Mini Projet IA

Classification 'One-hundred plant species leaves'

En utilisant les reseaux de neurons

Par : Hadjerci Mohamed Allaeddine
M1 MIV

1 Introduction

La classification est une tache très importante dans la vision artificiel (Computer Vision) qui consiste a catégoriser une image selon l'objet dominant dans l'image par exemple si une image contient plusieurs de chats et un seul petit chien l'image va être classifier comme chat.[1]
Le but de ce projet et de construire un classificateur qui permet d'identifier le nom d'une plante selon certaine caractéristique extraite depuis des images de leur feuille en utilisant les réseaux de neurons.

2 Outils

2.1 Dataset

Le dataset compris 100 espèce de plant, pour chaque espèce il ya 16 différente spécimens et pour chaque spécimen il ya 192 caractéristique extraite depuis des images de feuille de la plante qui sont 64 information sur la marge (margin), 64 information sur la forme (shape) et 64 sur la texture.[2][3]

Donc en total on a :

- 1600 'features'.
- 192 attribut pour chaque 'feature'.
- 100 labels différente.

2.2 Langage de programmation

Dans ce projet, Python est utilisée comme langage de programmation c'est un langage 'open-source' qui est très facile et offre plusieurs de bibliothèque pour 'data science' qui sont facile a manipule comme numpy, pickle, matplotlib ... etc.

2.3 Environnement d'apprentissage

Pour l'environnement d'apprentissage on va utiliser Tensorflow avec keras.
Tensoflow c'est une bibliothèque ou un environnement d'apprentissage 'open-source' développer par Google qui permet une manipulation des réseaux de neurons.[4]

Keras c'est bibliothèque qui marche avec environnement d'apprentissage Tensorflow ou Theano, Keras offre une manipulation des réseaux de neurons en haut-niveau c'est a dire elle facilite la manipulation des réseaux de neurons.[5]

2.4 Tensorboard

C'est un outil de tensorflow qui permet de visualiser les graphes d'apprentissage qui contient des information comme :

Accuracy (acc) : C'est la précision sur les données de batch.
Loss : C'est l'erreur sur les données de batch.
Validation Accuracy (val acc) : C'est la précision sur les données de validation.
Validation loss (val loss) : C'est l'erreur sur les données de batch.

3 Traitement

3.1 Preprocessing

Avant de commencer l'apprentissage on doit commencer par 'preprocessing' qui consiste a traiter les donnée de datasets pour l'utiliser dans l'apprentissage.
Les données de dataset sont 3 fichier (margin, shape, texture) chaqun des fichier contient 64 information pour traiter ces données on utilise les étapes suivants.

3.1.1 Fusion des donnée

Comme les données sont dans 3 fichiers différents on doit les fusionner dans un seul fichier donc le fichier de devient pour chaque feature il contient 192 information 64 sur margin et 64 sur shape ensuite 64 sur texte.
Note : le fichier contient 1600 features en total mais comme il ya des données manquant sur la texture d'une spécimen de Acer Campster on est besoin de la supprimer donc le fichier contient au total 1599 feature après la fusion.

