

# TP Intelligence Artificielle

## Reconnaissance d'images

Avril 2025

# Sommaire

1 Répartition du travail et organisation

2 Type d'apprentissage et algorithmes utilisés  
par TensorFlow

3 Comment une machine reconnaît-elle une  
image?

4 Présentation de la solution

5 Contraintes et réussites

6 Questions / Réponses

# Répartition du travail et organisation

# Notre équipe



**Antoine**

Configuration MongoDB Atlas  
DevOps  
Documentation



**Charlotte**

Interface utilisateur React  
Intégration TensorFlow.js



**Said**

Architecture backend  
API REST  
Modèle de données

# Organisation du travail

## Outils utilisés



### GitHub

Versionnement, Pull Requests, Issues



### Discord

Communication en temps réel



### Trello

Suivi des tâches, méthodologie Agile

## Méthodologie



### Sprints hebdomadaires

Organisation Agile avec des cycles courts



### Code Review

Validation par les pairs avant merge



### Tests

Tests unitaires et d'intégration

# Type d'apprentissage et algorithmes utilisés par TensorFlow

# Types d'apprentissage pris en charge

## Apprentissage supervisé

Données étiquetées: classification, régression

## Apprentissage non supervisé

Patterns sans étiquettes: clustering

## Apprentissage par renforcement

Apprentissage par récompenses/pénalités

## Apprentissage par transfert

Réutilisation de modèles pré-entraînés

# Algorithmes et modèles principaux

## Réseaux de neurones profonds

- MLP (Multi-Layer Perceptron)
- CNN (Convolutional Neural Networks)
- RNN, LSTM, GRU (Recurrent Neural Networks)

## Transformers

- Mécanismes d'attention
- Encodeurs-décodeurs
- BERT, GPT (NLP)
- ViT (Vision Transformer)

## AutoEncoders

- VAE (Variational AutoEncoders)
- Encodeurs pour la réduction de dimensionnalité
- Débruitage et reconstruction

## GAN

- Génération d'images
- Synthèse de données
- StyleGAN, CycleGAN

## Algorithmes de gradient

- SGD (Stochastic Gradient Descent)
- Adam, RMSProp
- Adagrad, Adadelta

## Modèles spécifiques

- MobileNet (pour appareils mobiles)
- EfficientNet (scaling optimal)
- ResNet (réseaux résiduels)

Comment une machine  
reconnait-elle une image ?



# Fonctionnement détaillé

## Couches principales

- **Convolution:** filtres détectant les motifs
- **Pooling:** réduction des dimensions
- **Couches connectées:** classification finale

## Processus d'analyse

- Extraction progressive des caractéristiques
- Hiérarchie de motifs (arêtes → formes → objets)
- Rétropropagation pour l'apprentissage

L'optimisation par gradients permet d'ajuster les poids pour minimiser l'erreur entre la prédiction et la réalité.

# Présentation de la solution

# Notre solution

## Frontend (React)

Interface responsive,  
TensorFlow.js, MobileNet

## Base de données

MongoDB Atlas, schéma flexible  
NoSQL

## Backend (Node.js)

API REST, stockage des analyses

## Fonctionnalités

- ✓ Upload et analyse d'images
- ✓ Classification en temps réel
- ✓ Historique et statistiques

# Démonstration de l'application

# Contraintes et réussites

# Défis rencontrés

## Challenges techniques

- ⚠️ Intégration TensorFlow.js dans React
- ⚠️ Performance et optimisation
- ⚠️ Gestion des formats incompatibles

## Réussites

- ✓ Application fonctionnelle end-to-end
- ✓ Interface intuitive et responsive
- ✓ Classification en temps réel
- ✓ Persistance MongoDB Atlas

## Limitations

- ⚠️ Classification limitée (1000 classes)
- ⚠️ Pas de détection multi-objets

**82%**

Taux de précision  
Sur 200 images testées



## Support Webcam

Analyse d'images en direct depuis la caméra



## Détection multi-objets

Identifier plusieurs éléments dans une même image



## Statistiques avancées

Visualisation des données d'analyse



## Application mobile

Version native pour iOS et Android



## Fine-tuning

Adaptation du modèle à des domaines spécifiques



## API publique

Service de reconnaissance accessible via API

# Questions / Réponses

**Merci pour votre attention !**

Des questions ?

END