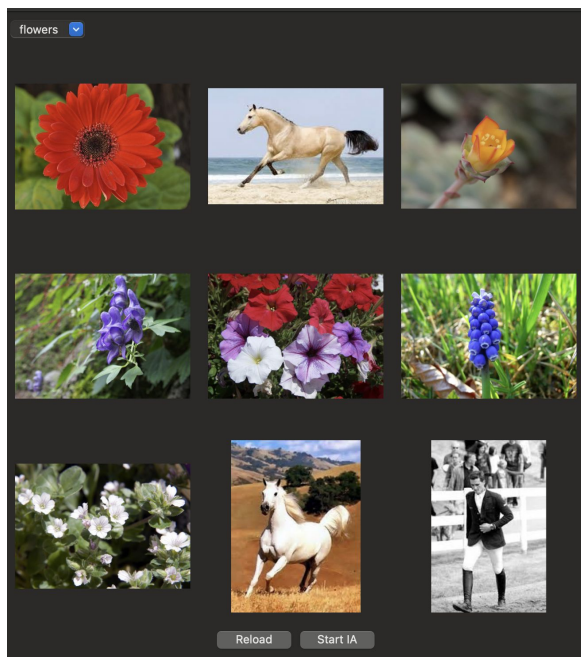


Présentation IA Captcha



Moulard Hugo

Mussard Wassim



UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE

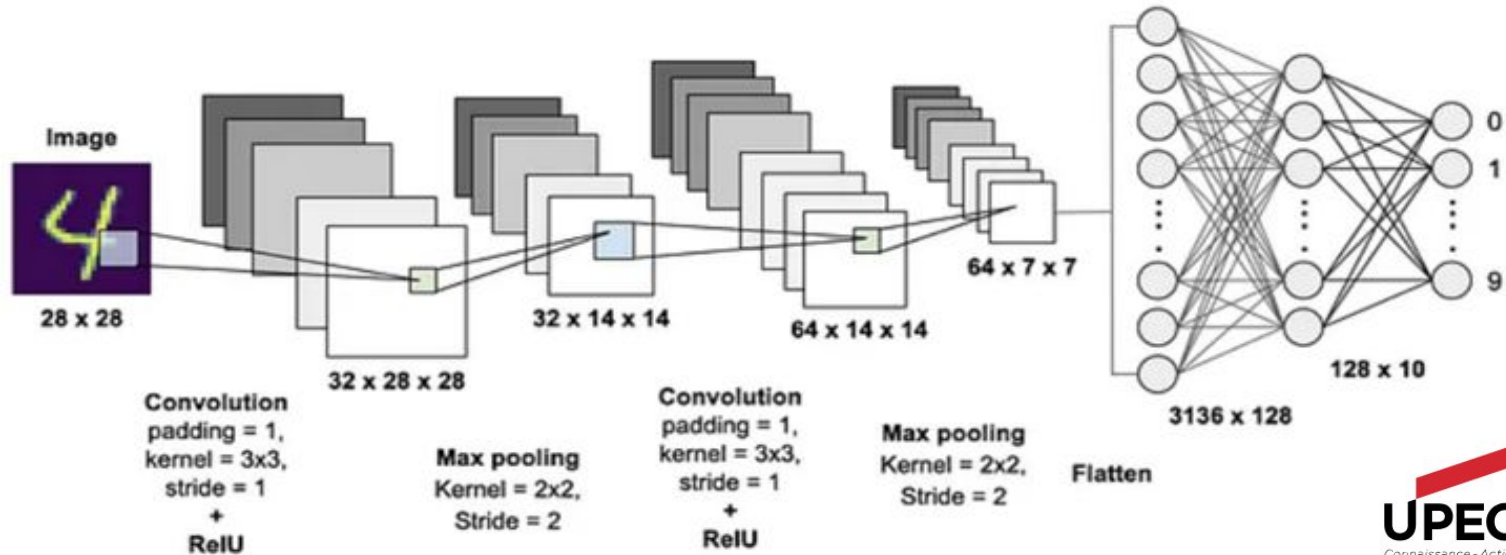


Introduction

Context :

- Création d'une ia capable de résoudre des captchas (reconnaissance d'image).
- Utilisation de tensor flow et keras.
- Kaggle
- Utilisation d'un modèle **réseau neuronal convolutif (CNN)**

Qu'est ce qu'un modèle CNN ?



