Déployez un modèle dans le cloud

Fruits!

Sommaire

Préambule

"**Fruits!" -** Pionnier de l'AgriTech

Contexte :
Entreprise : Start-up "Fruits!"
Secteur : AgriTech, focalisé sur la biodiversité des fruits
Innovation : Robots cueilleurs pour une récolte adaptée à chaque espèce de fruits

Application Mobile :
Fonction : Photo de fruits pour identifier et informer
Technologie : Classification avancée d'images

Enjeu Technologique :
Appropriation et poursuite des travaux préexistants
Construction d'une architecture Big Data sur AWS EMR

Présentation des données

Données issues d'un kernel Kaggle :

- 90380 images et 131 sous répertoires qui contiennent des images de fruits ou de légumes

- 2 jeux de données training (67692) et test (22688)

Un dossier contenant les données réduites :

- 12455 images et 24 classes

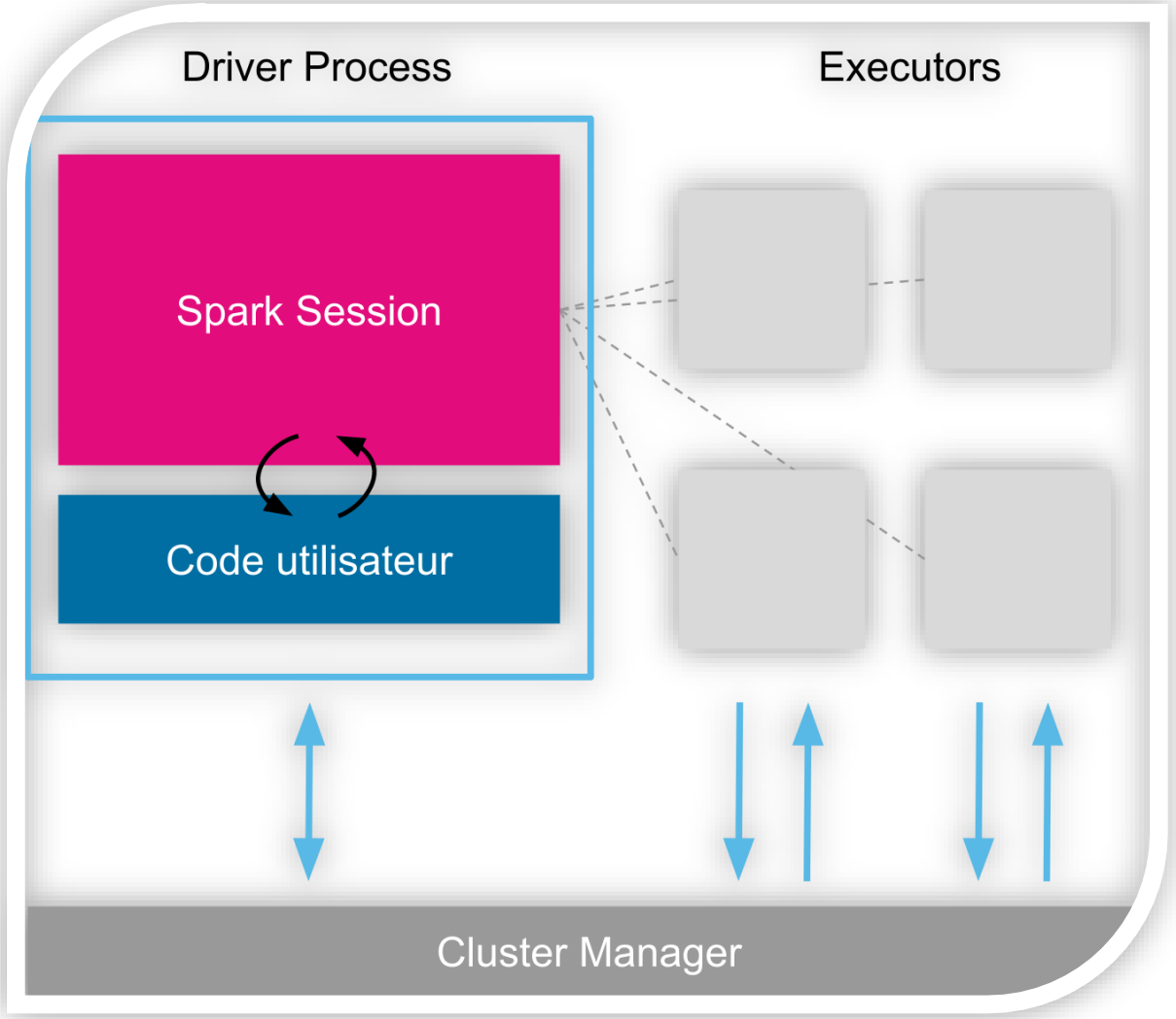
- 3 jeux de données training (6231), test1 (3110) et validation (3114)