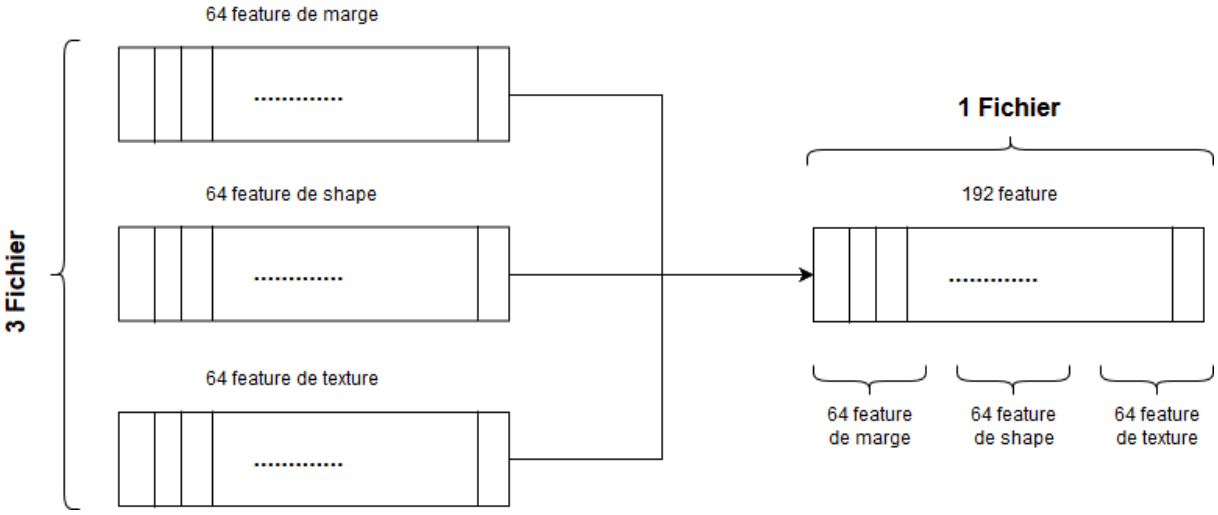


FIGURE 1 – Comment on Fusionne

3.1.2 Générer Labels

Pour les labels on génère one-hot arrays pour chaque label c'est dire on a des one-hot arrays de taille 100 car on a 100 espèce de plant différente,après la génèration des one-hot arrays on doit les enregistrer dans un dictionnaire pour les utilisée au plutard.

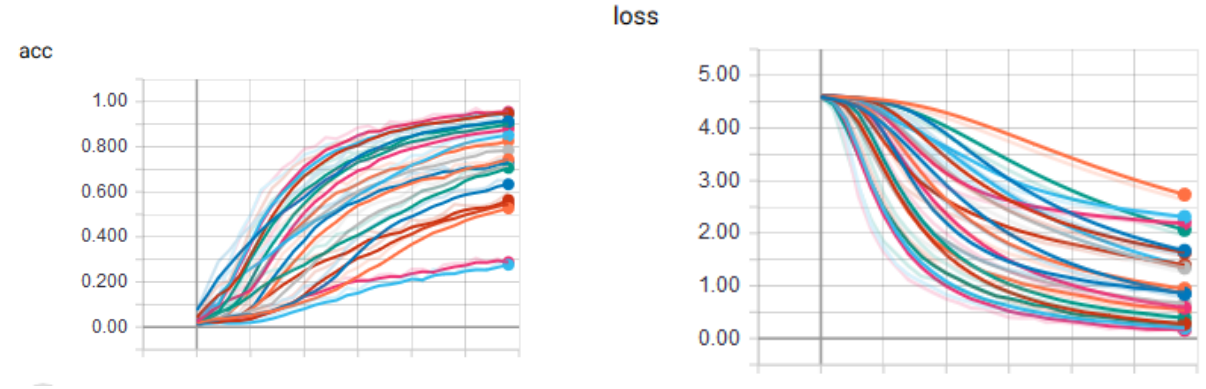
3.1.3 Division des données

Après la fusion, les données doit être divisée en deux tranche, tranche d'apprentissage qui va être utilisée dans l'apprentissage et tranche de test pour tester le modèle après l'apprentissage.

- Le tranche d'apprentissage contient 1299 feature.
- Le tranche de test contient 300 feature.