Démonstration 1

Notre IA permettant de reconnaître des voitures

- Pré traitement des données
 - Notions d'époque
- > Que fait elle
- Réalisation de prédiction sur les "batch"
- Comparaison des résultats
 - Ajustement des poids.

```
# Générateur de validation
valid_datagen = ImageDataGenerator(
    rescale=1.0/255.0
)

# Récupération des images pour la phase d'entraînement
train_batches = train_datagen.flow_from_directory(
    train_path,
    target_size=(100, 100),
    batch_size=batch_size,
    class_mode='binary' # Classification Binaire : car / not_car
)

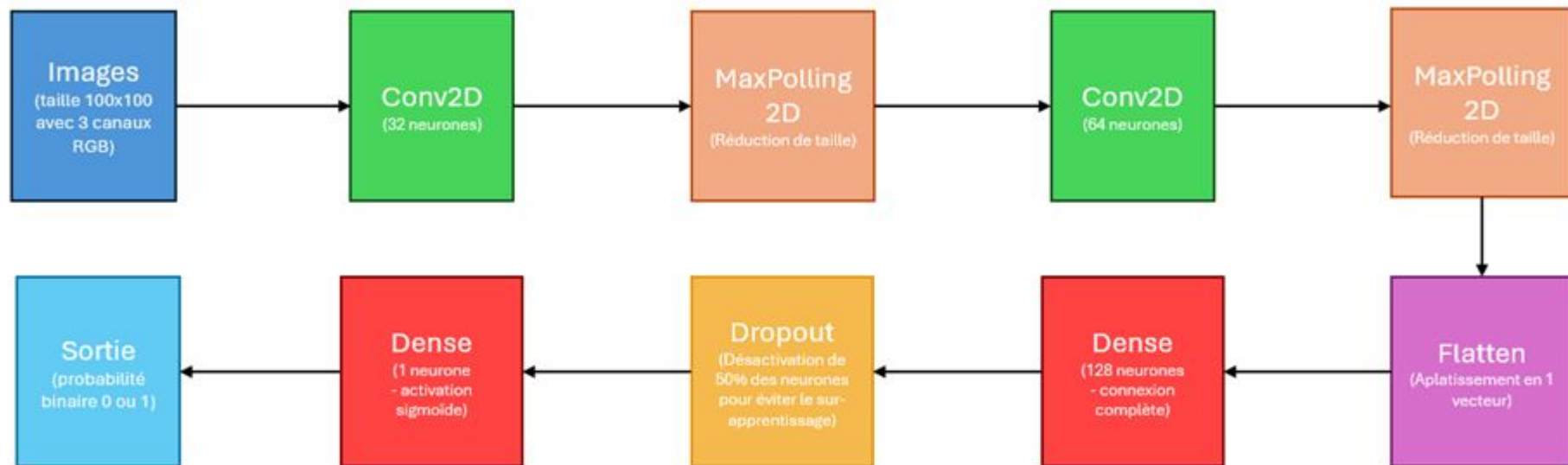
# Récupération des images pour la phase de test
valid_batches = valid_datagen.flow_from_directory(
    valid_path,
    target_size=(100, 100),
    batch_size=batch_size,
    class_mode='binary'
)

# Vérification des indices de classe
print(train_batches.class_indices) # Devrait afficher {'car': 0, 'not

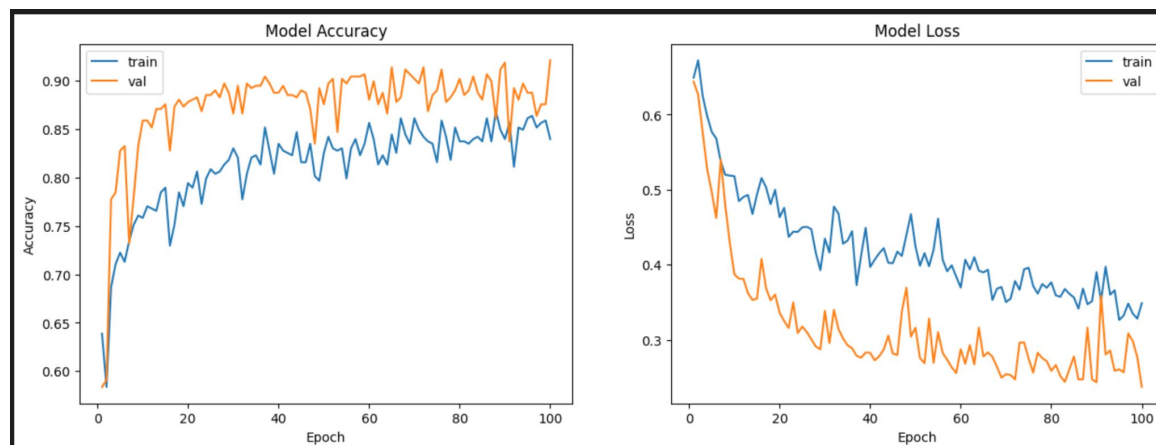
✓ 0.0s

Found 418 images belonging to 2 classes.
Found 418 images belonging to 2 classes.
{'car': 0, 'not_car': 1}
```

