CNN – Classification Intel Image Classification

Objectif : entraîner un CNN pour une classification binaire et analyser l'impact du nombre de convolutions.

In [20]:
```
# Import des librairies necessaires pour le traitement des images,
# la creation du CNN, l'entrainement du modele et le suivi des ressources
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, Callback
import os
import time
import GPUtil
import psutil
```

In [21]:
```
# Parametres globaux pour la taille des images, le batch et le nombre d'epochs
IMG_SIZE = (150, 150)
BATCH_SIZE = 32
EPOCHS = 20
```

In [22]:
```
# Preparation des donnees pour le dataset Intel (classification multi-classes)
# Normalisation des images et augmentation de donnees pour l'entrainement
train_datagen = ImageDataGenerator(
    rescale=1./255,
    zoom_range=0.2,
    shear_range=0.2,
    horizontal_flip=True
)

validation_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    "../dataset/seg_train",
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode="categorical"  # multiclasse
)

validation_generator = validation_datagen.flow_from_directory(
    "../dataset/seg_test",
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode="categorical"  # multiclasse
)
```

```
Found 14034 images belonging to 6 classes.
Found 3000 images belonging to 6 classes.
```

In [23]:
```
# Fonction qui construit un CNN pour le dataset Intel
# Le modele peut avoir 2 ou 3 convolutions et est adapte a la classification mul
def build_cnn(nb_conv):
```

```
    model = Sequential()

    # 1ère convolution
    model.add(Conv2D(32, (3,3), activation='relu', input_shape=(150,150,3)))
    model.add(MaxPooling2D((2,2)))

    # 2ème convolution
    model.add(Conv2D(32, (3,3), activation='relu'))
    model.add(MaxPooling2D((2,2)))

    # 3ème convolution optionnelle
    if nb_conv == 3:
        model.add(Conv2D(64, (3,3), activation='relu'))
        model.add(MaxPooling2D((2,2)))

    # Passage en vecteur et couches denses
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))  # regularisation pour limiter l'overfitting
    model.add(Dense(6, activation='softmax'))  # 6 classes pour Intel

    # Compilation du modele
    model.compile(
        optimizer='adam',
        loss='categorical_crossentropy',
        metrics=['accuracy']
    )

    return model
```

In [24]:
```
# Fonctions pour recuperer l'utilisation du CPU, de la RAM et du GPU pendant l'e
def get_cpu_ram():
    cpu_percent = psutil.cpu_percent(interval=1)
    ram = psutil.virtual_memory()
    ram_used_mb = ram.used / (1024 ** 2)
    return cpu_percent, ram_used_mb

def get_gpu_stats():
    try:
        gpus = GPUtil.getGPUs()
        if not gpus:
            return None, None, None
        gpu = gpus[0]
        return gpu.load * 100, gpu.memoryUsed, gpu.memoryTotal
    except:
        return None, None, None
```

In [25]:
```
# Callback personnalise pour afficher le temps et l'utilisation des ressources
# + fonction pour creer les callbacks (early stopping et sauvegarde des meilleur
class PerformanceCallback(Callback):
    def on_epoch_begin(self, epoch, logs=None):
        self.start_time = time.time()

    def on_epoch_end(self, epoch, logs=None):
        cpu, ram = get_cpu_ram()
        gpu, vram_used, vram_total = get_gpu_stats()
        duration = time.time() - self.start_time

        msg = f" | CPU: {cpu:.1f}% | RAM: {ram:.0f} MB | Time: {duration:.1f}s"
```

```
        if gpu is not None:
            msg += f" | GPU: {gpu:.1f}% | VRAM: {vram_used}/{vram_total} MB"
        print(msg)

def get_callbacks(nb_conv):
    # Creation du dossier pour stocker les poids du modele
    ckpt_dir = f"checkpoints_cnn_{nb_conv}"
    os.makedirs(ckpt_dir, exist_ok=True)

    # Sauvegarde des meilleurs poids selon la validation
    checkpoint = ModelCheckpoint(
        filepath=os.path.join(ckpt_dir, "weights_epoch_{epoch:02d}.h5"),
        monitor="val_accuracy",
        save_weights_only=True,
        save_best_only=True,
        verbose=1
    )

    # Early stopping pour arreter l'entrainement si la validation n'amelior plus
    early_stop = EarlyStopping(
        monitor="val_accuracy",
        patience=5,
        restore_best_weights=True,
        verbose=1
    )

    return checkpoint, early_stop, ckpt_dir
```

