CNN – Classification Dogs vs Cats

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

In [42]:
```
# 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 [43]:
```
# Parametres globaux pour la taille des images, le batch et le nombre d'epochs
IMG_SIZE = (64, 64)
BATCH_SIZE = 32
EPOCHS = 20
```

In [44]:
```
# Creation des generateurs d'images avec normalisation et augmentation de donnee
# Les images sont chargees depuis les dossiers et preparees pour l'entrainement
train_gen = ImageDataGenerator(
    rescale=1./255,
    zoom_range=0.2,
    shear_range=0.2,
    horizontal_flip=True
)

test_gen = ImageDataGenerator(rescale=1./255)

train_set = train_gen.flow_from_directory(
    "../dataset/training_set",
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode="binary"
)

test_set = test_gen.flow_from_directory(
    "../dataset/test_set",
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode="binary"
)
```

```
Found 8000 images belonging to 2 classes.
Found 2000 images belonging to 2 classes.
```

In [45]:
```
# Fonction qui construit un CNN avec 2 ou 3 couches de convolution selon le para
def build_cnn(nb_conv):
    model = Sequential()
```

```
    # 1ère convolution
    model.add(Conv2D(32, (3,3), activation='relu', input_shape=(64,64,3)))
    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 puis couches denses pour la classification
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))

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

    return model
```

In [46]:

```
# Fonctions utilitaires pour recuperer l'utilisation du CPU, de la RAM et du GPU
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 [47]:

```
# Callback personnalise pour afficher le temps d'execution et l'utilisation des
# + fonction pour creer les callbacks (early stopping et sauvegarde des poids)
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):
    # Dossier pour stocker les poids du modele
    ckpt_dir = f"checkpoints_cnn_{nb_conv}"
    os.makedirs(ckpt_dir, exist_ok=True)

    # Sauvegarde des poids a chaque epoch
    checkpoint = ModelCheckpoint(
        filepath=os.path.join(ckpt_dir, "weights_epoch_{epoch:02d}.h5"),
        monitor="val_loss",
        save_weights_only=True,
        save_best_only=False,
        verbose=1
    )

    # Early stopping pour arreter l'entrainement si la validation n'ameliore plu
    early_stop = EarlyStopping(
        monitor="val_loss",
        patience=3,
        restore_best_weights=True,
        verbose=1
    )

    return checkpoint, early_stop, ckpt_dir
```

In [48]:
```
# Creation et entrainement du modele CNN a 2 convolutions
model_2 = build_cnn(nb_conv=2)

checkpoint, early_stop, ckpt_dir = get_callbacks(2)

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