Représentation de notre modèle



Résultat obtenues après 100 epochs



Model took 234.02 seconds to train

1/1  0s 50ms/step – accuracy: 0.9062 – loss: 0.2331

Test-set classification accuracy: 90.62%

Résultats finaux

1/1 — 0s 22ms/step

Vraie classe : car



Prédiction : C'est une voiture avec 95.76% de confiance

1/1 — 0s 25ms/step

Vraie classe : not_car



Prédiction : Ce n'est pas une voiture avec 99.38% de confiance

1/1 — 0s 45ms/step

Vraie classe : not_car



Prédiction : C'est une voiture avec 69.41% de confiance

1/1 — 0s 43ms/step



Limitations et Axes d'améliorations

Limitations :

- Base de donnée limitée.
- Reconnaissance d'un seul type d'objet (voiture)

Solutions :

- Création d'une nouvelle ia capable de reconnaître plusieurs obj.
- Augmentation de la base de donnée.
- Passage d'une classification binaire -> catégorique

Changement dans notre nouveau modèle

```
valid_gen = ImageDataGenerator(rescale=1.0/255)

# Chargement des données d'entraînement
train_data = train_gen.flow_from_directory(
    "dataset/train/",
    target_size=(128, 128),
    batch_size=32,
    class_mode="categorical"
)

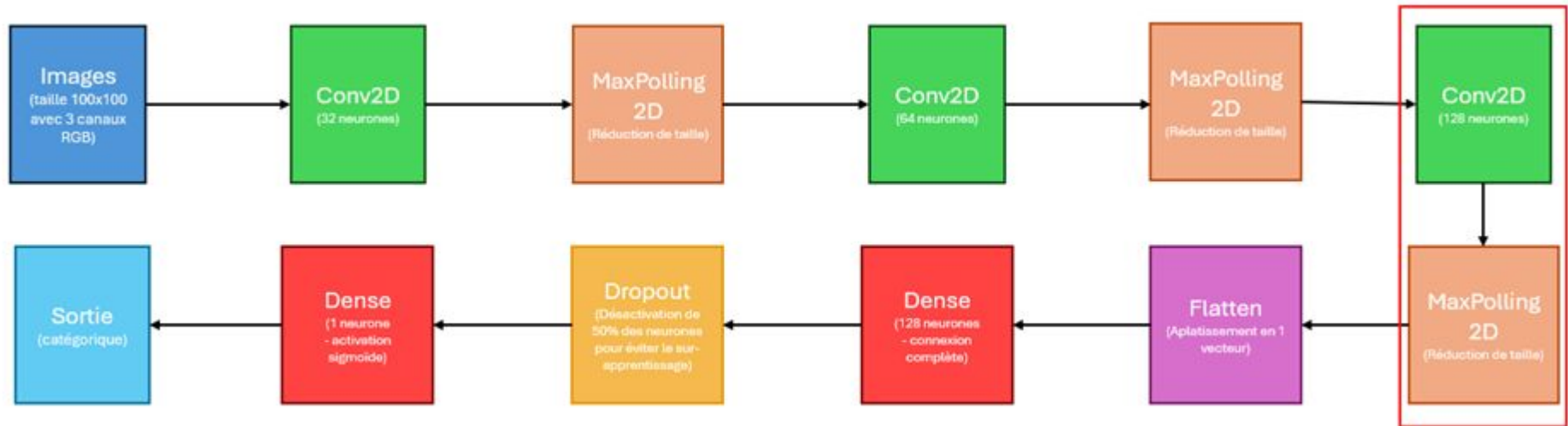
# Chargement des données de validation
val_data = valid_gen.flow_from_directory(
    "dataset/valid/",
    target_size=(128, 128),
    batch_size=32,
    class_mode="categorical"
)

categories = list(train_data.class_indices.keys()) # Extraire les catégories
print("Catégories :", categories)
```

