

*Direction Régionale : Casablanca – Settat*

*Cité des métiers et des compétences*

**Contrôle continu N° 2 : APPRENTISSAGE PROFOND (M108)**  
**Année 2024/2025**

**Filière :** Intelligence Artificielle

**Niveau :** TS

**Groupe :** IA101

**Nom et Prénom :** .....

**Durée :** 2H

**Variante :** 2

**Barème :** ...../20

**Partie théorique :**

**QCM (...../10 points) : Cocher la bonne réponse.**

1. **Quelle est la principale différence entre Flatten et GlobalAveragePooling dans un CNN ?** (1 point)
  - ☐ Flatten applique un filtre alors que GlobalAveragePooling ne le fait pas
  - ☐ GlobalAveragePooling augmente le nombre de paramètres du modèle
  - ☐ Flatten transforme un tenseur 2D ou 3D en vecteur, tandis que GlobalAveragePooling résume chaque carte par sa moyenne
  - ☐ Les deux donnent exactement le même résultat
2. **Quel est le rôle principal d'une couche de convolution dans un CNN ?** (1 point)
  - ☐ Réduire la dimension des images
  - ☐ Extraire des caractéristiques locales
  - ☐ Classifier directement l'image
  - ☐ Ajouter du bruit aux données
3. **Que fait une opération de convolution ?** (1 point)
  - ☐ Multiplie chaque pixel par un scalaire
  - ☐ Transforme l'image en vecteurs de caractéristiques
  - ☐ Applique un filtre glissant sur l'image pour détecter des motifs
  - ☐ Réduit la profondeur de l'image
4. **Dans un CNN, que fait une couche flatten ?** (1 point)
  - ☐ Elle applique un filtre 3D à l'image
  - ☐ Elle transforme une matrice 2D ou 3D en un vecteur 1D
  - ☐ Elle ajoute une couche de bruit
  - ☐ Elle change la profondeur de l'image
5. **Le max-pooling consiste à :** (1 point)
  - ☐ Supprimer les bords de l'image
  - ☐ Calculer la moyenne des valeurs dans chaque région
  - ☐ Conserver la valeur minimale d'un patch
  - ☐ Conserver la valeur maximale dans chaque région explorée
6. **Qu'est-ce que le transfer learning (apprentissage par transfert) ?** (1 point)
  - ☐ Entraîner un modèle à partir de zéro
  - ☐ Copier un modèle sans l'entraîner
  - ☐ Réutiliser un modèle pré-entraîné sur une nouvelle tâche
  - ☐ Transférer un modèle d'un ordinateur à un autre
7. **Quelle fonction d'activation est la plus couramment utilisée dans les CNN ?** (1 point)
  - ☐ Sigmoid
  - ☐ Softmax
  - ☐ Tanh
  - ☐ ReLU

8. Quelle couche est généralement remplacée lors d'une adaptation en transfer learning ? (1 point)

- ☐ Les premières couches convolutionnelles
- ☐ Les couches de pooling
- ☐ La couche dense (fully connected) finale
- ☐ La couche de normalisation

9. Dans le transfer learning, que signifie "geler" des couches d'un réseau ? (1 point)

- ☐ Réduire leur taille
- ☐ Les entraîner plus rapidement
- ☐ Ne pas mettre à jour leurs poids pendant l'entraînement
- ☐ Les supprimer du modèle

10. Quel est l'objectif d'une couche de pooling (max-pooling ou average-pooling) ? (1 point)

- ☐ Réduire la taille des données
- ☐ Augmenter la taille des images
- ☐ Apprendre des poids supplémentaires
- ☐ Changer la couleur des images

## Exercice 1 : Compléter les parties manquantes du code ci-dessous (.../10 points)

On souhaite entraîner un modèle de classification pour distinguer entre 20 000 images de voiture, camion, avion et train. Pour cela, nous allons construire un réseau de neurones convolutionnel (CNN) simple, avec l'architecture suivante :

- Une première couche de convolution avec 16 filtres de taille  $3 \times 3$
- Une couche de max pooling de taille  $2 \times 2$
- Une deuxième couche de convolution avec 32 filtres de taille  $2 \times 2$
- Une deuxième couche de max pooling de taille  $2 \times 2$

Prétraitement (3.5 point)

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

```
train_datagen=ImageDataGenerator(rescale=_____, validation_split=_____)
val_datagen=ImageDataGenerator(rescale=_____, validation_split=_____)
test_datagen=ImageDataGenerator(rescale=_____)
```

```
train_data=_____.flow_from_directory(train_path,
                                     target_size=(224,224),
                                     class_mode='_____',
                                     batch_size=32,
                                     subset='_____')
val_data=_____.flow_from_directory(train_path,
                                   target_size=(224,224),
                                   class_mode='_____',
                                   batch_size=32,
                                   subset='_____')
test_data=_____.flow_from_directory(test_path,
                                   target_size=(224,224),
                                   class_mode='_____',
                                   batch_size=32,
                                   shuffle=_____)
```

Création d'un modèle CNN, entraînement et évaluation du modèle (5.5 points)

```
model=_____([
    _____(_____,_____,activation='_____', input_shape=_____),
    _____(_____),
    _____(_____,_____,activation='_____' ),
    _____(_____),
    _____(),
    _____(128,activation='_____' ),
    _____(_____,activation='_____' )
])
```

```
model._____(optimizer='_____', loss='_____', metrics=['accuracy'])
```

```
history=model._____(train_data, _____=10, validation_data=_____)
```

```
loss, accuracy = model._____(test_data)
print("Accuracy:", accuracy*100)
```