable. Le plus gros problème de ce modèle et qu’il dépend beaucoup de la lumière et de l’inclinaison de la main sur l’image, c’est pour quoi nous recommandons de faire un plus grand dataset avec des image avec le plus d’angle de main possible et pour la lumière soit le plus de lumière différente soit une unique lumière.