)

Présentation du DataSet

Introduction

La reconnaissance optique de caractères (OCR) est une technique permettant de transformer le texte présent sur une photographie en fichier texte.

Le système OCR utilise les plus récentes technologies pour collecter les informations d'un document (texte, photographie) scanné et le convertit ensuite en un fichier texte. Pour cela, le système OCR compare les couleurs noires et blanches d'un document pour déterminer chaque code alphanumérique. Le système reconnaît ensuite chaque caractère, et le convertit en texte ASCII (Code américain normalisé pour l'échange d'information).

Exploration des données

La banque d’image utilisée est la IAM Handwriting Database. Réalisée avec la participations de 657 personnes ayant rédigés des fichiers manuscrits:

- 1 539 pages de texte scannées
- 5 685 phrases isolées et labellisées
- 13 353 lignes de texte isolées et labellisées
- 115 320 mots isolés et labellisés

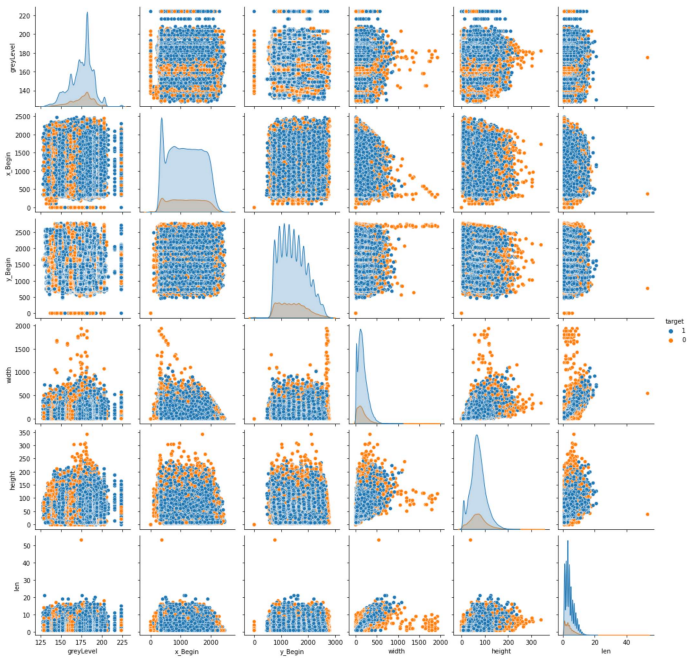
Signification des variables :

```
FileName:     Nom du fichier png
greyLevel:    Nuance de gris de l'image ( 0 noir -> 255 blanc)
target:       Détection du mot ( 0 si non, 1 si Oui par l’algorithme source)
X_Begin:      Position départ abscisse
Y_Begin:      Position départ ordonnée
width:        Largeur de l'image
Height:       Hauteur de l'image
Tag:          Nature du mot
Word:         Mot identifié
Path:         Chemin d'accès à l'image
```

Affichage des Pairplot

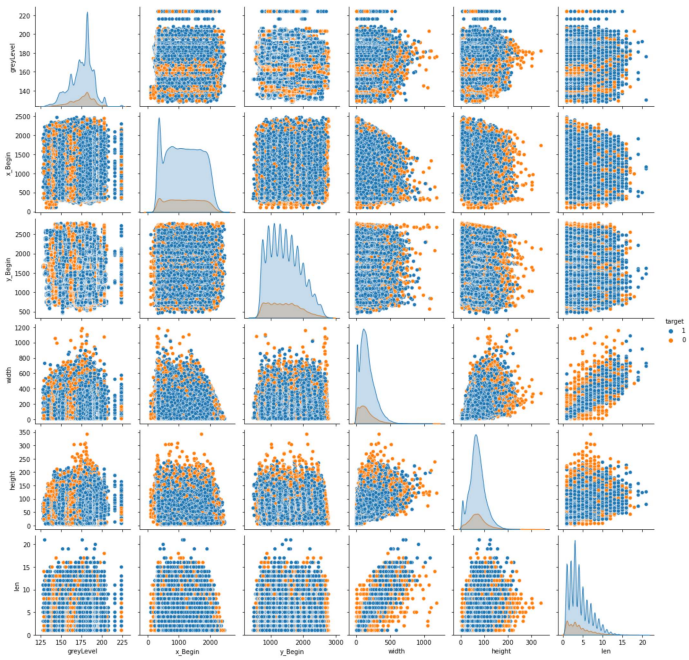
On constate sur le pairplot la présence de point trés éloignés pour les graphiques de mise en relation des variables largeur du mot et nombre de lettre Pour cela, nous avons décidé de séparer le DataSet en 2 groupes:

Avec les valeurs abhérentes

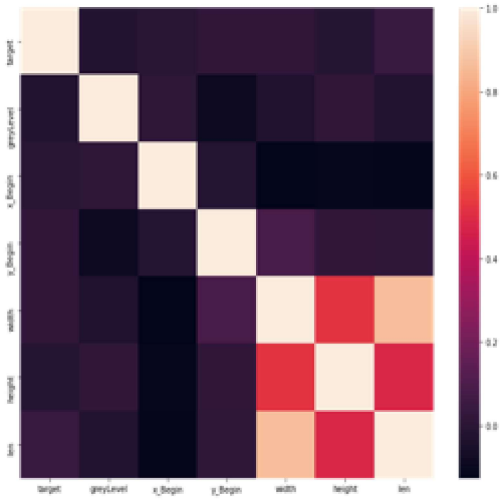


- Largeur image > 1200 px
- Longueur mot > 30 lettres

Sans les valeurs abhérentes



Matrice de correlation



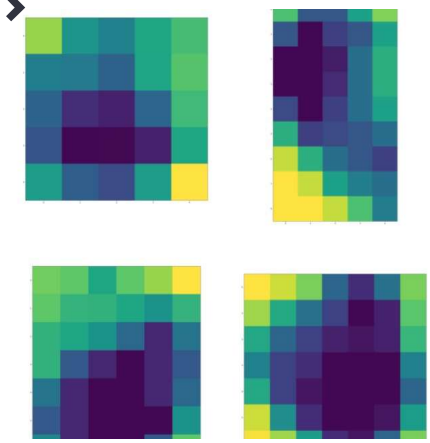
L'analyse des corrélations entre les varaibles ne nous indice pas grand chose mise à part un forte corrélation entre le nombre de caractère d'un mot et la largeur de l'image (assez attendu) Il semble également y avoir une corrélation entre largeur et hauteur mais pas aussi forte que ce que l'on pourrait intuiter Il semblerait que plus le mot contient de caractères plus celui-ci aura une image grande. Cela est du à la plus grande probabilité d'avoir une lettre 'grande' dans un mot si celui-ci est long (ex : L/ F /G)

Affichage des images non reconnues

Image trop large

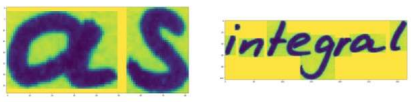
On peut observer que la présence de ponctuation d'apostrophe ou de point sur des I pourrait avoir une incidence sur la dectection et qui explique parfois les hauteurs plus grandes des mots

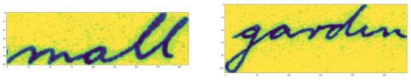
Image trop petite



Même après un bon découpage de la ponctuation, celle-ci ne semble pas être correctement reconnue. Bien qu'il semble y avoir plusieurs cas d'exemple dans le dataset, le modéle ne semble pas avoir appris à les reconnaitre

Niveau de gris posant problème





On peut voir sur cette échantillion du bruit sur les images ce qui a probablement empêché la detection On peut suppose qu'en améliorant le traitement de l'image on pourrait

augmenter le taux de reconnaissance

Conclusion de l’analyse des données

Cette première analyse du Data Set nous ouvre un grand nombre de voies sur la compréhension des méthodes de détection. En effet l'analyse de ces résultats et principalement des KO nous apprennent beaucoup sur les cas de figure problématiques rencontrées par le modèle :

- Difficulté dans le split des phrases en mots
- Bruit de l'image un peu trop fort
- Présence de mots raturés
- Mots tronqués

Cette dernière nous permet aussi de pouvoir émettre quelques hypothèses à valider pour améliorer les performance du modèle:

- Lettre 'accidentogène' voir même en poussant le raisonnement des combinaisons du type 'RS'
- Le ratio w/h pourrait permettre d'identifier des images KO avant traitement - Un traitement du bruit des image pourrait améliorer performance du modèle