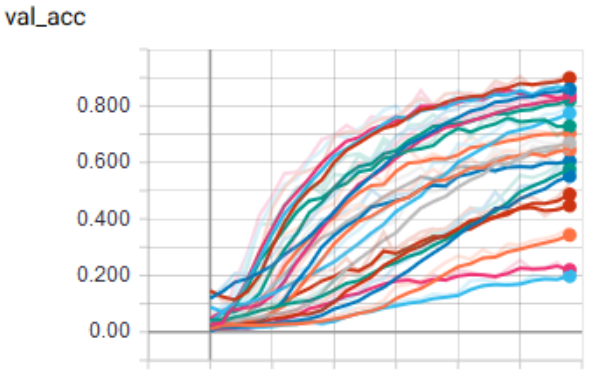
3.2 Apprentissage

Pour l'apprentissage, un script est utilisée pour tester plusieurs des réseaux de neurons pour déterminer la meilleur architecture pour ce problème.
Le script va tester au total 20 réseaux de neurons différente de 1 couche cachée jusqu'à 5 couche cachée et pour chaque couche on teste pour les tailles suivant : 16, 32, 64, 128 neurons.
Après le déroulement du script avec 30 epochs de chaque réseau on utilise tensorboard pour étudier les graphs obtenue qui contient certaine information (val acc, acc ... définie dans 2.4 au dessus) qui va nous aider a choisir la meilleur architecture pour ce problème de classification (figure 2).

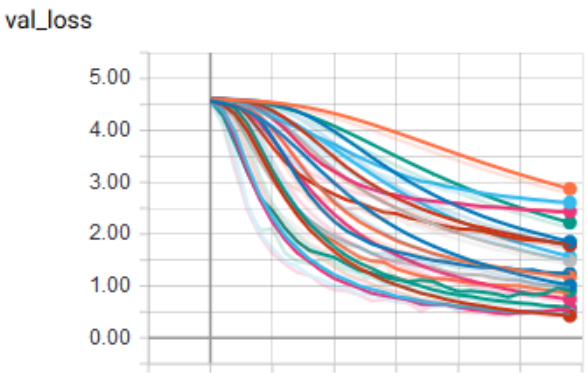


(a) Les courbes de 'accuracy'

(b) Les courbes de 'loss'



(c) Les courbes de 'validation accuracy'

