

## Projets à réaliser

### Réglementation :

1. Les projets peuvent être réalisés au minimum en binômes et un maximum de cinq étudiants.
2. Les projets peuvent être exposés devant l'ensemble des étudiants.
3. Le choix du projet est laissé libre aux étudiants.
4. Les projets avec les mêmes contenus se verront attribuer un 'zéro'.

### Dates :

La date **limite** de remise des projets est prévue pour le vendredi 10/12/2021.

### A remettre :

1. Un rapport format Word détaillant le travail effectué.
2. Un code source en python.
3. Une petite interface d'utilisation.
4. Une présentation format PowerPoint.

### Intitulés :

- I. Filtrage, binarisation
  1. Réaliser une interface qui permet de choisir une suite de filtres pour un bon rendu de l'écriture.
  2. Proposer une technique adaptative interactive de binarisation de l'écriture.
- II. Linéarisation dans le domaine cartésien
  3. Segmentation de l'écriture arabe en parties connexes.
  4. Implémentation de l'algorithme de squelettisation « Zhang & Suen ».
  5. Proposer une technique de linéarisation de l'écriture : S'inspirer des directions de Freeman pour les directions à adopter.
- III. Linéarisation avec une structure d'arbres n-aires
  1. Segmentation de l'écriture arabe en parties connexes.
  2. Implémentation de l'algorithme de squelettisation « Zhang & Suen ».
  3. Représentation des parties connexes dans des structures d'arbres n-aires.
  4. Proposer une technique de linéarisation de l'écriture en utilisant la même structure : S'inspirer des directions de Freeman pour les directions à adopter.
- IV. Orthogonalisation dans le domaine cartésien
  1. Segmentation de l'écriture arabe en parties connexes.
  2. Implémentation de l'algorithme de squelettisation « Zhang & Suen ».

3. Proposer une technique d'orthogonalisation de l'écriture : on prend les deux directions horizontale et verticale.

#### V. Orthogonalisation

1. Segmentation de l'écriture arabe en parties connexes.
2. Implémentation de l'algorithme de squelettisation « Zhang & Suen ».
3. Représentation des parties connexes dans des structures d'arbres n-aires.
4. Proposer une technique d'orthogonalisation de l'écriture en utilisant la même structure : on prend les deux directions horizontale et verticale.

#### VI. Segmentation

1. Proposer une technique de segmentation en parties connexes.
  - a. Appliquée à l'écriture contrainte
  - b. Appliquée à l'écriture non contrainte
2. Séparer les parties principales des diacritiques et proposer une technique de segmentation en graphèmes avec affectation des diacritiques aux graphèmes appropriés.

#### VII. Extraction de caractéristiques

1. Implémenter trois méthodes d'extraction de caractéristiques.
2. Implémenter une méta-heuristique de votre choix pour sélectionner les caractéristiques de chaque méthode, puis appliquer la méta-heuristique pour les trois ensembles résultants.
3. Valider la méthode de sélection avec un classifieur de votre choix.

#### VIII. Extraction de caractéristiques

1. Implémenter trois méthodes d'extraction de caractéristiques.
2. Implémenter une méta-heuristique de votre choix pour sélectionner les caractéristiques de l'ensemble des trois méthodes, combinées de façon ordonnée, puis de façon aléatoire.
3. Valider la méthode de sélection avec un classifieur de votre choix.

#### IX. Combinaison de classifieurs

1. Choisir une méthode d'extraction de caractéristiques à implémenter.
2. Choisir trois classifieurs à utiliser.
3. Proposer une technique de combinaison de ces classifieurs.

#### X. Deep learning

1. Proposer une architecture de DL pour la reconnaissance du manuscrit arabe traitant une image de mot entier.
2. Étudier l'effet de quelques techniques d'augmentation des données

#### XI. Utilisation de Tesseract

1. Installation de Tesseract et Pythesseract
2. Configuration pour travailler avec la langue arabe

3. Utilisation de Python-OpenCV-Pytesseract pour reconnaître des images avec écriture arabe : imprimée, manuscrite contrainte, manuscrite non contrainte ?

## **XII. Utilisation de EasyOCR**

1. Installation de EasyOCR
2. Configuration pour travailler avec la langue arabe
3. Utilisation de Python-OpenCV- EasyOCR pour reconnaître des images avec écriture arabe : imprimée, manuscrite contrainte, manuscrite non contrainte ?