

Projet 8 : Déployez un modèle dans le cloud

Lancelot LECLERCQ

10 juin 2022

1. Introduction

2. Architecture Big Data

Introduction

Problématique

- Préservation de la biodiversité des fruits
 - traitements spécifiques pour chaque espèce de fruits par des robots cueilleurs intelligents
- 1ère étape développer une application mobile
 - Permettre à l'utilisateur d'obtenir des informations sur un fruit à partir d'une photo
 - Sensibiliser le grand public à la biodiversité des fruits
 - Mettre en place une première version du moteur de classification des images de fruits
 - Construire une première version de l'architecture Big Data

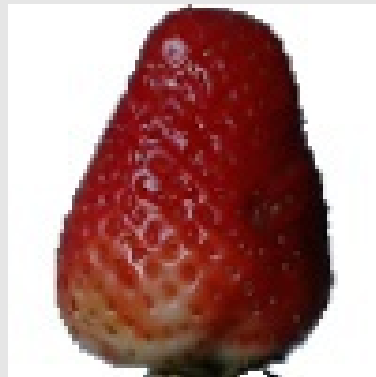


Fruits!

Données

- Utilisation du jeu de données Kaggle :
<https://www.kaggle.com/datasets/moltean/fruits>
- Nombre total d'images : 90483
- Taille du jeu d'entraînement : 67692 images
- Taille du jeu de test : 22688 images
- Nombre de fruits : 131

- Exemple d'image de fruit :



Architecture Big Data

AWS

- Utilisation d'une machine EC2 hébergée par Amazon Web Services
- Utilisation de Debian
- Installation de java, git



Spark

- Parallélisation des calculs
 - Utilisation de plusieurs machines
 - Distribution des calculs
 - Budget
- Utilisation de PySpark
 - Interface Spark en python



PySpark

- Utilisation de PySpark une interface Spark en python

```
ImgData = spark.read.format('binaryFile') \
    .option('pathGlobFilter', '*.jpg') \
    .option('recursiveFileLookup', 'true') \
    .load(path) \
    .select('path', 'content')
```

- Importation des images
au format binaire :

```
udf_image = F.udf(
    get_desc,
    ArrayType(ArrayType(FloatType(), containsNull=False), containsNull=False))

ImgDesc = ImgData.withColumn("descriptors", F.explode(udf_image("content")))
```

- Création d'une colonne
avec les descripteurs
détectés pas OpenCV :

```
VA = VectorAssembler(inputCols=BoVW.drop('label').columns,
    outputCol='features')
pca = PCA(k=100, inputCol='features', outputCol='pca_features')
pipe = Pipeline(stages=[VA, pca])
```

- Réduction de dimension
par PCA :

```
pipePCA = pipe.fit(BoVW)

pcaData = pipePCA.transform(BoVW)
pcaDataDF = pcaData.select(['label', 'pca_features']).toPandas()
```