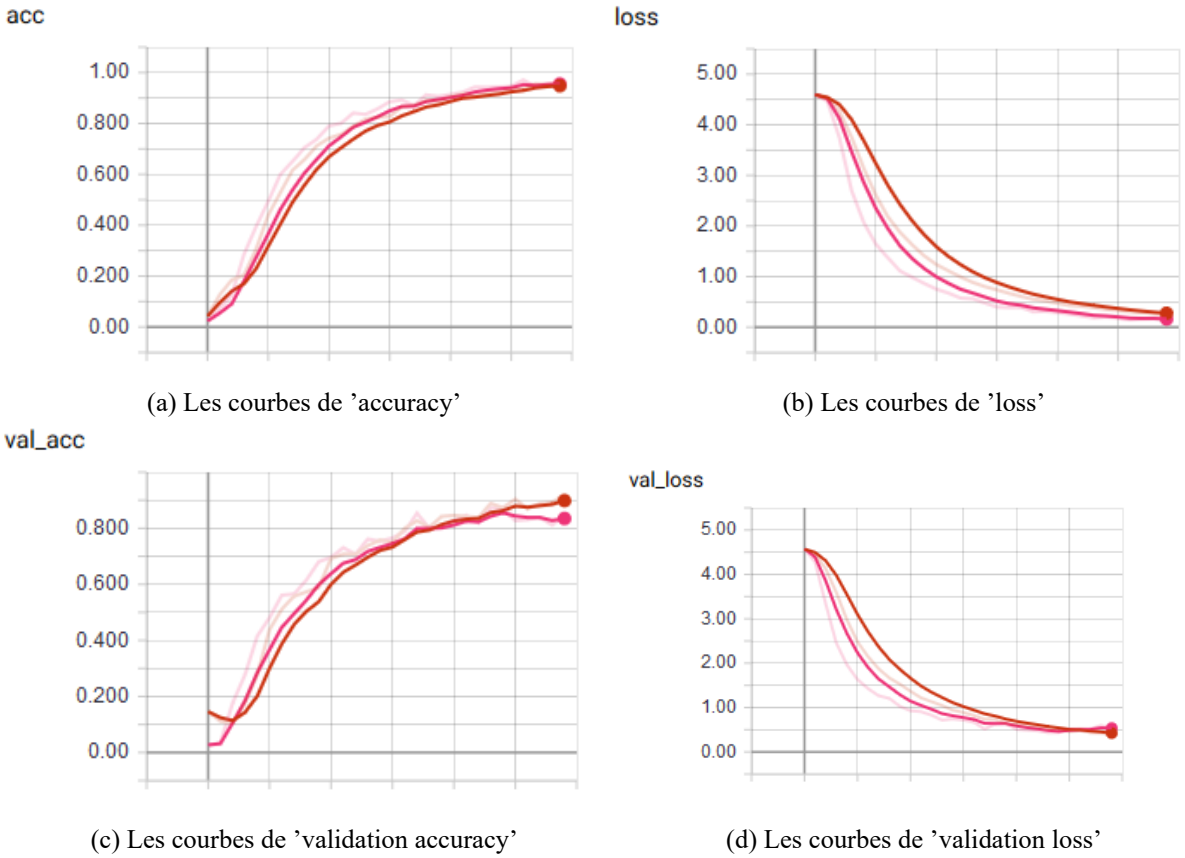


(d) Les courbes de 'validation loss'

FIGURE 2 – Coubres

D'après les graphes les réseaux de neurons avec [2 couche cachée, 128 neurons dans chaque couche] et [4 couche cachée, 128 neurons dans chaque couche] donne les meilleur validation accuracy et accuracy et les plus moins valeurs du loss et comme la différence entre les deux architecture est petite (figure 3) on choisie celle qui a moins de complexité.

On choisie le modèle du réseau de neuron le moins complexe car la différence entre le réseau de 4 couche et 2 couche est très petit donc il est meilleur de gagner l'espace et le performance.



(a) Les courbes de 'accuracy'

(b) Les courbes de 'loss'

(c) Les courbes de 'validation accuracy'

(d) Les courbes de 'validation loss'

FIGURE 3 – Les Courbes des deux Architecture (4 couches et 2 couches avec 128 nœud)

- La meilleur architecture est celle de [2 couche cachée, 128 neurons].

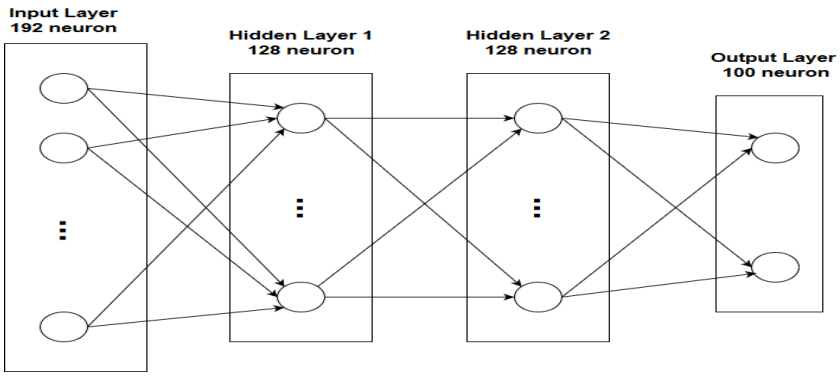
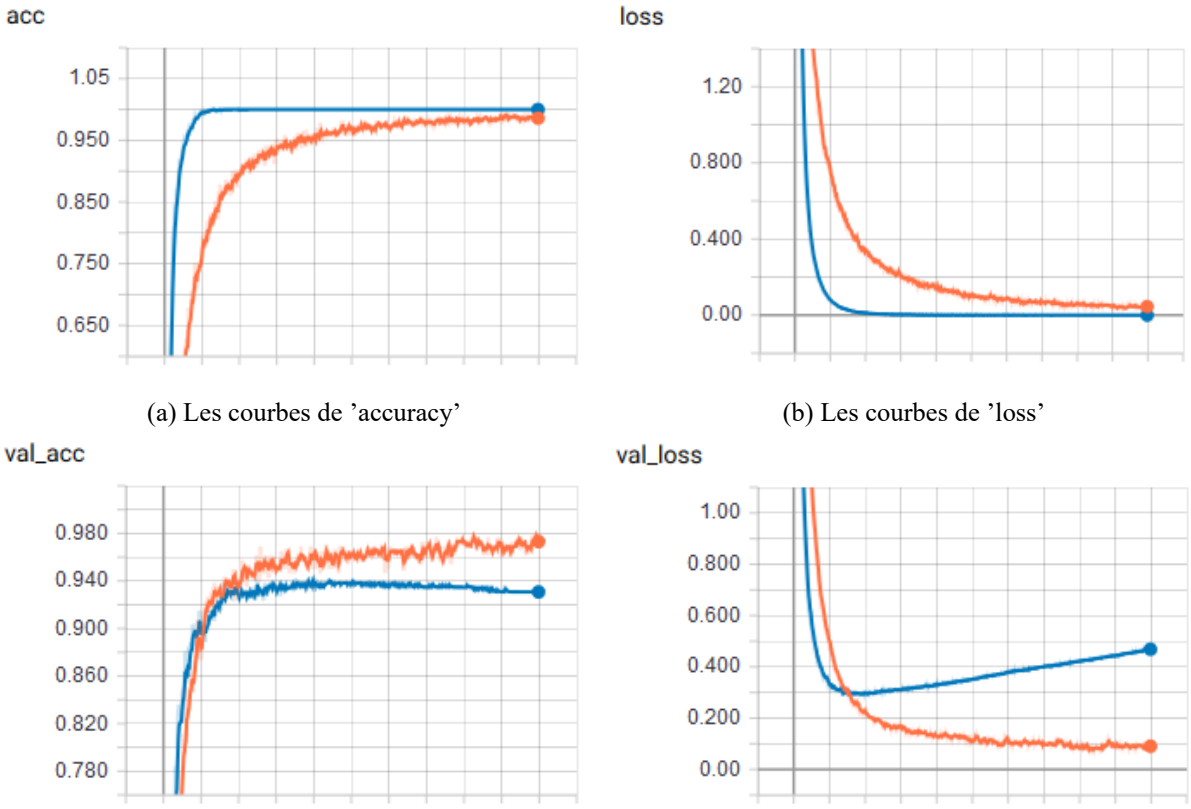