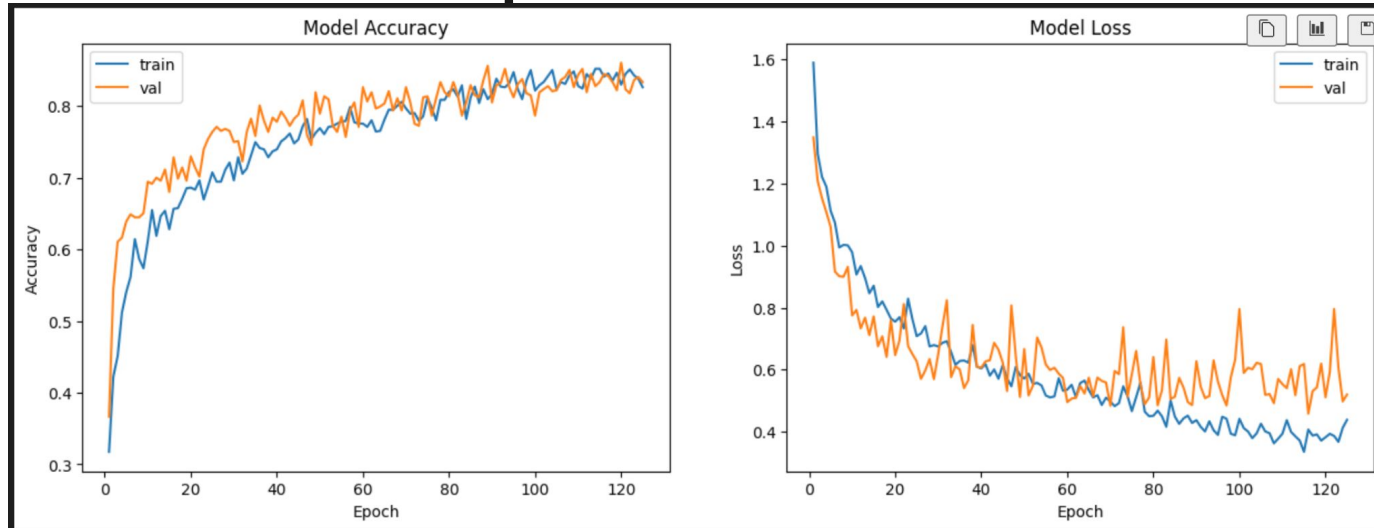
6] ✓ 0.0s

```
Found 1084 images belonging to 5 classes.
Found 704 images belonging to 5 classes.
Catégories : ['car', 'cats', 'dogs', 'flowers', 'none']
```

Ajouts de couches supplémentaires



Résultats obtenus après entraînement



Model took 2960.13 seconds to train

1/1 ————— 0s 67ms/step — accuracy: 0.9062 — loss: 0.2724

Test-set classification accuracy: 90.62%

Démonstration 2



Conclusion

Points faibles :

- Performance : Notre modèle manque encore d'entraînement dû à une quantité limitée de donnée.
- Reconnaissance : Si les objets sont similaires l'ia peut facilement les confondre (ex: chien, chats)

Points forts :

- Reconnaissance : Sur des objets non similaires le taux de reconnaissance est proche de 100%
- Performance : Entraînement rapide.