Première phase : Mise en Place Locale

Première étape : Choix Techniques

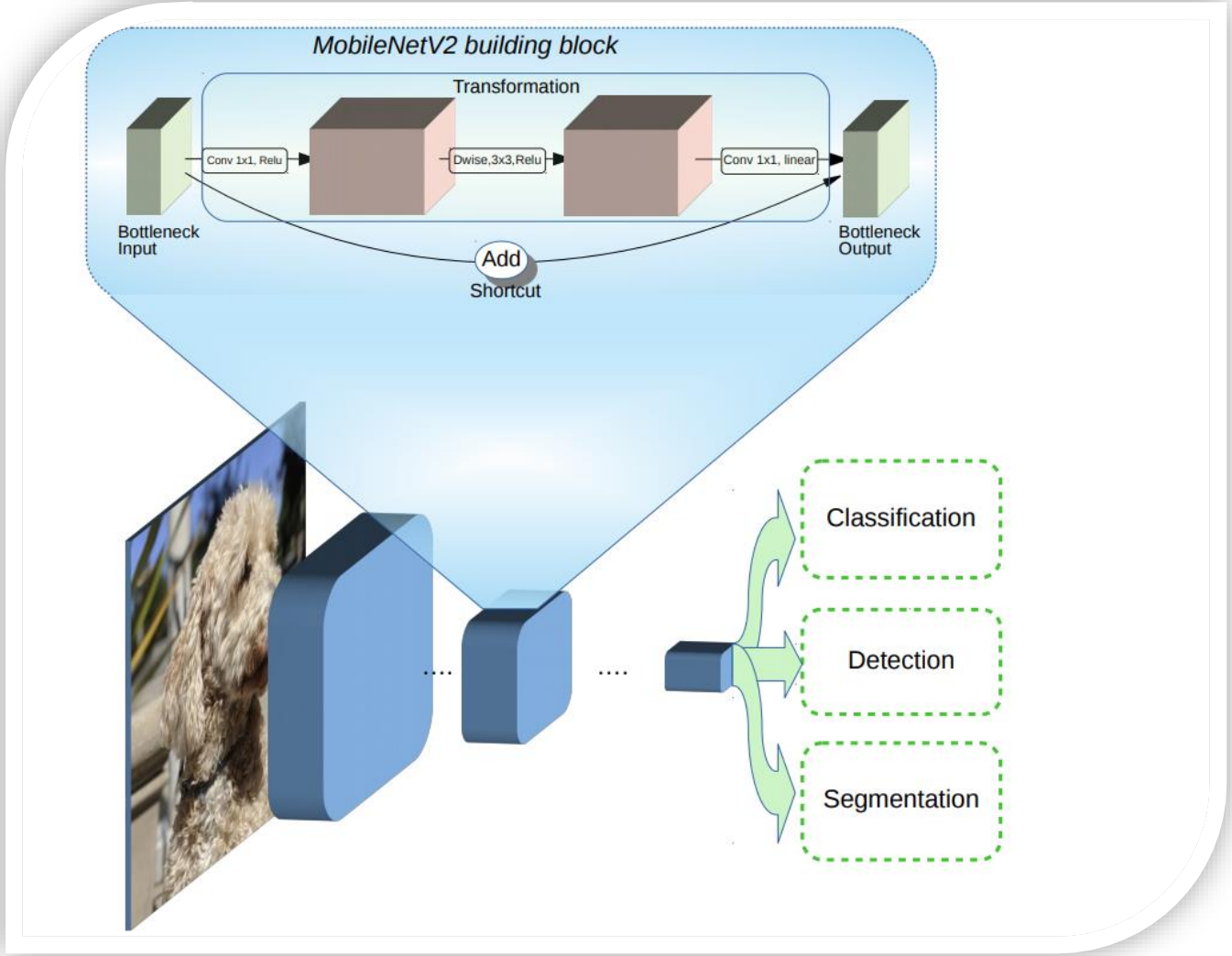
Calcul Distribué avec PySpark

- Objectif: Gérer l'augmentation rapide du volume de données
- Qu'est-ce que PySpark? Outil pour communiquer avec Spark en Python, idéal pour le traitement de données volumineuses en utilisant le calcul distribué
- Utilisation: En local pour simuler le calcul distribué; sur le cloud pour une exécution réelle sur un cluster



Transfert Learning avec MobileNetV2

- Objectif: Chaîne de traitement incluant le preprocessing et une réduction de dimension