FIGURE 4 – Meilleur reseau de neurons

3.3 Amelioration

Après l'apprentissage de 500 epochs le modèle peut pas depasser 0.92 de 'validation accuracy', Une solution qui peut résoudre ce problème est le dropout, le dropout c'est une méthode qui permet d'éviter l'overfit, quand le dropou est appliquer sur certain couche cachée elle va supprimer certaine neurons d'une manière aléatoire, le but de cette suppression est diminuer la complexité d'un réseau de neurons car quand un modèle est complexe ça veut dire qu'il existe beaucoup de relation et beaucoup de calcule qui va risquer que notre modèle peut apprendre des choses qui nous concerne pas notre problème c'est a dire un bruit, le dropout va diminuer ce bruit et diminuer la chance d'overfit.[6]



(a) Les courbes de 'accuracy'

(b) Les courbes de 'loss'

(c) Les courbes de 'validation accuracy'

(d) Les courbes de 'validation loss'

FIGURE 5 – Les Courbes des reseaux de neuron avec dropout et sans dropout

La courbe orange c'est celle du réseaux de neuron avec dropout et La courbe bleu c'est le réseau sans dropout on peut clairement voir que le réseau de neuron avec dropout donne des meilleur résultat en terme de 'validation accuracy' et de 'loss'(erreur) le réseau avec dropout est plus bon que le réseau sans dropout.

3.4 Test

Après l'apprentissage on teste notre réseau de neuron avec les données de test qui contient 300 'features', après le test on trouve 0.96 de précision.
Le calcule de précision se fait comme suit :

Précision = (nombre de classification juste) / (nombre total de 'features').