```
Epoch 1/20
250/250 [==============================] - ETA: 0s - loss: 0.6836 - accuracy: 0.5
564
Epoch 1: saving model to checkpoints_cnn_2\weights_epoch_01.h5
 | CPU: 4.6% | RAM: 15655 MB | Time: 14.0s | GPU: 30.0% | VRAM: 764.0/16303.0 MB
250/250 [==============================] - 14s 56ms/step - loss: 0.6836 - accurac
y: 0.5564 - val_loss: 0.6417 - val_accuracy: 0.6028
Epoch 2/20
249/250 [============================>.] - ETA: 0s - loss: 0.6052 - accuracy: 0.6
711
Epoch 2: saving model to checkpoints_cnn_2\weights_epoch_02.h5
 | CPU: 3.6% | RAM: 15688 MB | Time: 12.1s | GPU: 42.0% | VRAM: 764.0/16303.0 MB
250/250 [==============================] - 12s 48ms/step - loss: 0.6051 - accurac
y: 0.6712 - val_loss: 0.5606 - val_accuracy: 0.7233
Epoch 3/20
249/250 [============================>.] - ETA: 0s - loss: 0.5646 - accuracy: 0.7
101
Epoch 3: saving model to checkpoints_cnn_2\weights_epoch_03.h5
 | CPU: 2.7% | RAM: 15681 MB | Time: 12.1s | GPU: 2.0% | VRAM: 774.0/16303.0 MB
250/250 [==============================] - 12s 48ms/step - loss: 0.5647 - accurac
y: 0.7100 - val_loss: 0.5898 - val_accuracy: 0.7082
Epoch 4/20
249/250 [============================>.] - ETA: 0s - loss: 0.5413 - accuracy: 0.7
265
Epoch 4: saving model to checkpoints_cnn_2\weights_epoch_04.h5
 | CPU: 4.1% | RAM: 15687 MB | Time: 12.3s | GPU: 34.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 49ms/step - loss: 0.5412 - accurac
y: 0.7269 - val_loss: 0.5411 - val_accuracy: 0.7349
Epoch 5/20
249/250 [============================>.] - ETA: 0s - loss: 0.5200 - accuracy: 0.7
390
Epoch 5: saving model to checkpoints_cnn_2\weights_epoch_05.h5
 | CPU: 2.3% | RAM: 15684 MB | Time: 11.8s | GPU: 3.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.5201 - accurac
y: 0.7391 - val_loss: 0.5263 - val_accuracy: 0.7404
Epoch 6/20
249/250 [============================>.] - ETA: 0s - loss: 0.4991 - accuracy: 0.7
572
Epoch 6: saving model to checkpoints_cnn_2\weights_epoch_06.h5
 | CPU: 1.6% | RAM: 15669 MB | Time: 11.7s | GPU: 1.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.4998 - accurac
y: 0.7570 - val_loss: 0.4955 - val_accuracy: 0.7656
Epoch 7/20
249/250 [============================>.] - ETA: 0s - loss: 0.4843 - accuracy: 0.7
643
Epoch 7: saving model to checkpoints_cnn_2\weights_epoch_07.h5
 | CPU: 2.4% | RAM: 15545 MB | Time: 11.5s | GPU: 2.0% | VRAM: 833.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4839 - accurac
y: 0.7648 - val_loss: 0.5282 - val_accuracy: 0.7535
Epoch 8/20
249/250 [============================>.] - ETA: 0s - loss: 0.4732 - accuracy: 0.7
713
Epoch 8: saving model to checkpoints_cnn_2\weights_epoch_08.h5
 | CPU: 3.4% | RAM: 15680 MB | Time: 11.7s | GPU: 2.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.4733 - accurac
y: 0.7711 - val_loss: 0.4756 - val_accuracy: 0.7767
Epoch 9/20
249/250 [============================>.] - ETA: 0s - loss: 0.4618 - accuracy: 0.7
810
Epoch 9: saving model to checkpoints_cnn_2\weights_epoch_09.h5
```

```
 | CPU: 1.9% | RAM: 15681 MB | Time: 11.4s | GPU: 2.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 11s 46ms/step - loss: 0.4617 - accurac
y: 0.7807 - val_loss: 0.5345 - val_accuracy: 0.7566
Epoch 10/20
249/250 [============================>.] - ETA: 0s - loss: 0.4509 - accuracy: 0.7
844
Epoch 10: saving model to checkpoints_cnn_2\weights_epoch_10.h5
 | CPU: 4.8% | RAM: 15701 MB | Time: 11.6s | GPU: 3.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4512 - accurac
y: 0.7841 - val_loss: 0.4737 - val_accuracy: 0.7802
Epoch 11/20
249/250 [============================>.] - ETA: 0s - loss: 0.4373 - accuracy: 0.8
007
Epoch 11: saving model to checkpoints_cnn_2\weights_epoch_11.h5
 | CPU: 2.2% | RAM: 15737 MB | Time: 11.5s | GPU: 4.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4368 - accurac
y: 0.8009 - val_loss: 0.4498 - val_accuracy: 0.7888
Epoch 12/20
249/250 [============================>.] - ETA: 0s - loss: 0.4283 - accuracy: 0.7
984
Epoch 12: saving model to checkpoints_cnn_2\weights_epoch_12.h5
 | CPU: 1.0% | RAM: 15726 MB | Time: 11.5s | GPU: 4.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4284 - accurac
y: 0.7983 - val_loss: 0.4527 - val_accuracy: 0.7918
Epoch 13/20
249/250 [============================>.] - ETA: 0s - loss: 0.4161 - accuracy: 0.8
047
Epoch 13: saving model to checkpoints_cnn_2\weights_epoch_13.h5
 | CPU: 2.8% | RAM: 15718 MB | Time: 11.5s | GPU: 5.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4165 - accurac
y: 0.8046 - val_loss: 0.4879 - val_accuracy: 0.7767
Epoch 14/20
249/250 [============================>.] - ETA: 0s - loss: 0.4046 - accuracy: 0.8
146
Epoch 14: saving model to checkpoints_cnn_2\weights_epoch_14.h5
Restoring model weights from the end of the best epoch: 11.
 | CPU: 2.1% | RAM: 15719 MB | Time: 11.6s | GPU: 2.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4044 - accurac
y: 0.8148 - val_loss: 0.4688 - val_accuracy: 0.7828
Epoch 14: early stopping
```

In [49]:

```python
# Creation et entrainement du modele CNN a 3 convolutions pour comparer avec le
model_3 = build_cnn(nb_conv=3)

checkpoint, early_stop, ckpt_dir = get_callbacks(3)

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

```
Epoch 1/20
249/250 [============================>.] - ETA: 0s - loss: 0.6688 - accuracy: 0.5
833
Epoch 1: saving model to checkpoints_cnn_3\weights_epoch_01.h5
 | CPU: 1.9% | RAM: 15724 MB | Time: 12.2s | GPU: 7.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 48ms/step - loss: 0.6691 - accurac
y: 0.5831 - val_loss: 0.5921 - val_accuracy: 0.6951
Epoch 2/20
249/250 [============================>.] - ETA: 0s - loss: 0.6036 - accuracy: 0.6
679
Epoch 2: saving model to checkpoints_cnn_3\weights_epoch_02.h5
 | CPU: 2.7% | RAM: 15732 MB | Time: 11.9s | GPU: 5.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 48ms/step - loss: 0.6032 - accurac
y: 0.6681 - val_loss: 0.5703 - val_accuracy: 0.7107
Epoch 3/20
249/250 [============================>.] - ETA: 0s - loss: 0.5748 - accuracy: 0.6
963
Epoch 3: saving model to checkpoints_cnn_3\weights_epoch_03.h5
 | CPU: 2.4% | RAM: 15729 MB | Time: 11.8s | GPU: 2.0% | VRAM: 833.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.5751 - accurac
y: 0.6960 - val_loss: 0.5256 - val_accuracy: 0.7399
Epoch 4/20
249/250 [============================>.] - ETA: 0s - loss: 0.5270 - accuracy: 0.7
324
Epoch 4: saving model to checkpoints_cnn_3\weights_epoch_04.h5
 | CPU: 2.0% | RAM: 15726 MB | Time: 11.8s | GPU: 2.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.5270 - accurac
y: 0.7325 - val_loss: 0.5045 - val_accuracy: 0.7560
Epoch 5/20
249/250 [============================>.] - ETA: 0s - loss: 0.5022 - accuracy: 0.7
551
Epoch 5: saving model to checkpoints_cnn_3\weights_epoch_05.h5
 | CPU: 1.7% | RAM: 15699 MB | Time: 11.7s | GPU: 1.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.5022 - accurac
y: 0.7549 - val_loss: 0.4900 - val_accuracy: 0.7621
Epoch 6/20
249/250 [============================>.] - ETA: 0s - loss: 0.4769 - accuracy: 0.7
668
Epoch 6: saving model to checkpoints_cnn_3\weights_epoch_06.h5
 | CPU: 2.6% | RAM: 15702 MB | Time: 11.7s | GPU: 2.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.4762 - accurac
y: 0.7674 - val_loss: 0.4569 - val_accuracy: 0.7772
Epoch 7/20
249/250 [============================>.] - ETA: 0s - loss: 0.4580 - accuracy: 0.7
816
Epoch 7: saving model to checkpoints_cnn_3\weights_epoch_07.h5
 | CPU: 2.8% | RAM: 15725 MB | Time: 11.8s | GPU: 1.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.4586 - accurac
y: 0.7807 - val_loss: 0.4387 - val_accuracy: 0.7944
Epoch 8/20
249/250 [============================>.] - ETA: 0s - loss: 0.4457 - accuracy: 0.7
868
Epoch 8: saving model to checkpoints_cnn_3\weights_epoch_08.h5
 | CPU: 1.8% | RAM: 15696 MB | Time: 11.5s | GPU: 4.0% | VRAM: 833.0/16303.0 MB
250/250 [==============================] - 11s 46ms/step - loss: 0.4457 - accurac
y: 0.7868 - val_loss: 0.4399 - val_accuracy: 0.7959
Epoch 9/20
249/250 [============================>.] - ETA: 0s - loss: 0.4183 - accuracy: 0.8
021
Epoch 9: saving model to checkpoints_cnn_3\weights_epoch_09.h5
```

```
 | CPU: 1.9% | RAM: 15706 MB | Time: 11.5s | GPU: 1.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4184 - accurac
y: 0.8019 - val_loss: 0.4710 - val_accuracy: 0.7737
Epoch 10/20
249/250 [============================>.] - ETA: 0s - loss: 0.4059 - accuracy: 0.8
111
Epoch 10: saving model to checkpoints_cnn_3\weights_epoch_10.h5
 | CPU: 1.7% | RAM: 15718 MB | Time: 11.5s | GPU: 2.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.4056 - accurac