- Pourquoi MobileNetV2? Pour sa légèreté, sa rapidité d'exécution et faible dimensionnalité de son vecteur de sortie
- Méthode: Utiliser la connaissance du modèle pré-entraîné, récupérer l'avant-dernière couche, produisant un vecteur de dimension (1,1,1280) pour la classification des fruits



Deuxième étape : installer l'environnement de travail Spark

Les étapes de la mise en place locale

Environnement:

- ❖ Linux Ubuntu sur une machine virtuelle (Windows)
- ❖ Installation: Spark, Python packages (Pandas, TensorFlow, PySpark, Pillow, PyArrow)

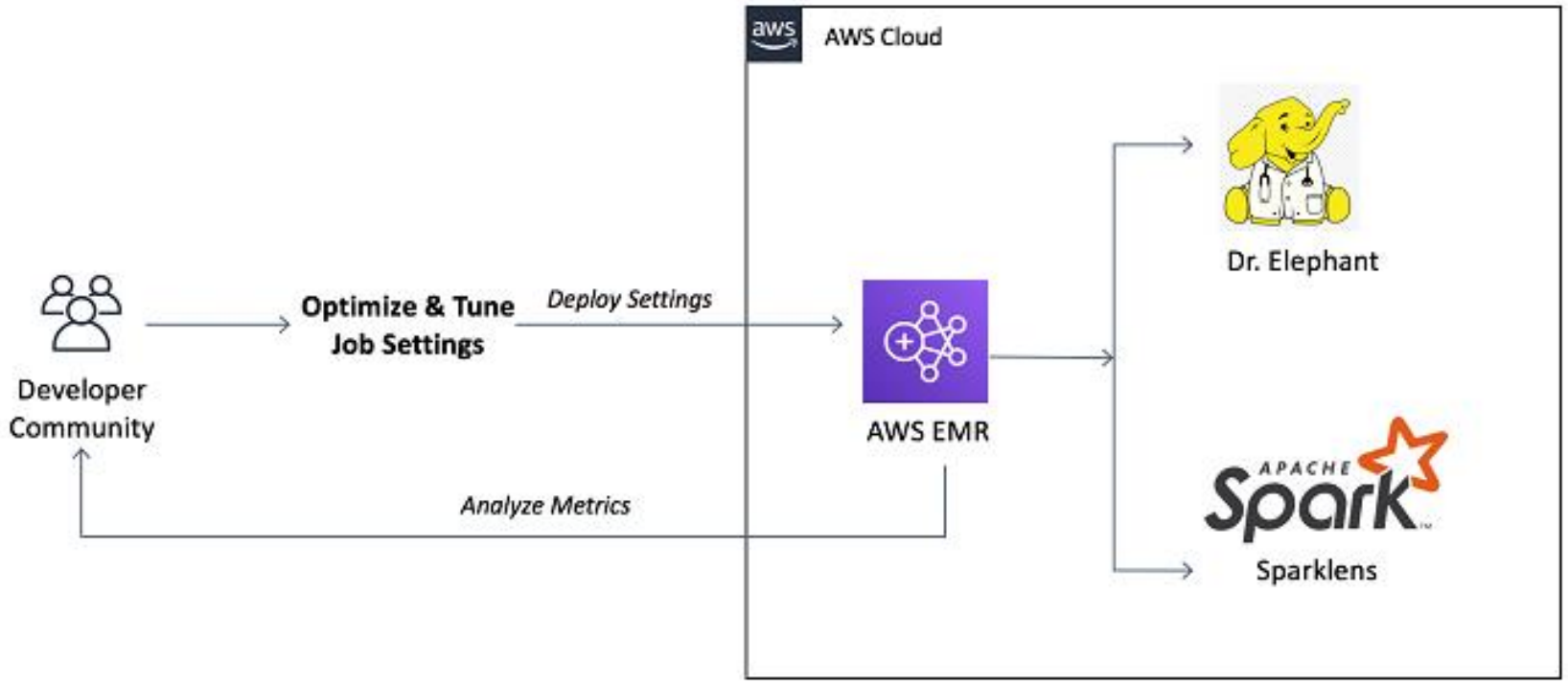
Chargement et Préparation des Données :