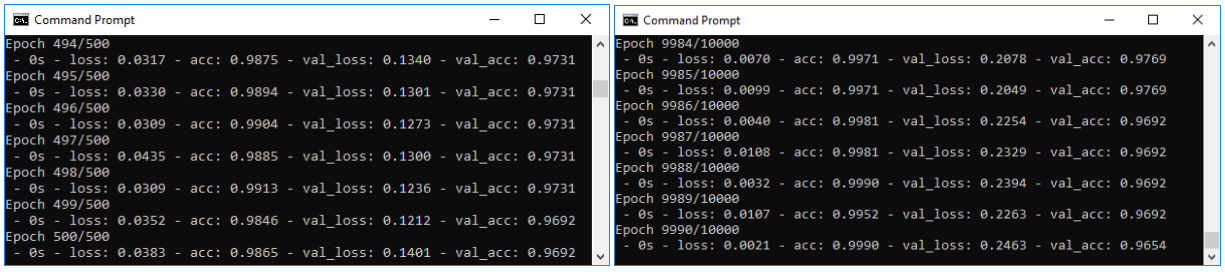
```
Wrong Classification
Classification :  Alnus Cordata , True Label :  Acer Campestre
Classification :  Tilia Tomentosa , True Label :  Acer Rufinerve
Classification :  Quercus Palustris , True Label :  Acer Saccharinum
Classification :  Cornus Controversa , True Label :  Cornus Macrophylla
Classification :  Quercus Vulcanica , True Label :  Morus Nigra
Classification :  Quercus Phillyraeoides , True Label :  Populus Adenopoda
Classification :  Quercus Semecarpifolia , True Label :  Populus Grandidentata
Classification :  Populus Grandidentata , True Label :  Populus Nigra
Classification :  Populus Nigra , True Label :  Quercus Alnifolia
Classification :  Quercus Greggii , True Label :  Quercus Castaneifolia
Classification :  Cornus Chinensis , True Label :  Quercus Rhysophylla
Accuracy :  0.9633333333333334
```

FIGURE 6 – Accuracy et les mal classification

3.5 Quelque Information Sur L'apprentisssage

- L'apprentissage est fait avec un GPU NVIDIA GTX 1060.
- La fonction d'apprentissage utilisee est ADAM.
- La fonction d'erreur (loss) est cross-entropy.
- La fonction d'activation est RELU (Rectified Linear Unit).

L'apprentissage final est fait pour 500 epochs, qui donne des bonne resultat ,on a tester un apprentissage de 10000 epochs mais il donne pas de resultat meilleur, avec 10000 epochs on risque d'avoir un overtrain.



(a) Apprentissage de 500 Epochs (b) Apprentissage de 10000 Epochs

FIGURE 7 – Apprentissage

4 Application

Pour l'application graphique c'est un programme très simple qui ouvre un fichier text de 'features' sans ou avec labels avec la button 'Open File'.

