API WEB

Pour l’API web nous avons choisi de nous orienter vers un serveur node.js pour le back-end en utilisant React.js pour la partie front-end. Ce sont des technologies javascript toutes les deux, ce qui fait que c’est finalement plus homogène.

Nous avons simplement créé deux routes :

* Une route (/app) qui permet de délivrer l’application React (front) au client
* Une seconde route (/submit) qui permet d’envoyer l’image que le client veut analyser

Sur la partie front-end, le client a la possibilité d’uploader une image, seules des images peuvent être uploadées, il y a un mécanisme qui permet de vérifier cela.

Une fois l’image uploadée, celle-ci est automatiquement redimensionnée pour le moment à des dimensions de 100 x 100 px (pour être le plus proche des images du dataset elles-mêmes redimensionnées en 100 x 100px).

A ce moment-là, il ne reste plus qu’à envoyer l’image en question à l’API Python qui traitera l’image et nous reverra une réponse (drapeau Français, drapeau Américain etc…). Avec cette réponse, il n’y plus qu’a l’envoyer au client et afficher le drapeau qui correspond bien.

Si une erreur quelconque se produit le client est averti, par un message lui informant qu’il y a eu un problème avec le serveur ou avec sa connexion.

Dataset

A l’heure actuelle (28/05/2019), le dataset est composé d’environ 3000 images de drapeau français et 1300 drapeaux américain.

Pour le construire nous avons utilisé un script Python capable de télécharger automatiquement les images de Google Images en lui donnant en paramètre des keywords en l’occurrence : « drapeau francais », « french flag », « drapeau americain », « american flag » et d’autres combinaisons pour trouver un maximum d’images de drapeau différents.

Par la suite, pour 2 raisons nous avons choisi de redimensionner toutes les images à une taille de 100 x 100px.

Premièrement, une image de 100 par 100 px totalise 100\*100pixels soit 10000 pixels en sachant que l’on doit en traiter un très grand nombre d’image (3000 pour les drapeaux FR) il est plus rapide de traiter des images d’une taille plus petite.

Deuxièmement, au niveau de l’apprentissage nous pensons que cela permettra d’homogénéiser les résultats car l’algorithme ne sera perturbé par le fait qu’une image fasse 1000\*1000 ou 1920\*1080 ou peu importe la taille à la base.