y: 0.8111 - val_loss: 0.4241 - val_accuracy: 0.7989
Epoch 11/20
249/250 [============================>.] - ETA: 0s - loss: 0.3997 - accuracy: 0.8
148
Epoch 11: saving model to checkpoints_cnn_3\weights_epoch_11.h5
 | CPU: 2.9% | RAM: 15737 MB | Time: 11.6s | GPU: 3.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.3994 - accurac
y: 0.8149 - val_loss: 0.4228 - val_accuracy: 0.8029
Epoch 12/20
249/250 [============================>.] - ETA: 0s - loss: 0.3874 - accuracy: 0.8
228
Epoch 12: saving model to checkpoints_cnn_3\weights_epoch_12.h5
 | CPU: 1.2% | RAM: 15706 MB | Time: 11.5s | GPU: 5.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 11s 46ms/step - loss: 0.3878 - accurac
y: 0.8225 - val_loss: 0.4186 - val_accuracy: 0.8024
Epoch 13/20
249/250 [============================>.] - ETA: 0s - loss: 0.3719 - accuracy: 0.8
350
Epoch 13: saving model to checkpoints_cnn_3\weights_epoch_13.h5
 | CPU: 2.0% | RAM: 15718 MB | Time: 11.5s | GPU: 1.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 11s 46ms/step - loss: 0.3716 - accurac
y: 0.8351 - val_loss: 0.4223 - val_accuracy: 0.8130
Epoch 14/20
249/250 [============================>.] - ETA: 0s - loss: 0.3586 - accuracy: 0.8
414
Epoch 14: saving model to checkpoints_cnn_3\weights_epoch_14.h5
 | CPU: 2.2% | RAM: 15705 MB | Time: 11.5s | GPU: 5.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.3585 - accurac
y: 0.8413 - val_loss: 0.4069 - val_accuracy: 0.8085
Epoch 15/20
250/250 [==============================] - ETA: 0s - loss: 0.3501 - accuracy: 0.8
439
Epoch 15: saving model to checkpoints_cnn_3\weights_epoch_15.h5
 | CPU: 1.2% | RAM: 15750 MB | Time: 11.7s | GPU: 5.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.3501 - accurac
y: 0.8439 - val_loss: 0.3980 - val_accuracy: 0.8216
Epoch 16/20
249/250 [============================>.] - ETA: 0s - loss: 0.3381 - accuracy: 0.8
478
Epoch 16: saving model to checkpoints_cnn_3\weights_epoch_16.h5
 | CPU: 1.6% | RAM: 15761 MB | Time: 11.5s | GPU: 3.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.3380 - accurac
y: 0.8478 - val_loss: 0.4236 - val_accuracy: 0.8100
Epoch 17/20
249/250 [============================>.] - ETA: 0s - loss: 0.3332 - accuracy: 0.8
564
Epoch 17: saving model to checkpoints_cnn_3\weights_epoch_17.h5
 | CPU: 1.8% | RAM: 15733 MB | Time: 11.5s | GPU: 1.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 11s 46ms/step - loss: 0.3328 - accurac
y: 0.8568 - val_loss: 0.4226 - val_accuracy: 0.8155
Epoch 18/20
```

```
249/250 [============================>.] - ETA: 0s - loss: 0.3142 - accuracy: 0.8
579
Epoch 18: saving model to checkpoints_cnn_3\weights_epoch_18.h5
 | CPU: 2.1% | RAM: 15730 MB | Time: 11.6s | GPU: 1.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 47ms/step - loss: 0.3138 - accurac
y: 0.8583 - val_loss: 0.3847 - val_accuracy: 0.8367
Epoch 19/20
249/250 [============================>.] - ETA: 0s - loss: 0.3071 - accuracy: 0.8
630
Epoch 19: saving model to checkpoints_cnn_3\weights_epoch_19.h5
 | CPU: 1.1% | RAM: 15722 MB | Time: 11.6s | GPU: 3.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.3074 - accurac
y: 0.8629 - val_loss: 0.3756 - val_accuracy: 0.8392
Epoch 20/20
249/250 [============================>.] - ETA: 0s - loss: 0.2918 - accuracy: 0.8
732
Epoch 20: saving model to checkpoints_cnn_3\weights_epoch_20.h5
 | CPU: 2.4% | RAM: 15619 MB | Time: 11.6s | GPU: 3.0% | VRAM: 781.0/16303.0 MB
250/250 [==============================] - 12s 46ms/step - loss: 0.2924 - accurac
y: 0.8733 - val_loss: 0.5256 - val_accuracy: 0.7802
```