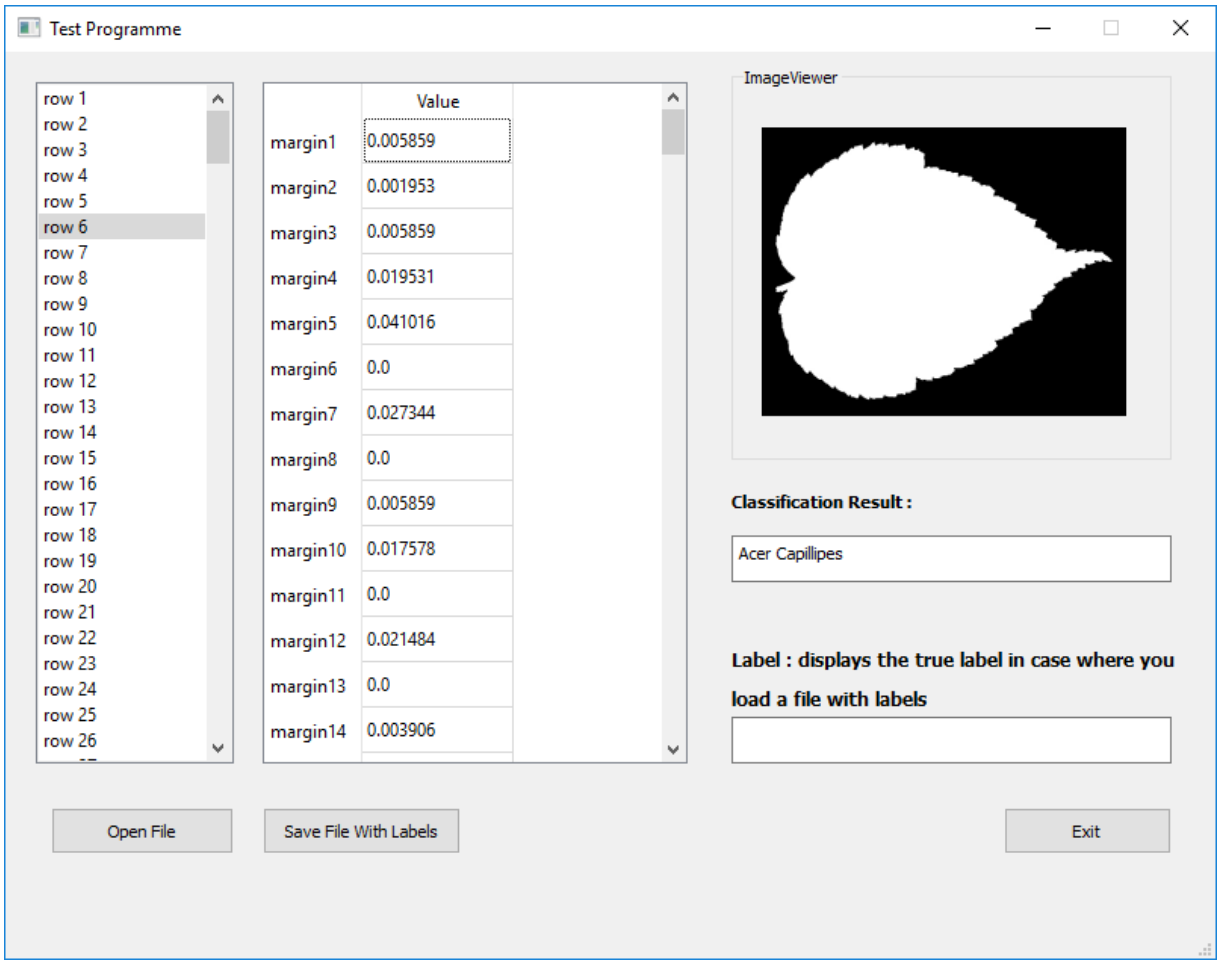


FIGURE 8 – Interface 1

Quand on ouvre un fichier, Le programme va lister tout les lignes de fichier texte et lorsque on sélectionne une ligne (row) le programme va afficher les information de cette ligne (margin, shape , texture) et affiche le résultat de classification pour cette ligne et une image de la feuille de plant.

La button 'Save File With Labels' va produire un fichier qui contient tout les lignes (features) de fichier qu'on a ouvrer avec la classification de chaque ligne (feature).

```
Acer Capillipes,0.001953,0.001953,0.025391,0.017578,
Acer Capillipes,0.001953,0.001953,0.013672,0.011719,
Acer Capillipes,0.005859,0.001953,0.005859,0.019531,
Acer Circinatum,0.0,0.0,0.0,0.011719,0.041016,0.0,0.
Acer Circinatum,0.001953,0.005859,0.005859,0.017578,
Acer Circinatum,0.0,0.0,0.001953,0.017578,0.035156,0
Acer Mono,0.03125,0.029297,0.019531,0.005859,0.0,0.0
Acer Mono,0.017578,0.039062,0.035156,0.029297,0.0039
```

FIGURE 9 – Fichier Generer depuis la button save

Le cas ou on ouvre un fichier qui contient labels et features le programme va afficher le résultat de classification par le réseau de neuron et le label qui est dans le fichier donc on peut voire si le programme fait une bonne classification ou une mal classification.