In [26]:

```
# Creation et entrainement du modele CNN a 2 convolutions sur le dataset Intel
model_2 = build_cnn(nb_conv=2)

checkpoint, early_stop, ckpt_dir = get_callbacks(2)

history_2 = model_2.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // BATCH_SIZE,
    epochs=20,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // BATCH_SIZE,
    callbacks=[checkpoint, early_stop, PerformanceCallback()],
    verbose=1
)
```

```
Epoch 1/20
438/438 [==============================] - ETA: 0s - loss: 1.1502 - accuracy: 0.5
572
Epoch 1: val_accuracy improved from -inf to 0.63474, saving model to checkpoints_
cnn_2\weights_epoch_01.h5
 | CPU: 2.2% | RAM: 15761 MB | Time: 49.2s | GPU: 5.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 112ms/step - loss: 1.1502 - accura
cy: 0.5572 - val_loss: 0.9365 - val_accuracy: 0.6347
Epoch 2/20
438/438 [==============================] - ETA: 0s - loss: 0.9286 - accuracy: 0.6
520
Epoch 2: val_accuracy improved from 0.63474 to 0.70430, saving model to checkpoin
ts_cnn_2\weights_epoch_02.h5
 | CPU: 3.6% | RAM: 15807 MB | Time: 48.8s | GPU: 9.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.9286 - accura
cy: 0.6520 - val_loss: 0.7825 - val_accuracy: 0.7043
Epoch 3/20
438/438 [==============================] - ETA: 0s - loss: 0.8284 - accuracy: 0.6
971
Epoch 3: val_accuracy improved from 0.70430 to 0.75907, saving model to checkpoin
ts_cnn_2\weights_epoch_03.h5
 | CPU: 1.2% | RAM: 15846 MB | Time: 48.0s | GPU: 4.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 48s 110ms/step - loss: 0.8284 - accura
cy: 0.6971 - val_loss: 0.6935 - val_accuracy: 0.7591
Epoch 4/20
438/438 [==============================] - ETA: 0s - loss: 0.7674 - accuracy: 0.7
181
Epoch 4: val_accuracy did not improve from 0.75907
 | CPU: 2.8% | RAM: 15813 MB | Time: 47.6s | GPU: 7.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 48s 109ms/step - loss: 0.7674 - accura
cy: 0.7181 - val_loss: 0.7013 - val_accuracy: 0.7369
Epoch 5/20
438/438 [==============================] - ETA: 0s - loss: 0.7004 - accuracy: 0.7
510
Epoch 5: val_accuracy improved from 0.75907 to 0.76882, saving model to checkpoin
ts_cnn_2\weights_epoch_05.h5
 | CPU: 4.3% | RAM: 15607 MB | Time: 48.6s | GPU: 4.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.7004 - accura
cy: 0.7510 - val_loss: 0.6346 - val_accuracy: 0.7688
Epoch 6/20
438/438 [==============================] - ETA: 0s - loss: 0.6545 - accuracy: 0.7
675
Epoch 6: val_accuracy improved from 0.76882 to 0.80712, saving model to checkpoin
ts_cnn_2\weights_epoch_06.h5
 | CPU: 3.4% | RAM: 15597 MB | Time: 48.8s | GPU: 5.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.6545 - accura
cy: 0.7675 - val_loss: 0.5594 - val_accuracy: 0.8071
Epoch 7/20
438/438 [==============================] - ETA: 0s - loss: 0.6261 - accuracy: 0.7
796
Epoch 7: val_accuracy did not improve from 0.80712
 | CPU: 3.7% | RAM: 15626 MB | Time: 48.8s | GPU: 2.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.6261 - accura
cy: 0.7796 - val_loss: 0.5756 - val_accuracy: 0.7913
Epoch 8/20
438/438 [==============================] - ETA: 0s - loss: 0.6006 - accuracy: 0.7
893
Epoch 8: val_accuracy improved from 0.80712 to 0.82661, saving model to checkpoin
ts_cnn_2\weights_epoch_08.h5
 | CPU: 2.4% | RAM: 15610 MB | Time: 48.5s | GPU: 2.0% | VRAM: 908.0/16303.0 MB
```

```
438/438 [==============================] - 49s 111ms/step - loss: 0.6006 - accura
cy: 0.7893 - val_loss: 0.5288 - val_accuracy: 0.8266
Epoch 9/20
438/438 [==============================] - ETA: 0s - loss: 0.5795 - accuracy: 0.7
972
Epoch 9: val_accuracy did not improve from 0.82661
 | CPU: 1.8% | RAM: 15616 MB | Time: 48.8s | GPU: 1.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.5795 - accura
cy: 0.7972 - val_loss: 0.5726 - val_accuracy: 0.8048
Epoch 10/20
438/438 [==============================] - ETA: 0s - loss: 0.5519 - accuracy: 0.8
035
Epoch 10: val_accuracy did not improve from 0.82661
