Import Req Lib

In [1]:
```
%matplotlib inline


import shutil
import random
import numpy as np
from warnings import filterwarnings
filterwarnings('ignore')

from tensorflow.keras import layers,regularizers, optimizers
from tensorflow.keras import models
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import LeakyReLU,Dense, Activation, Flatten, Dropout, BatchNormalization,Conv2D, MaxPooling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf

import os
import time
import csv

from matplotlib import figure
```

Define 2 worker

In [2]:
```
# Set the number of threads
number_of_worker = 2
os.environ['OMP_NUM_THREADS'] = '2'  # OpenMP threads for parallelism
os.environ['TF_NUM_INTEROP_THREADS'] = '2'  # Threads for inter-operation parallelism
os.environ['TF_NUM_INTRAOP_THREADS'] = '2'  # Threads for intra-operation parallelism

# Confirm TensorFlow is using the specified number of threads
tf.config.threading.set_inter_op_parallelism_threads(number_of_worker)
tf.config.threading.set_intra_op_parallelism_threads(number_of_worker)
```

Train Val data Split

In [3]:
```
source_dir = r"C:\Users\nikhi\OneDrive\Desktop\Final Project\DATA\Convert_Audio_File_to_jpg_file"
target_dir = r'genres_train_val_split_data'
split_ratio = 0.8

def Train_Test_Split(source_dir,target_dir,split_ratio):
    # Define source and target directories
    train_dir = os.path.join(target_dir, 'train')
    val_dir = os.path.join(target_dir, 'val')

    # Create target directories if they don't exist
    os.makedirs(train_dir, exist_ok=True)
    os.makedirs(val_dir, exist_ok=True)

    # Get the list of class directories
    classes = [d for d in os.listdir(source_dir) if os.path.isdir(os.path.join(source_dir, d))]

    for class_name in classes:
        # Create class directories in train and val folders
        os.makedirs(os.path.join(train_dir, class_name), exist_ok=True)
        os.makedirs(os.path.join(val_dir, class_name), exist_ok=True)

        # Get list of images in the class directory
        class_dir = os.path.join(source_dir, class_name)
        images = [f for f in os.listdir(class_dir) if os.path.isfile(os.path.join(class_dir, f))]

        # Shuffle the images
        random.shuffle(images)

        # Compute the split point
        split_point = int(len(images) * split_ratio)

        # Split the images into training and validation sets
        train_images = images[:split_point]
        val_images = images[split_point:]
```

```
        # Move the images to the respective directories
        for img in train_images:
            shutil.copy(os.path.join(class_dir, img), os.path.join(train_dir, class_name, img))

        for img in val_images:
            shutil.copy(os.path.join(class_dir, img), os.path.join(val_dir, class_name, img))

    print("Data split completed successfully!")
```

In [4]:
```
Train_Test_Split(source_dir,target_dir,split_ratio)
```

```
Data split completed successfully!
```

Load the Data

In [5]:
```
WIDTH = 64
HEIGHT = 64
BATCH_SIZE = 32
TRAIN_DIR=r'genres_train_val_split_data/train'
val_dir = r'genres_train_val_split_data/val'

# data prep
train_datagen = ImageDataGenerator(
    rescale=1./255.,validation_split=0.25)


train_generator = train_datagen.flow_from_directory(
    TRAIN_DIR,
    target_size=(HEIGHT, WIDTH),
        batch_size=BATCH_SIZE,
        class_mode='categorical')

validation_gen = train_datagen.flow_from_directory(
    val_dir,target_size = (HEIGHT,WIDTH),
    batch_size = BATCH_SIZE,
    class_mode = 'categorical'
 )
```

```
Found 800 images belonging to 10 classes.
Found 200 images belonging to 10 classes.
```