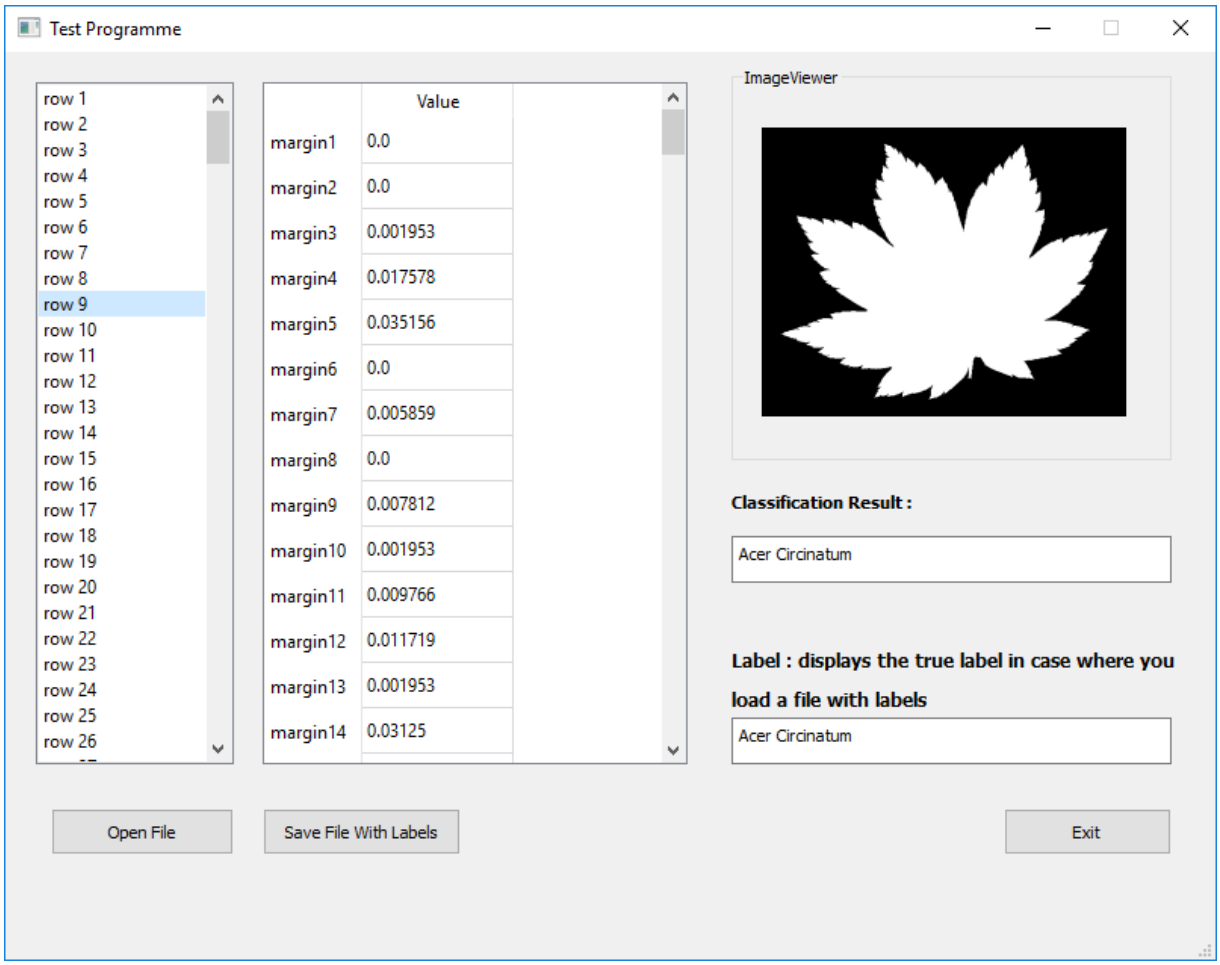


FIGURE 10 – Interface 2

La syntaxe de fichier que le programme peut lit doit être comme suit :

— Chaque ligne : Valeur1, Valeur2, ..., Valeur 192
— Chaque ligne : Label, Valeur1, Valeur2, ..., Valeur 192

Références

[1] https://en.wikipedia.org/wiki/Computer_vision

[2] James Orwell Charles Mallah, James Cope. Plant leaf classi ?cation using probabilistic integration of shape, texture and margin features. Signal Processing, Pattern Recognition and Applications, in press.

[3] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set

[4] https://en.wikipedia.org/wiki/TensorFlow

[5] https://en.wikipedia.org/wiki/Keras

[6] Dropout : A Simple Way to Prevent Neural Networks from Overfitting