 | CPU: 1.9% | RAM: 15692 MB | Time: 48.6s | GPU: 2.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.5519 - accura
cy: 0.8035 - val_loss: 0.5552 - val_accuracy: 0.8048
Epoch 11/20
438/438 [==============================] - ETA: 0s - loss: 0.5320 - accuracy: 0.8
133
Epoch 11: val_accuracy improved from 0.82661 to 0.82964, saving model to checkpoi
nts_cnn_2\weights_epoch_11.h5
 | CPU: 1.2% | RAM: 15640 MB | Time: 48.5s | GPU: 2.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 48s 111ms/step - loss: 0.5320 - accura
cy: 0.8133 - val_loss: 0.5166 - val_accuracy: 0.8296
Epoch 12/20
438/438 [==============================] - ETA: 0s - loss: 0.5234 - accuracy: 0.8
157
Epoch 12: val_accuracy did not improve from 0.82964
 | CPU: 2.5% | RAM: 15728 MB | Time: 48.5s | GPU: 2.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.5234 - accura
cy: 0.8157 - val_loss: 0.4966 - val_accuracy: 0.8266
Epoch 13/20
438/438 [==============================] - ETA: 0s - loss: 0.5053 - accuracy: 0.8
210
Epoch 13: val_accuracy did not improve from 0.82964
 | CPU: 3.1% | RAM: 15672 MB | Time: 48.6s | GPU: 5.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 111ms/step - loss: 0.5053 - accura
cy: 0.8210 - val_loss: 0.5189 - val_accuracy: 0.8283
Epoch 14/20
438/438 [==============================] - ETA: 0s - loss: 0.4957 - accuracy: 0.8
231
Epoch 14: val_accuracy improved from 0.82964 to 0.83065, saving model to checkpoi
nts_cnn_2\weights_epoch_14.h5
 | CPU: 2.8% | RAM: 15707 MB | Time: 49.0s | GPU: 4.0% | VRAM: 908.0/16303.0 MB
438/438 [==============================] - 49s 112ms/step - loss: 0.4957 - accura
cy: 0.8231 - val_loss: 0.5127 - val_accuracy: 0.8306
Epoch 15/20
438/438 [==============================] - ETA: 0s - loss: 0.4798 - accuracy: 0.8
288
Epoch 15: val_accuracy did not improve from 0.83065
 | CPU: 2.1% | RAM: 15672 MB | Time: 49.1s | GPU: 2.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 112ms/step - loss: 0.4798 - accura
cy: 0.8288 - val_loss: 0.5483 - val_accuracy: 0.8196
Epoch 16/20
438/438 [==============================] - ETA: 0s - loss: 0.4738 - accuracy: 0.8
318
Epoch 16: val_accuracy did not improve from 0.83065
 | CPU: 1.9% | RAM: 15740 MB | Time: 49.3s | GPU: 4.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 112ms/step - loss: 0.4738 - accura
cy: 0.8318 - val_loss: 0.5332 - val_accuracy: 0.8135
```

```
Epoch 17/20
438/438 [==============================] - ETA: 0s - loss: 0.4645 - accuracy: 0.8
355
Epoch 17: val_accuracy improved from 0.83065 to 0.84341, saving model to checkpoi
nts_cnn_2\weights_epoch_17.h5
 | CPU: 2.9% | RAM: 15715 MB | Time: 49.2s | GPU: 2.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 112ms/step - loss: 0.4645 - accura
cy: 0.8355 - val_loss: 0.4700 - val_accuracy: 0.8434
Epoch 18/20
438/438 [==============================] - ETA: 0s - loss: 0.4445 - accuracy: 0.8
409
Epoch 18: val_accuracy did not improve from 0.84341
 | CPU: 3.8% | RAM: 15628 MB | Time: 49.0s | GPU: 3.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 49s 112ms/step - loss: 0.4445 - accura
cy: 0.8409 - val_loss: 0.4839 - val_accuracy: 0.8397
Epoch 19/20
438/438 [==============================] - ETA: 0s - loss: 0.4496 - accuracy: 0.8
390
Epoch 19: val_accuracy improved from 0.84341 to 0.84677, saving model to checkpoi
nts_cnn_2\weights_epoch_19.h5
 | CPU: 2.0% | RAM: 15626 MB | Time: 49.9s | GPU: 2.0% | VRAM: 856.0/16303.0 MB
438/438 [==============================] - 50s 114ms/step - loss: 0.4496 - accura
cy: 0.8390 - val_loss: 0.4738 - val_accuracy: 0.8468
Epoch 20/20
438/438 [==============================] - ETA: 0s - loss: 0.4330 - accuracy: 0.8
442
Epoch 20: val_accuracy improved from 0.84677 to 0.85148, saving model to checkpoi
nts_cnn_2\weights_epoch_20.h5
 | CPU: 2.2% | RAM: 15672 MB | Time: 49.3s | GPU: 4.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 49s 113ms/step - loss: 0.4330 - accura
cy: 0.8442 - val_loss: 0.4574 - val_accuracy: 0.8515
```