Model Architecture

In [6]:
```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same',
                 input_shape=(64,64,3)))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))
model.add(Conv2D(128, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))
model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
model.compile(optimizers.RMSprop(learning_rate=0.0005, decay=1e-6),loss="categorical_crossentropy",metrics=["accuracy"])
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 32)	896
activation (Activation)	(None, 64, 64, 32)	0
conv2d_1 (Conv2D)	(None, 62, 62, 64)	18,496
activation_1 (Activation)	(None, 62, 62, 64)	0
max_pooling2d (MaxPooling2D)	(None, 31, 31, 64)	0
dropout (Dropout)	(None, 31, 31, 64)	0
conv2d_2 (Conv2D)	(None, 31, 31, 64)	36,928
activation_2 (Activation)	(None, 31, 31, 64)	0
conv2d_3 (Conv2D)	(None, 29, 29, 64)	36,928
activation_3 (Activation)	(None, 29, 29, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
dropout_1 (Dropout)	(None, 14, 14, 64)	0
conv2d_4 (Conv2D)	(None, 14, 14, 128)	73,856
activation_4 (Activation)	(None, 14, 14, 128)	0
conv2d_5 (Conv2D)	(None, 12, 12, 128)	147,584
activation_5 (Activation)	(None, 12, 12, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_2 (Dropout)	(None, 6, 6, 128)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 512)	2,359,808
activation_6 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 10)	5,130

Total params: 2,679,626 (10.22 MB)

Trainable params: 2,679,626 (10.22 MB)

Non-trainable params: 0 (0.00 B)

In [7]:
```
STEP_SIZE_TRAIN=train_generator.n//train_generator.batch_size
# Measure the execution time
start_time = time.time()

model.fit(train_generator,validation_data=validation_gen,epochs=200)

