

# Rapport Extension Mini Projet Machine Learning

**Alexis Araujo**

<b>Introduction</b>	<b>2</b>
<b>DATASET</b>	<b>2</b>
<b>Préparation du Dataset</b>	<b>2</b>
<b>Création du Dataset</b>	<b>2</b>
Prétraitement des Images	2
Conversion en Fichier HDF5	2
Gestion du Processus	3
Chargement des données	3
<b>Apprentissage avec Perceptron</b>	<b>3</b>
Paramétrage et visualisation des données	3
Initialisation des paramètres	3
Modèle de prédiction	3
Calcul de la perte	4
Calcul des gradients	4
Mise à jour des paramètres	4
Entraînement du modèle	4
Évaluation et Visualisation	4
Prédiction d'une donnée	4
<b>Conclusion</b>	<b>4</b>

# Introduction

Ce projet avait pour objectif de construire un modèle de classification entre les tulipes et les roses en se basant sur des images de ces fleurs. J'ai utilisé un algorithme d'apprentissage supervisé : le perceptron, afin de prédire si une image donnée représente une tulipe ou une rose en se basant sur leurs caractéristiques visuelles.

## DATASET

### Préparation du Dataset

Pour commencer j'ai renommé toutes mes images en rose.1.jpg, rose.2.jpg, ... et pareille pour les tulipes. J'ai fait ainsi pour la création des fichiers en hdf5 par la suite. Ce code a été fait dans le fichier renommage.

### Création du Dataset

Pour constituer un ensemble de données exploitable pour l'apprentissage du modèle, une fonction spécifique `images_to_hdf5_dataset` a été mise en place. Cette fonction accomplit plusieurs étapes cruciales :

#### Prétraitement des Images

Les images de tulipes et de roses, stockées dans des dossiers distincts, ont été sélectionnées pour leur transformation en matrices exploitables. Chaque image a été redimensionnée à une taille uniforme de 64x64 pixels pour assurer la cohérence des données.

#### Conversion en Fichier HDF5

La fonction `images_to_hdf5_dataset` charge les images depuis le répertoire spécifié. Elle les transforme ensuite en matrices en niveaux de gris et les redimensionne aux dimensions requises pour les stocker dans un fichier HDF5.

#### Attribution des Étiquettes

Chaque image est associée à une étiquette correspondant à sa catégorie : 0 pour les roses et 1 pour les tulipes. Ces étiquettes sont intégrées dans le fichier HDF5 pour faciliter l'apprentissage supervisé.

## Gestion du Processus

La fonction effectue un suivi du nombre total de fichiers images dans le répertoire, crée des ensembles de données dans le fichier HDF5 pour stocker les images redimensionnées et leurs étiquettes respectives.

## Chargement des données

Une fois les données converties en fichiers HDF5 pour les ensembles d'entraînement et de test, ces données peuvent être facilement chargées et utilisées pour entraîner et tester le modèle de classification entre les tulipes et les roses.

## Apprentissage avec Perceptron

Après avoir chargé et prétraité les données d'entraînement et de test, j'ai utilisé un perceptron pour l'apprentissage. Voici les étapes clés de cet apprentissage :

### Paramétrage et visualisation des données

J'ai réalisé des affichages pour connaître le nombre de données d'entraînement et de test, ainsi que la dimension 64,64 et le 3 qui signifie des données en RVB.

J'ai ajouté une ligne de reshape avec la 1er colonne du tableau pour passer à une classification binaire.

Un affichage avec np.unique pour observer la répartition de la classification des données de 1 et de 0.

Pour visualiser les images j'ai utilisé la librairie matplotlib, en réalisant une boucle sur les images d'entraînement.

Ensuite je normalise les données pour avoir une bonne forme.

### Initialisation des paramètres

Les poids (W) et le biais (b) du modèle ont été initialisés aléatoirement pour démarrer le processus d'apprentissage.

### Modèle de prédiction

Le modèle a été construit pour calculer les prédictions en utilisant une fonction sigmoïde pour déterminer si une image donnée représente une tulipe ou une rose.

## Calcul de la perte

J'ai utilisé la fonction de perte logistique pour évaluer les performances du modèle.

## Calcul des gradients

Les gradients pour les poids et le biais ont été calculés pour ajuster ces paramètres lors de l'apprentissage.

## Mise à jour des paramètres

J'ai mis à jour les poids et le biais en utilisant la descente de gradient afin d'améliorer progressivement la performance du modèle.

## Entraînement du modèle

L'apprentissage s'est déroulé sur un nombre d'itérations fixées à 15000. À chaque itération, le modèle a calculé les prédictions, les gradients, mis à jour les poids et le biais, tout en enregistrant la perte pour évaluer les performances du modèle.

## Évaluation et Visualisation

J'ai évalué les performances du modèle en traçant les courbes de coût et d'exactitude pour les ensembles d'entraînement et de test. Ces courbes ont montré la progression de l'apprentissage et l'ajustement du modèle au fil des itérations.

## Prédiction d'une donnée

J'ai évalué sur une donnée d'image si le modèle fonctionnait. La conclusion est satisfaisante, le modèle arrive à bien prédire si c'est une rose ou une tulipe à partir de données qu'il ne connaît pas encore.

## Conclusion

Ce mini projet m'a permis de mettre en pratique l'apprentissage supervisé, en particulier l'utilisation du Perceptron pour la classification entre les tulipes et les roses en se basant sur des données visuelles. Malgré quelques ajustements, le modèle a

démontré une capacité à distinguer ces deux types de fleurs avec une précision satisfaisante.