In [27]:

```
# Creation et entrainement du modele CNN a 3 convolutions pour le dataset Intel
model_3 = build_cnn(nb_conv=3)

checkpoint, early_stop, ckpt_dir = get_callbacks(3)

history_3 = model_3.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // BATCH_SIZE,
    epochs=20,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // BATCH_SIZE,
    callbacks=[checkpoint, early_stop, PerformanceCallback()],
    verbose=1
)
```

```
Epoch 1/20
438/438 [==============================] - ETA: 0s - loss: 1.1305 - accuracy: 0.5
575
Epoch 1: val_accuracy improved from -inf to 0.67977, saving model to checkpoints_
cnn_3\weights_epoch_01.h5
 | CPU: 2.3% | RAM: 15695 MB | Time: 51.3s | GPU: 2.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 51s 117ms/step - loss: 1.1305 - accura
cy: 0.5575 - val_loss: 0.8409 - val_accuracy: 0.6798
Epoch 2/20
438/438 [==============================] - ETA: 0s - loss: 0.9041 - accuracy: 0.6
563
Epoch 2: val_accuracy improved from 0.67977 to 0.75974, saving model to checkpoin
ts_cnn_3\weights_epoch_02.h5
 | CPU: 2.6% | RAM: 15666 MB | Time: 52.0s | GPU: 2.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 52s 119ms/step - loss: 0.9041 - accura
cy: 0.6563 - val_loss: 0.6708 - val_accuracy: 0.7597
Epoch 3/20
438/438 [==============================] - ETA: 0s - loss: 0.7605 - accuracy: 0.7
265
Epoch 3: val_accuracy improved from 0.75974 to 0.78293, saving model to checkpoin
ts_cnn_3\weights_epoch_03.h5
 | CPU: 2.5% | RAM: 15652 MB | Time: 51.3s | GPU: 2.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 51s 117ms/step - loss: 0.7605 - accura
cy: 0.7265 - val_loss: 0.6015 - val_accuracy: 0.7829
Epoch 4/20
438/438 [==============================] - ETA: 0s - loss: 0.6685 - accuracy: 0.7
590
Epoch 4: val_accuracy improved from 0.78293 to 0.80981, saving model to checkpoin
ts_cnn_3\weights_epoch_04.h5
 | CPU: 2.4% | RAM: 15642 MB | Time: 50.9s | GPU: 4.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 51s 116ms/step - loss: 0.6685 - accura
cy: 0.7590 - val_loss: 0.5332 - val_accuracy: 0.8098
Epoch 5/20
438/438 [==============================] - ETA: 0s - loss: 0.6147 - accuracy: 0.7
805
Epoch 5: val_accuracy improved from 0.80981 to 0.81586, saving model to checkpoin
ts_cnn_3\weights_epoch_05.h5
 | CPU: 2.2% | RAM: 15678 MB | Time: 51.0s | GPU: 4.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 51s 116ms/step - loss: 0.6147 - accura
cy: 0.7805 - val_loss: 0.5376 - val_accuracy: 0.8159
Epoch 6/20
438/438 [==============================] - ETA: 0s - loss: 0.5599 - accuracy: 0.8
008
Epoch 6: val_accuracy improved from 0.81586 to 0.84140, saving model to checkpoin
ts_cnn_3\weights_epoch_06.h5
 | CPU: 1.7% | RAM: 15714 MB | Time: 51.7s | GPU: 2.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 52s 118ms/step - loss: 0.5599 - accura
cy: 0.8008 - val_loss: 0.4670 - val_accuracy: 0.8414
Epoch 7/20
438/438 [==============================] - ETA: 0s - loss: 0.5377 - accuracy: 0.8
119
Epoch 7: val_accuracy improved from 0.84140 to 0.85013, saving model to checkpoin
ts_cnn_3\weights_epoch_07.h5
 | CPU: 1.9% | RAM: 15726 MB | Time: 51.6s | GPU: 2.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 52s 118ms/step - loss: 0.5377 - accura
cy: 0.8119 - val_loss: 0.4576 - val_accuracy: 0.8501
Epoch 8/20
438/438 [==============================] - ETA: 0s - loss: 0.5085 - accuracy: 0.8
200
Epoch 8: val_accuracy did not improve from 0.85013
```

```
 | CPU: 5.8% | RAM: 15711 MB | Time: 51.3s | GPU: 4.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 51s 117ms/step - loss: 0.5085 - accura
cy: 0.8200 - val_loss: 0.4614 - val_accuracy: 0.8454
Epoch 9/20
438/438 [==============================] - ETA: 0s - loss: 0.4905 - accuracy: 0.8
288
Epoch 9: val_accuracy improved from 0.85013 to 0.85181, saving model to checkpoin
ts_cnn_3\weights_epoch_09.h5
 | CPU: 2.1% | RAM: 15676 MB | Time: 50.6s | GPU: 4.0% | VRAM: 935.0/16303.0 MB
438/438 [==============================] - 51s 115ms/step - loss: 0.4905 - accura
cy: 0.8288 - val_loss: 0.4358 - val_accuracy: 0.8518
Epoch 10/20