end_time = time.time()
elapsed_time = end_time - start_time
```

```
Epoch 1/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 20s 675ms/step - accuracy: 0.0658 - loss: 2.3332 - val_accuracy: 0.1000 - val_loss: 2.3021
Epoch 2/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 693ms/step - accuracy: 0.0986 - loss: 2.3021 - val_accuracy: 0.1000 - val_loss: 2.2934
Epoch 3/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 16s 658ms/step - accuracy: 0.1090 - loss: 2.2960 - val_accuracy: 0.2100 - val_loss: 2.2949
Epoch 4/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 16s 652ms/step - accuracy: 0.1633 - loss: 2.2699 - val_accuracy: 0.1900 - val_loss: 2.1670
Epoch 5/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 15s 607ms/step - accuracy: 0.2279 - loss: 2.1263 - val_accuracy: 0.2100 - val_loss: 2.0336
Epoch 6/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 15s 598ms/step - accuracy: 0.2639 - loss: 2.0327 - val_accuracy: 0.2800 - val_loss: 2.0367
Epoch 7/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 15s 595ms/step - accuracy: 0.2955 - loss: 1.9982 - val_accuracy: 0.3250 - val_loss: 1.9601
Epoch 8/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 15s 594ms/step - accuracy: 0.2485 - loss: 2.0284 - val_accuracy: 0.2950 - val_loss: 1.9328
Epoch 9/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 15s 589ms/step - accuracy: 0.2926 - loss: 1.9276 - val_accuracy: 0.3500 - val_loss: 1.9102
Epoch 10/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 15s 602ms/step - accuracy: 0.2897 - loss: 1.9524 - val_accuracy: 0.3400 - val_loss: 1.8158
Epoch 11/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 16s 650ms/step - accuracy: 0.3268 - loss: 1.8249 - val_accuracy: 0.3350 - val_loss: 1.7215
Epoch 12/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 687ms/step - accuracy: 0.3834 - loss: 1.7091 - val_accuracy: 0.3750 - val_loss: 1.6773
Epoch 13/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 700ms/step - accuracy: 0.3712 - loss: 1.7670 - val_accuracy: 0.2500 - val_loss: 2.0661
Epoch 14/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 699ms/step - accuracy: 0.3958 - loss: 1.7096 - val_accuracy: 0.3350 - val_loss: 1.6078
Epoch 15/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.4427 - loss: 1.5865 - val_accuracy: 0.3950 - val_loss: 1.5535
Epoch 16/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 733ms/step - accuracy: 0.4159 - loss: 1.5572 - val_accuracy: 0.3500 - val_loss: 1.8125
Epoch 17/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 19s 741ms/step - accuracy: 0.4032 - loss: 1.6410 - val_accuracy: 0.4050 - val_loss: 1.5667
Epoch 18/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 733ms/step - accuracy: 0.4755 - loss: 1.4892 - val_accuracy: 0.4050 - val_loss: 1.5534
Epoch 19/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 721ms/step - accuracy: 0.4703 - loss: 1.4501 - val_accuracy: 0.4250 - val_loss: 1.5390
Epoch 20/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 715ms/step - accuracy: 0.4298 - loss: 1.5520 - val_accuracy: 0.3700 - val_loss: 1.6879
Epoch 21/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 727ms/step - accuracy: 0.4590 - loss: 1.4803 - val_accuracy: 0.4050 - val_loss: 1.5830
Epoch 22/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.4832 - loss: 1.4678 - val_accuracy: 0.4850 - val_loss: 1.3881
Epoch 23/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 721ms/step - accuracy: 0.5379 - loss: 1.2982 - val_accuracy: 0.4600 - val_loss: 1.4611
Epoch 24/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 734ms/step - accuracy: 0.5154 - loss: 1.3152 - val_accuracy: 0.4250 - val_loss: 1.4089
Epoch 25/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 724ms/step - accuracy: 0.5432 - loss: 1.2727 - val_accuracy: 0.5100 - val_loss: 1.3474
Epoch 26/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 719ms/step - accuracy: 0.5513 - loss: 1.2137 - val_accuracy: 0.5150 - val_loss: 1.3525
Epoch 27/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.5891 - loss: 1.1408 - val_accuracy: 0.5200 - val_loss: 1.3607
Epoch 28/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.5416 - loss: 1.1971 - val_accuracy: 0.4800 - val_loss: 1.4404
Epoch 29/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 738ms/step - accuracy: 0.5836 - loss: 1.1399 - val_accuracy: 0.5000 - val_loss: 1.3671
Epoch 30/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 735ms/step - accuracy: 0.5671 - loss: 1.1366 - val_accuracy: 0.5200 - val_loss: 1.3093
Epoch 31/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 19s 751ms/step - accuracy: 0.6065 - loss: 1.0926 - val_accuracy: 0.5100 - val_loss: 1.2397
Epoch 32/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 737ms/step - accuracy: 0.5776 - loss: 1.0719 - val_accuracy: 0.5250 - val_loss: 1.3024
Epoch 33/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 707ms/step - accuracy: 0.6042 - loss: 1.0833 - val_accuracy: 0.5250 - val_loss: 1.3458
Epoch 34/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 690ms/step - accuracy: 0.5982 - loss: 1.1189 - val_accuracy: 0.5650 - val_loss: 1.2984
Epoch 35/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 715ms/step - accuracy: 0.6693 - loss: 0.9085 - val_accuracy: 0.5400 - val_loss: 1.2657
Epoch 36/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 735ms/step - accuracy: 0.6650 - loss: 0.9373 - val_accuracy: 0.5600 - val_loss: 1.1906
Epoch 37/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 730ms/step - accuracy: 0.6666 - loss: 0.9179 - val_accuracy: 0.5850 - val_loss: 1.1938
Epoch 38/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 735ms/step - accuracy: 0.6960 - loss: 0.8488 - val_accuracy: 0.6050 - val_loss: 1.1426
Epoch 39/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.6970 - loss: 0.8214 - val_accuracy: 0.5650 - val_loss: 1.2680
Epoch 40/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 712ms/step - accuracy: 0.6785 - loss: 0.8729 - val_accuracy: 0.5750 - val_loss: 1.1458
Epoch 41/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 717ms/step - accuracy: 0.6635 - loss: 0.9008 - val_accuracy: 0.6000 - val_loss: 1.1763
```

