Sorbonne Université
Master ANDROIDE

Reconnaissance de nombres contenus sur surfaces sphériques : application au tirage du loto

UE de projet M1

Antony LECLERC – Shirel AMOZIEG – Zhirui CAI
Sous la direction de : Thibaut Lust

2023 – 2024

Table des matières

Chapitre 1

Introduction

Le projet avait pour but de développer des techniques de reconnaissance de nombres sur une surface sphérique, les algorithmes d'aujourd'hui étant principalement adaptés pour reconnaître des nombres sur surface plane, on peut s'attendre à ce que ces derniers soient inadaptés si on les applique sur des surfaces sphériques.

Lien vers le dépôt git du projet.

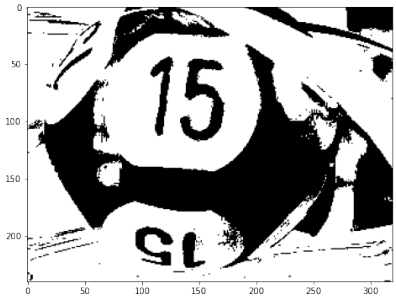
Chapitre 2

État de l'art

Les méthodes déjà existantes pour cela sont en général des méthodes se basant sur l'*OCR* (pour « *Optical Character Recognition* »), soit la Reconnaissance Optique de Caractères.

Le principe de l'*OCR* est de lire une image, et de binariser cette dernière, les zones sombres de l'image seront classées comme 0, et les zones claires comme 1. Et par la suite effectuer un *matching pattern* avec ce que le programme utilisé connait déjà.

(a) Image normale



(b) Image binarisée

FIGURE 2.1 – Exemple d'image binarisée pour *OCR*.

Chapitre 3

Contribution

Les deux outils principaux que nous avons ainsi choisi d'utiliser sont des CNN (« *Convolutional Neural Network* »), soit Réseau de Neurones Convolutionnel en Français. Ces derniers sont particulièrement adaptés pour travailler sur des images.

Ainsi que le langage de programmation Python.

Plusieurs outils s'offraient à nous :

- Pytorch
- Tensorflow
- Keras
- Vertex AI
- Scikit-Learn

Tous ont cependant leurs avantages et leurs inconvénients.

	Avantages	Inconvénients
PyTorch	• Simple d'utilisation • Rapide • Efficace	• Peu stable • Petite communauté
TensorFlow	• Beaucoup d'outils • Flexible • Grande communauté	• Difficile à prendre en main (pour les débutants) • Lent
Keras	• Se base sur TensorFlow • Prototypes rapides et faciles	• Pour des implémentations rapides • Petits jeux de données • Petite communauté
Vertex AI	• Bien pour les néophytes de l'IA • Conventions • Beaucoup d'outils	• Conventions, changer la manière dont Vertex fonctionne peut être problématique • Anciens codes souvent incompatibles
Scikit-Learn	• Facile à utiliser • Accessible • Grande communauté internationale	• Inadapté à l'apprentissage profond • Moins efficace que TensorFlow

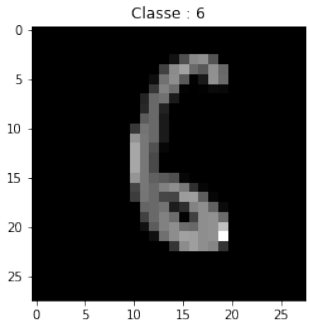
TABLE 3.1 – Comparaison des outils disponibles

Nous avons ainsi choisi pour ce projet d'utiliser la bibliothèque **Tensorflow** en plus de **Keras**, (Keras se basant sur Tensorflow, les deux étant ainsi complémentaires).

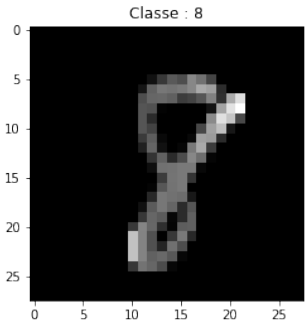
Chapitre 4

Reconnaissance de nombres sur surface plane

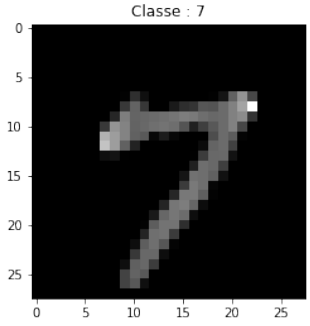
Pour prendre en main les réseaux de neurones (non convolutionnel dans un premier temps). La création de neurones avec Tensorflow se fait en quelques lignes. Par la suite pour un apprentissage sur surface plane, nous avons choisi d'entraîner notre réseau de neurones sur la base de donnée MNIST qui offre une multitutde d'images de taille 28×28 écrit à la main.



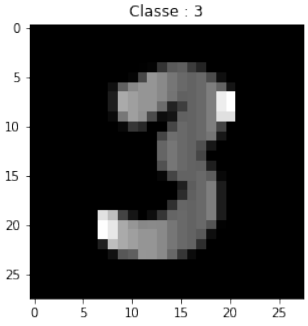
(a) Un 6 sur la base de donnée MNIST



(b) Un 8 sur la base de donnée MNIST



(c) Un 7 sur la base de donnée MNIST



(d) Un 3 sur la base de donnée MNIST

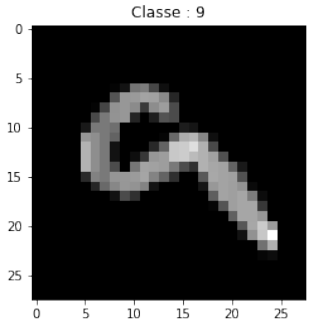
FIGURE 4.1 – Numéros écrits à la main dans la base de données MNIST

Un premier problème se pose cependant, les images disponibles sur la base de donnée MNIST sont des images de taille 28×28 pixels, tandis que nos images de boules de Loto sont de taille 240×320 pixels.

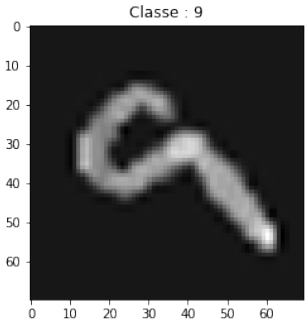
Pour remédier à cela, nous avons dans un premier temps extrait à la main, entre 3 à 5 images par numéro (de 1 à 50) autour du numéro, une zone de 70×70 pixels.

Les réseaux de neurones demandant cepandant une taille d'images identique à chaque fois, les images 28×28 de MNIST sont problématiques.

La librairie 'opencv-python' (*cv2*) offre la possibilité de redimensionner des images, opération que nous avons appliqué sur les datasets d'entraînement et de tests sur MNIST.



(a) Un 9 en taille 28×28



(b) Un 9 en taille $70x70$

FIGURE 4.2 – Comparaison d'une image de la base de donnée MNIST.
Une de taille 28×28.
L'autre de taille 70×70.

On peut aussi remarquer que le redimensionnement de l'image ayant necéssité une interpolation, l'image est légèrement floutée par rapport à l'image initiale.

Chapitre 5

Conclusion

La conclusion de votre rapport doit brièvement résumer ce que vous avez fait, en mettant en lumière les points forts et les points faibles de votre travail. Enfin, il faut décrire les perspectives de poursuite qui peuvent être envisagées.

Annexe A

Cahier des charges

Rappel du cahier des charges.

Annexe B

Manuel utilisateur

Un manuel utilisateur permettant la prise en main de votre code.