438/438 [==============================] - ETA: 0s - loss: 0.4688 - accuracy: 0.8
356
Epoch 10: val_accuracy did not improve from 0.85181
 | CPU: 2.3% | RAM: 15717 MB | Time: 51.6s | GPU: 9.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 52s 118ms/step - loss: 0.4688 - accura
cy: 0.8356 - val_loss: 0.4526 - val_accuracy: 0.8464
Epoch 11/20
438/438 [==============================] - ETA: 0s - loss: 0.4521 - accuracy: 0.8
410
Epoch 11: val_accuracy did not improve from 0.85181
 | CPU: 2.4% | RAM: 16006 MB | Time: 52.3s | GPU: 3.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 52s 119ms/step - loss: 0.4521 - accura
cy: 0.8410 - val_loss: 0.4533 - val_accuracy: 0.8384
Epoch 12/20
438/438 [==============================] - ETA: 0s - loss: 0.4311 - accuracy: 0.8
482
Epoch 12: val_accuracy did not improve from 0.85181
 | CPU: 3.1% | RAM: 15954 MB | Time: 54.3s | GPU: 5.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 54s 124ms/step - loss: 0.4311 - accura
cy: 0.8482 - val_loss: 0.4725 - val_accuracy: 0.8306
Epoch 13/20
438/438 [==============================] - ETA: 0s - loss: 0.4209 - accuracy: 0.8
515
Epoch 13: val_accuracy improved from 0.85181 to 0.86022, saving model to checkpoi
nts_cnn_3\weights_epoch_13.h5
 | CPU: 1.7% | RAM: 15930 MB | Time: 54.0s | GPU: 2.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 54s 123ms/step - loss: 0.4209 - accura
cy: 0.8515 - val_loss: 0.4172 - val_accuracy: 0.8602
Epoch 14/20
438/438 [==============================] - ETA: 0s - loss: 0.4001 - accuracy: 0.8
594
Epoch 14: val_accuracy did not improve from 0.86022
 | CPU: 1.7% | RAM: 15666 MB | Time: 54.3s | GPU: 3.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 54s 124ms/step - loss: 0.4001 - accura
cy: 0.8594 - val_loss: 0.4368 - val_accuracy: 0.8511
Epoch 15/20
438/438 [==============================] - ETA: 0s - loss: 0.3937 - accuracy: 0.8
594
Epoch 15: val_accuracy improved from 0.86022 to 0.86358, saving model to checkpoi
nts_cnn_3\weights_epoch_15.h5
 | CPU: 2.4% | RAM: 15769 MB | Time: 53.7s | GPU: 1.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 54s 122ms/step - loss: 0.3937 - accura
cy: 0.8594 - val_loss: 0.4002 - val_accuracy: 0.8636
Epoch 16/20
438/438 [==============================] - ETA: 0s - loss: 0.3774 - accuracy: 0.8
662
Epoch 16: val_accuracy did not improve from 0.86358
 | CPU: 1.9% | RAM: 15707 MB | Time: 53.7s | GPU: 5.0% | VRAM: 883.0/16303.0 MB
```

```
438/438 [==============================] - 54s 123ms/step - loss: 0.3774 - accura
cy: 0.8662 - val_loss: 0.4448 - val_accuracy: 0.8636
Epoch 17/20
438/438 [==============================] - ETA: 0s - loss: 0.3771 - accuracy: 0.8
664
Epoch 17: val_accuracy improved from 0.86358 to 0.86593, saving model to checkpoi
nts_cnn_3\weights_epoch_17.h5
 | CPU: 6.4% | RAM: 15705 MB | Time: 53.7s | GPU: 2.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 54s 123ms/step - loss: 0.3771 - accura
cy: 0.8664 - val_loss: 0.3979 - val_accuracy: 0.8659
Epoch 18/20
438/438 [==============================] - ETA: 0s - loss: 0.3650 - accuracy: 0.8
685
Epoch 18: val_accuracy did not improve from 0.86593
 | CPU: 1.5% | RAM: 15666 MB | Time: 53.5s | GPU: 3.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 54s 122ms/step - loss: 0.3650 - accura
cy: 0.8685 - val_loss: 0.4513 - val_accuracy: 0.8632
Epoch 19/20
438/438 [==============================] - ETA: 0s - loss: 0.3375 - accuracy: 0.8
779
Epoch 19: val_accuracy improved from 0.86593 to 0.86862, saving model to checkpoi
nts_cnn_3\weights_epoch_19.h5
 | CPU: 3.2% | RAM: 15662 MB | Time: 53.5s | GPU: 3.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 53s 122ms/step - loss: 0.3375 - accura
cy: 0.8779 - val_loss: 0.4229 - val_accuracy: 0.8686
Epoch 20/20
438/438 [==============================] - ETA: 0s - loss: 0.3389 - accuracy: 0.8
767
Epoch 20: val_accuracy did not improve from 0.86862
 | CPU: 2.1% | RAM: 15650 MB | Time: 53.8s | GPU: 3.0% | VRAM: 883.0/16303.0 MB
438/438 [==============================] - 54s 123ms/step - loss: 0.3389 - accura
cy: 0.8767 - val_loss: 0.4119 - val_accuracy: 0.8659
```