- ❖ Les images de 'Test1'
- ❖ Traitement avec MobileNetV2, featurisation via Pandas UDF

Résultats:

- ❖ Sauvegardés dans un fichier local "Results"
- ❖ Vérification de la dimension des vecteurs

Deuxième phase : Créer un réel cluster de calculs



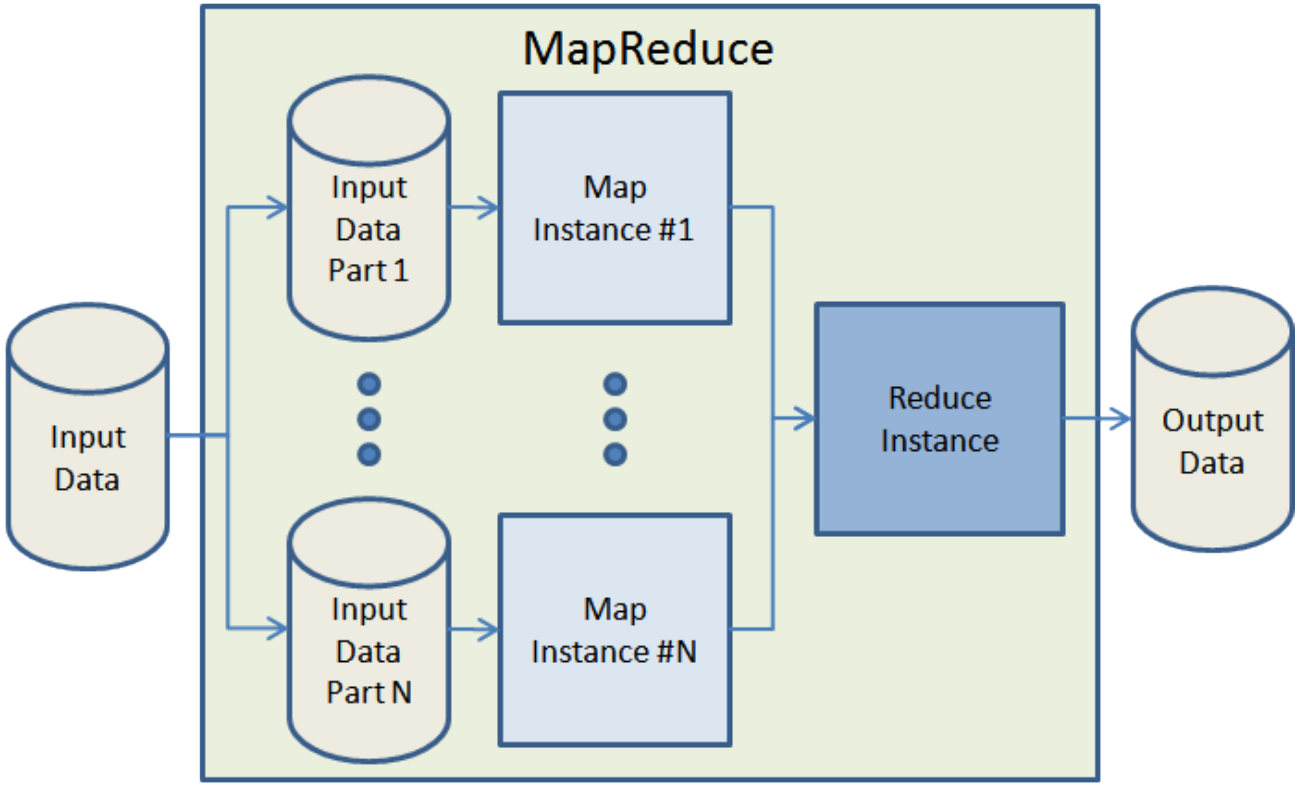
Première étape : Déploiement sur le Cloud