```
Epoch 42/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 707ms/step - accuracy: 0.6824 - loss: 0.8254 - val_accuracy: 0.5600 - val_loss: 1.1993
Epoch 43/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 734ms/step - accuracy: 0.7287 - loss: 0.8137 - val_accuracy: 0.6000 - val_loss: 1.1503
Epoch 44/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 725ms/step - accuracy: 0.7484 - loss: 0.6776 - val_accuracy: 0.5950 - val_loss: 1.1720
Epoch 45/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 730ms/step - accuracy: 0.7591 - loss: 0.7487 - val_accuracy: 0.5900 - val_loss: 1.2234
Epoch 46/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 708ms/step - accuracy: 0.7634 - loss: 0.6756 - val_accuracy: 0.5950 - val_loss: 1.1382
Epoch 47/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 721ms/step - accuracy: 0.7567 - loss: 0.6095 - val_accuracy: 0.6150 - val_loss: 1.2420
Epoch 48/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 710ms/step - accuracy: 0.7541 - loss: 0.6675 - val_accuracy: 0.6150 - val_loss: 1.2987
Epoch 49/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 735ms/step - accuracy: 0.7979 - loss: 0.5888 - val_accuracy: 0.6400 - val_loss: 1.2228
Epoch 50/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 731ms/step - accuracy: 0.7945 - loss: 0.5587 - val_accuracy: 0.5650 - val_loss: 1.4477
Epoch 51/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 728ms/step - accuracy: 0.8001 - loss: 0.5551 - val_accuracy: 0.6050 - val_loss: 1.4301
Epoch 52/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 728ms/step - accuracy: 0.7869 - loss: 0.6716 - val_accuracy: 0.5750 - val_loss: 1.3105
Epoch 53/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.7919 - loss: 0.5782 - val_accuracy: 0.6300 - val_loss: 1.2621
Epoch 54/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 719ms/step - accuracy: 0.8451 - loss: 0.4805 - val_accuracy: 0.6350 - val_loss: 1.2906
Epoch 55/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 720ms/step - accuracy: 0.8179 - loss: 0.4926 - val_accuracy: 0.6200 - val_loss: 1.4216
Epoch 56/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 725ms/step - accuracy: 0.8320 - loss: 0.4691 - val_accuracy: 0.6300 - val_loss: 1.4858
Epoch 57/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 20s 718ms/step - accuracy: 0.8099 - loss: 0.4724 - val_accuracy: 0.6100 - val_loss: 1.5430
Epoch 58/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 728ms/step - accuracy: 0.8482 - loss: 0.4111 - val_accuracy: 0.5950 - val_loss: 1.5319
Epoch 59/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 708ms/step - accuracy: 0.8596 - loss: 0.3476 - val_accuracy: 0.6100 - val_loss: 1.6917
Epoch 60/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 709ms/step - accuracy: 0.8134 - loss: 0.4908 - val_accuracy: 0.6200 - val_loss: 1.5236
Epoch 61/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 715ms/step - accuracy: 0.8909 - loss: 0.3588 - val_accuracy: 0.6200 - val_loss: 1.4370
Epoch 62/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.8414 - loss: 0.4690 - val_accuracy: 0.6300 - val_loss: 1.7034
Epoch 63/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 21s 716ms/step - accuracy: 0.8408 - loss: 0.5160 - val_accuracy: 0.5900 - val_loss: 1.6704
Epoch 64/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.8854 - loss: 0.3459 - val_accuracy: 0.6050 - val_loss: 1.6405
Epoch 65/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 723ms/step - accuracy: 0.9229 - loss: 0.2495 - val_accuracy: 0.6100 - val_loss: 1.6316
Epoch 66/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 20s 720ms/step - accuracy: 0.8997 - loss: 0.2830 - val_accuracy: 0.6150 - val_loss: 1.5437
Epoch 67/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.9092 - loss: 0.2726 - val_accuracy: 0.6000 - val_loss: 1.7793
Epoch 68/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 715ms/step - accuracy: 0.9390 - loss: 0.1926 - val_accuracy: 0.5800 - val_loss: 2.0801
Epoch 69/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 723ms/step - accuracy: 0.9058 - loss: 0.3264 - val_accuracy: 0.6200 - val_loss: 1.8446
Epoch 70/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 733ms/step - accuracy: 0.8686 - loss: 0.4121 - val_accuracy: 0.6000 - val_loss: 1.9129
Epoch 71/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 730ms/step - accuracy: 0.8806 - loss: 0.3671 - val_accuracy: 0.6050 - val_loss: 1.6417
Epoch 72/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 705ms/step - accuracy: 0.9386 - loss: 0.1723 - val_accuracy: 0.5900 - val_loss: 1.7384
Epoch 73/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 715ms/step - accuracy: 0.9273 - loss: 0.2003 - val_accuracy: 0.5850 - val_loss: 2.1487
Epoch 74/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9156 - loss: 0.2440 - val_accuracy: 0.6000 - val_loss: 1.9785
Epoch 75/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 21s 729ms/step - accuracy: 0.9193 - loss: 0.2718 - val_accuracy: 0.5850 - val_loss: 2.0705
Epoch 76/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 724ms/step - accuracy: 0.9459 - loss: 0.1662 - val_accuracy: 0.6100 - val_loss: 2.0769
Epoch 77/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 726ms/step - accuracy: 0.9097 - loss: 0.2411 - val_accuracy: 0.6250 - val_loss: 2.1611
Epoch 78/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 733ms/step - accuracy: 0.9575 - loss: 0.1431 - val_accuracy: 0.5850 - val_loss: 2.3373
Epoch 79/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.9352 - loss: 0.2195 - val_accuracy: 0.6050 - val_loss: 2.0010
Epoch 80/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 707ms/step - accuracy: 0.9505 - loss: 0.1423 - val_accuracy: 0.5950 - val_loss: 1.8676
Epoch 81/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 708ms/step - accuracy: 0.9476 - loss: 0.1579 - val_accuracy: 0.5950 - val_loss: 2.0369
Epoch 82/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 723ms/step - accuracy: 0.9688 - loss: 0.1092 - val_accuracy: 0.6150 - val_loss: 1.9845
```

