

Pour info, sur la slide 9 il y a des animations qui cachent le texte en dessous donc si possible plutôt regarder le ppt en mode diaporama pour voir le texte sur cette slide.

# Déployer un modèle dans le cloud



**Fruits!**

# Sommaire

1. Rappel de la problématique
2. Présentation du jeu de données
3. Présentation des étapes de la chaîne de traitement des images
4. Architecture Big Data retenue
5. Démonstration
6. Conclusion



# 1. Rappel de la problématique



**Fruits!**

# Rappel de la problématique

- ▶ « ***Fruits!*** » : jeune start-up de l'AgriTech qui cherche à proposer des solutions innovantes pour la récolte des fruits.
- ▶ Volonté dans un premier temps de se faire connaître en mettant à disposition du grand public une application mobile qui permettrait aux utilisateurs de ***prendre en photo un fruit et d'obtenir des informations sur ce fruit.***
- ▶ ***Objectif*** : développer dans un environnement Big Data une première chaîne de traitement des données qui comprendra le preprocessing et une étape de réduction de dimension.



## 2. Présentation du jeu de données



**Fruits!**

# Présentation du jeu de données

- ▶ Jeu de données constitué d'images de fruits avec les labels associés :
  - ❖ 131 variétés de fruits contenant des images de chaque angle du fruit



Apple Golden 1



Banana



Strawberry

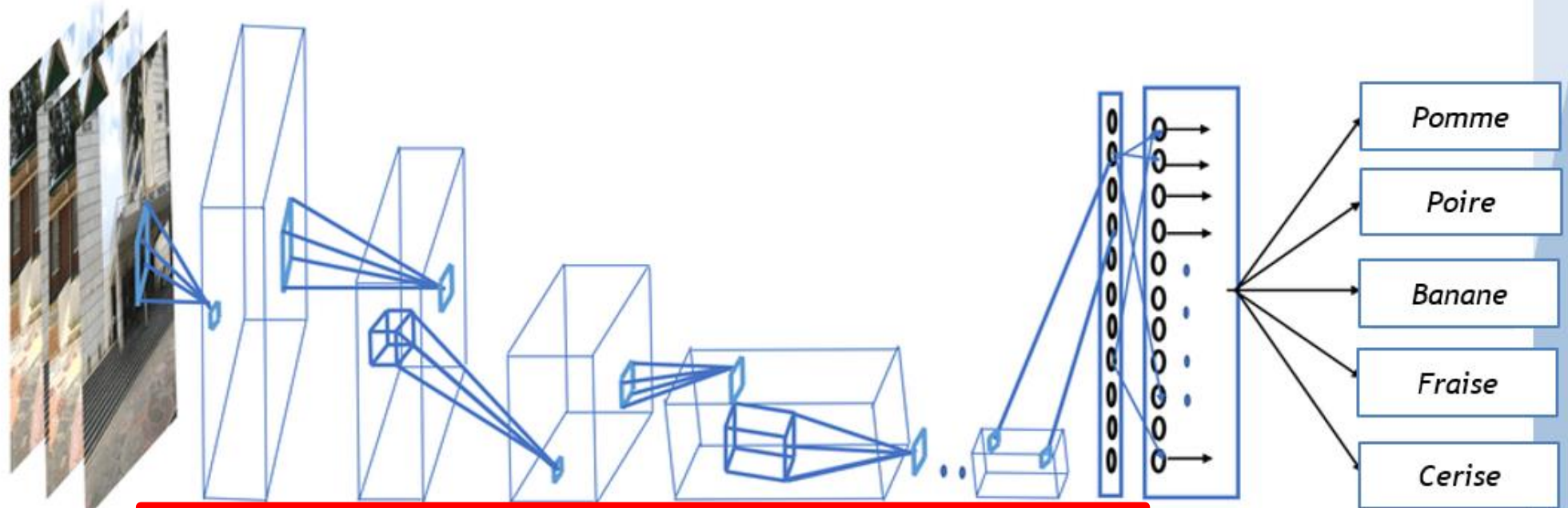


### 3. Présentation des étapes de la chaîne de traitement des images



**Fruits!**





Input

Convolution  
+  
ReLu

Pooling

Convolution  
+  
ReLu

Pooling

Flatten

SVM /  
Logistic  
Regression /  
Random  
Forest

Pomme

Poire

Banane

Fraise

Cerise

Redimensionnement  
des images en 224x224  
et utilisation du  
module de  
preprocessing du  
modèle ResNet50

Extraction des features en utilisant un modèle  
pré-entraîné (Resnet50)

Classification



Fruits!

# Présentation des étapes de la chaîne de traitement des images

- ▶ **Etape de réduction de dimension** : utilisation PCA (Principal Component Analysis) à deux composantes principales (nombre de composantes pouvant être ajusté).
- ▶ Projection des features de chaque image dans le nouvel espace formé par les deux composantes principales.
- ▶ Obtention d'un Dataframe récapitulant le lien vers l'image ainsi que la réduction de dimension des features dans le nouvel espace à deux dimensions.



## 4. Architecture Big Data retenue



**Fruits!**

# Architecture Big Data retenue

- ▶ **Objectif** : développer la chaîne de traitement de manière à pouvoir passer à l'échelle rapidement et facilement.
- ▶ *Données massives* : nécessité de distribuer les opérations de traitement sur plusieurs machines.
- ▶ *Framework Spark* : outil qui permet de gérer et de coordonner l'exécution de tâches sur des données à travers un groupe de machines.
- ▶ Utilisation de PySpark qui est l'implémentation de Spark pour le langage Python.



# Architecture Big Data retenue

- ▶ Afin que Spark puisse distribuer les calculs, nécessité d'avoir un cluster de machines à disposition.
- ▶ Utilisation d'une plateforme cloud :
  - ❖ Permet de louer des ressources matérielles (capacité de calcul et de stockage) pour une durée déterminée.
  - ❖ Permet la notion d'élasticité de l'architecture → possibilité d'agrandir ou de diminuer la capacité de l'architecture pour des durée variables.
- ▶ Choix d'utiliser la plateforme cloud d'Amazon : Amazon Web Services (AWS)

# Architecture Big Data retenue

## ► Solutions utilisées sur AWS :

- ❖ Simple Storage Service (S3)
- ❖ Elastic Map Reduce (EMR) et Elastic Cloud Compute (EC2)

## ► S3 : espace de stockage de données.

- ❖ Stockage des images de fruits.
- ❖ Stockage de la sortie de l'application PySpark contenant au format JSON les liens vers les images ainsi que la réduction de dimension de chaque image.

## ► EMR : lancement de clusters de calcul composés de machines/instances EC2 qui sont paramétrables selon les besoins.

- ❖ Création d'un cluster composé de 3 instances EC2 Linux de type 'm5.xlarge' (4 Cœurs virtuels, 16 Go de mémoire).



## 5. Démonstration



**Fruits!**

# Conclusion

- ▶ Développement d'une chaîne de traitement des images en PySpark comprenant un preprocessing et une réduction de dimension des features obtenues grâce au modèle pré-entraîné ResNet50.
- ▶ Utilisation de la plateforme cloud Amazon Web Services (AWS).
- ▶ Utilisation des solutions AWS S3 pour stocker les données et la sortie de l'application et d'EMR pour déployer un cluster de calculs distribués utilisant des instances EC2 Linux.
- ▶ L'architecture Big Data sur le cloud ainsi créée permettra d'avoir une très bonne flexibilité lorsqu'il faudra passer à l'échelle en termes de volume de données.