In [28]:

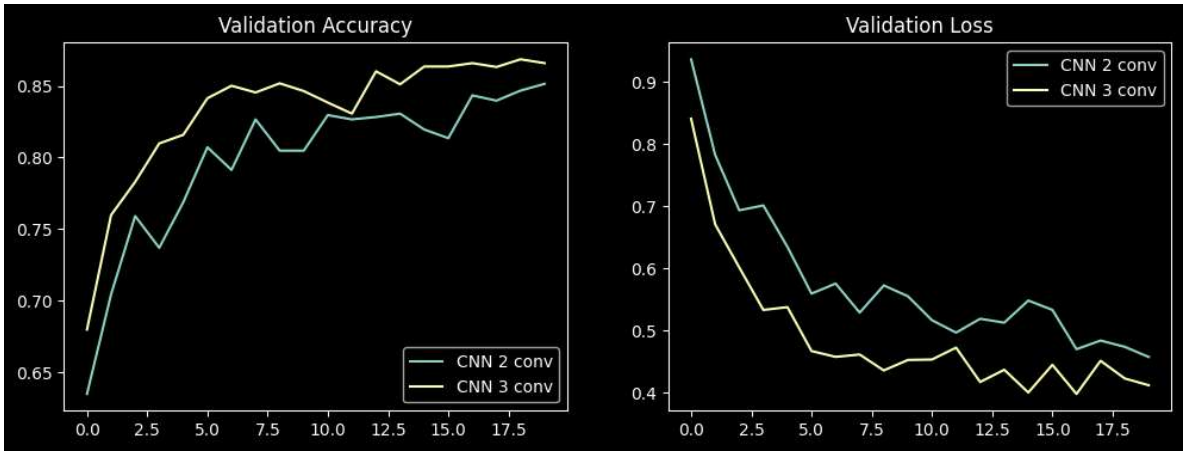
```python
import matplotlib.pyplot as plt

plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
plt.plot(history_2.history["val_accuracy"], label="CNN 2 conv")
plt.plot(history_3.history["val_accuracy"], label="CNN 3 conv")
plt.title("Validation Accuracy")
plt.legend()

plt.subplot(1,2,2)
plt.plot(history_2.history["val_loss"], label="CNN 2 conv")
plt.plot(history_3.history["val_loss"], label="CNN 3 conv")
plt.title("Validation Loss")
plt.legend()

plt.show()
```