Prestataire Cloud

- Amazon Web Services (AWS)

Solution Technique

- Elastic MapReduce (EMR) pour un calcul distribué



Stockage

❖ Données stockées sur Amazon Simple Storage Service (S3)

Configuration & Sécurité

- Configuration personnalisée de l'EMR pour répondre aux besoins
- Sécurité garantie par des tunnels SSH et des politiques IAM

Exécution & Suivi

- ❖ Scripts PySpark déployés et exécutés
- ❖ Monitorage avec le Serveur d'Historique Spark pour traquer l'avancement

Deuxième étape : Gestion du Serveur

Les étapes de gestion

Résiliation de l'Instance EMR:

- ❖ Après l'achèvement des tâches, l'instance EMR est résiliée pour éviter des coûts inutiles
- ❖ Automatisation possible pour garantir une fermeture efficace

Clonage de l'Instance EMR:

- ❖ Possibilité de dupliquer l'instance pour des besoins futurs ou de sauvegarde
- ❖ Garantit une mise en place rapide avec les mêmes configurations

Structure Finale du Serveur S3:

- ❖ Organisé en dossiers et sous-dossiers pour une navigation aisée
- ❖ Données traitées et non traitées clairement séparées pour faciliter les mises à jour

Conclusion

Au terme de ce projet, nous avons:

- Établi un Processus Efficace : À travers un déploiement local, suivi d'une mise en œuvre sur le cloud, nous avons mis en place un pipeline robuste et scalable de traitement de données

- Exploité les Avancées Technologiques : En utilisant PySpark pour le calcul distribué et le Transfert Learning pour la featurisation des images, nous nous sommes assurés d'une analyse performante et précise

- Optimisé la Gestion des Coûts et des Ressources : Avec une gestion prudente des instances EMR et une structure organisée du serveur S3, nous avons garanti une utilisation efficace des ressources

Perspectives:

- Évolution du Modèle : L'intégration d'autres modèles pré-entraînés pourrait apporter une précision accrue

- Automatisation Améliorée : Automatiser davantage de processus, notamment la gestion d'instances, pour réduire l'intervention manuelle

- Expansion vers D'autres Clouds : Explorer les services offerts par d'autres fournisseurs pour comparer les performances et les coûts

Quelques liens qui ont servis à la réalisation de ce projet

https://vegastack.com/tutorials/how-to-install-anaconda-on-ubuntu-22-04/

https://www.objetconnecte.com/spark-presentation

https://datascientest.com/pyspark

https://www.veonum.com/apache-spark-pour-les-nuls/

https://repost.aws/fr/knowledge-center/ec2-instance-access-s3-bucket#

https://learn.microsoft.com/en-us/azure/databricks/_static/notebooks/deep-learning/keras-metadata.html