

## Résultats et performances

### Traitement des images

Nous avons obtenu lors de notre premier algorithme un résultat de % de réussite. Nous avons obtenu ce résultat avec la même fonction de calcul mais en utilisant un traitement d'images différent.

Lors de la première version, nous transformions les images sans traitement en tableaux dont nous récupérions la valeur moyenne de toutes les cases afin d'obtenir une seule valeur.

```
30 def loadDataset(folder):
31     x_array = []
32     y_array = []
33     listDir = os.listdir(folder)
34     for dir in listDir:
35         if dir[0] != ".":
36             for file in os.listdir(folder + dir):
37                 if(file[0] != "."):
38                     img = Image.open(folder + dir + '/' + file).convert('L')
39                     resize_img = img.resize((50, 50))
40                     img = asarray(resize_img)
41                     img_array = getImageValue(img)
42                     x_array.append(img_array)
43                     y_array.append(dir)
44     return {'samples': x_array, 'labels': y_array}
45
```

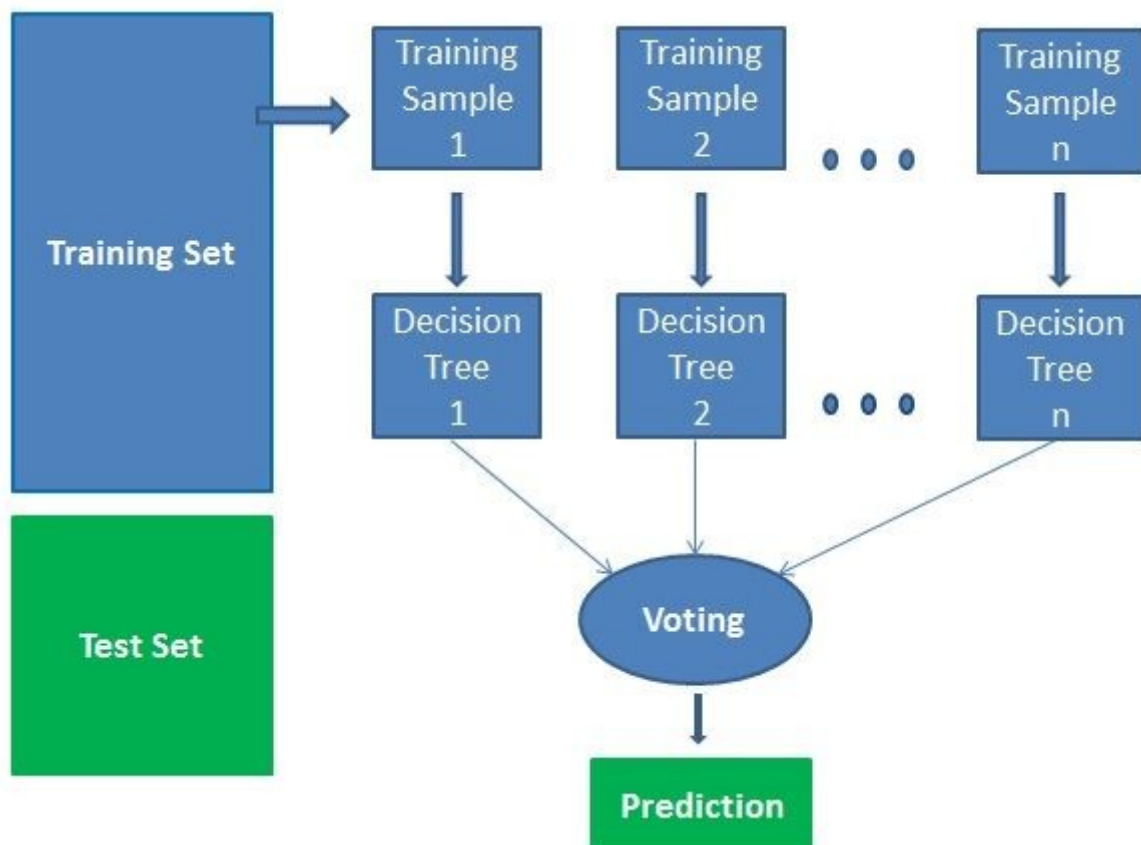
Nous avons donc changé d'approche et pour affiner les résultats, nous avons remplacé le fait d'obtenir une seule valeur par avoir des tableaux où chaque pixel correspond à une nuance de gris. Cette fonction nous a permis d'obtenir des valeurs beaucoup plus précises pour un traitement par notre algorithme car il disposait de plus de données pour faire ses prédictions.

Une amélioration possible aurait été de transformer nos images en trois tableaux correspondant aux taux de rouge vert et bleu, afin d'accroître la précision du traitement. Mais pour des images de scanners principalement en noir et blanc, cette amélioration ne présentait que peu d'intérêt.

## Algorithme

Nous avons obtenu un résultat de 80 % de précision en utilisant la fonction RandomForestClassifier de Sklearn.

Cet algorithme permet de créer une forêt d'arbres décisionnels à partir d'un dataset de test afin d'obtenir une série de prédiction. Ces prédictions nous permettront par la suite d'analyser d'autre set de données d'ont le résultat sera obtenu par un vote entre les résultats des arbres.



Cette technique est réputée assez précise car elle implique un nombre important d'arbres dans la décision, mais se révèle lente sur les gros datasets car il faut parcourir tous les arbres pour chaque décision.

## Réseau de neurones

Une autre approche que nous avons étudié est le réseau de neurones convolutif avec la librairie Keras.

Cette méthode applique une succession de filtres pour traiter les images en entrée et en ressortir une prédiction. Bien que très puissante, le réseau de neurones nous paraissait un peu trop dur pour une entrée en matière de l'IA.