Comparaison CNN 2 convolutions vs CNN 3 convolutions

L'entraînement des deux architectures montre que l'ajout d'une troisième couche de convolution améliore les performances du modèle, tout en conservant un entraînement stable sur ce nouveau dataset. Les tendances observées sont cohérentes avec celles obtenues précédemment sur le dataset Dogs & Cats.

CNN à 2 convolutions :

- Précision maximale sur l'ensemble de validation : 85,15 % (epoch 20)
- Perte de validation minimale : 0,4574
- Meilleur modèle atteint en fin d'entraînement, sans déclenchement de l'early stopping

CNN à 3 convolutions :

- Précision maximale sur l'ensemble de validation : 86,86 % (epoch 19)
- Perte de validation minimale : 0,4229
- Meilleur modèle atteint à l'epoch 19, avec une légère dégradation ensuite

Le graphique comparatif montre que le CNN à 3 convolutions converge plus rapidement et maintient une précision de validation supérieure à celle du CNN à 2 convolutions sur la majorité des epochs. La perte de validation plus faible observée pour le modèle à 3 convolutions traduit une meilleure capacité de généralisation, sans apparition d'overfitting significatif.

Ainsi, ce second jeu de données confirme que l'augmentation modérée de la profondeur d'un CNN améliore la performance globale du modèle, comme observé précédemment sur Dogs & Cats.

Lors de ces entraînements CNN, l'utilisation des ressources reste globalement faible et stable : le CPU tourne surtout autour de 1–4 % (avec quelques pointes vers 6 %), la RAM reste proche de 15,6–15,9 Go, et la VRAM reste < 1 Go (environ 856–935 Mo). L'utilisation GPU est modérée (souvent 2–5 %, avec un pic à 9 % sur le 3 conv), tandis que le temps par epoch est d'environ 49 s (2 conv) contre 51–54 s (3 conv), ce qui est cohérent avec un modèle plus profond.