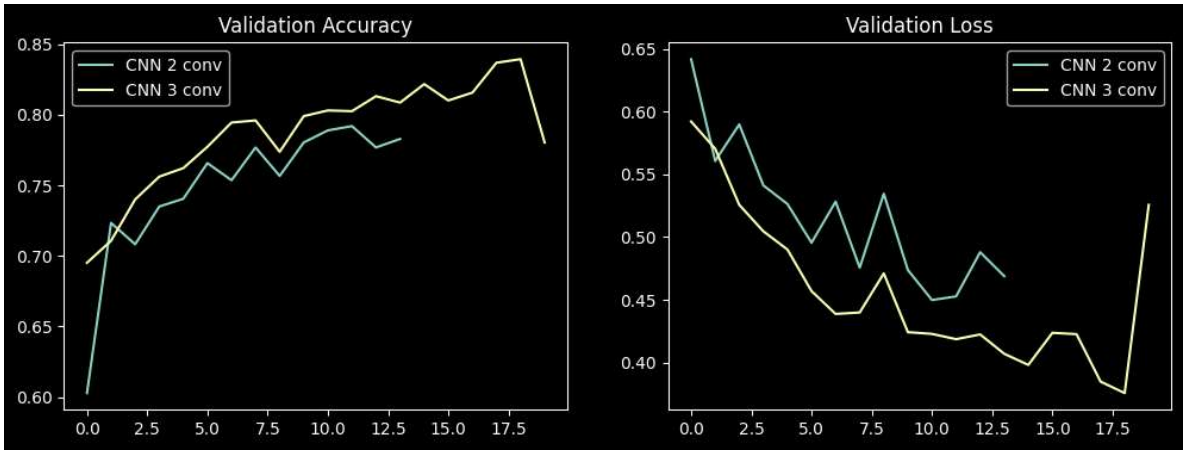
In [50]:
```python
import matplotlib.pyplot as plt

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 (Dogs & Cats)

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 et bien régularisé grâce à l'early stopping.

CNN 2 convolutions :

- Précision maximale sur l'ensemble de validation : ~79,2 %
- Perte de validation minimale : ~0,45
- Early stopping déclenché à l'epoch 14, meilleur modèle restauré à l'epoch 12

CNN 3 convolutions :

- Précision maximale sur l'ensemble de validation : ~83,9 %
- Perte de validation minimale : ~0,38
- Entraînement mené jusqu'à l'epoch 20, avec performances optimales observées entre les epochs 18 et 19

Le graphe de précision de validation montre que le CNN à 3 convolutions converge plus régulièrement et atteint une précision systématiquement supérieure à celle du CNN à 2 convolutions sur la majorité des epochs. De plus, le graphe de perte de validation indique une diminution plus marquée et plus stable pour le modèle à 3 convolutions, traduisant une meilleure capacité de généralisation.

Lors de l'entraînement des modèles CNN à 2 et 3 convolutions, l'utilisation des ressources matérielles reste faible et stable. Le CPU est peu sollicité (environ 1 à 5 %), le GPU présente une utilisation modérée avec quelques pics ponctuels (jusqu'à ~30–40 % en début d'entraînement pour le CNN à 2 convolutions), la RAM reste stable autour de 15 à 16 Go et la VRAM demeure inférieure à 1 Go. Ces observations montrent que l'entraînement n'est pas limité par le matériel et que les performances obtenues sont liées aux choix d'architecture.