```
Epoch 83/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 731ms/step - accuracy: 0.9479 - loss: 0.1606 - val_accuracy: 0.5800 - val_loss: 2.6820
Epoch 84/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 726ms/step - accuracy: 0.9545 - loss: 0.1623 - val_accuracy: 0.6100 - val_loss: 1.9176
Epoch 85/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 706ms/step - accuracy: 0.9567 - loss: 0.1448 - val_accuracy: 0.6100 - val_loss: 2.0825
Epoch 86/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 710ms/step - accuracy: 0.9462 - loss: 0.1407 - val_accuracy: 0.6050 - val_loss: 2.2098
Epoch 87/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 721ms/step - accuracy: 0.9378 - loss: 0.1713 - val_accuracy: 0.5900 - val_loss: 2.2017
Epoch 88/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 716ms/step - accuracy: 0.9610 - loss: 0.1053 - val_accuracy: 0.6200 - val_loss: 2.2934
Epoch 89/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 728ms/step - accuracy: 0.9620 - loss: 0.1244 - val_accuracy: 0.6000 - val_loss: 2.2978
Epoch 90/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.9526 - loss: 0.1334 - val_accuracy: 0.5850 - val_loss: 2.2852
Epoch 91/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 730ms/step - accuracy: 0.9629 - loss: 0.1154 - val_accuracy: 0.6000 - val_loss: 2.0158
Epoch 92/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 708ms/step - accuracy: 0.9730 - loss: 0.0986 - val_accuracy: 0.6100 - val_loss: 2.1760
Epoch 93/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 716ms/step - accuracy: 0.9814 - loss: 0.0740 - val_accuracy: 0.5800 - val_loss: 2.4375
Epoch 94/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 703ms/step - accuracy: 0.9482 - loss: 0.1256 - val_accuracy: 0.6200 - val_loss: 2.1586
Epoch 95/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 715ms/step - accuracy: 0.9786 - loss: 0.0733 - val_accuracy: 0.6100 - val_loss: 2.1589
Epoch 96/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 731ms/step - accuracy: 0.9680 - loss: 0.0907 - val_accuracy: 0.5900 - val_loss: 2.4064
Epoch 97/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 19s 754ms/step - accuracy: 0.9696 - loss: 0.0912 - val_accuracy: 0.5600 - val_loss: 2.4183
Epoch 98/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 733ms/step - accuracy: 0.9788 - loss: 0.0655 - val_accuracy: 0.6050 - val_loss: 2.2126
Epoch 99/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 20s 722ms/step - accuracy: 0.9470 - loss: 0.1776 - val_accuracy: 0.6300 - val_loss: 2.1396
Epoch 100/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 709ms/step - accuracy: 0.9711 - loss: 0.0763 - val_accuracy: 0.6050 - val_loss: 2.0356
Epoch 101/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 724ms/step - accuracy: 0.9650 - loss: 0.1307 - val_accuracy: 0.6050 - val_loss: 2.1148
Epoch 102/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 724ms/step - accuracy: 0.9790 - loss: 0.0682 - val_accuracy: 0.6000 - val_loss: 2.6045
Epoch 103/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.9743 - loss: 0.0784 - val_accuracy: 0.5750 - val_loss: 2.6396
Epoch 104/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 711ms/step - accuracy: 0.9579 - loss: 0.1212 - val_accuracy: 0.5750 - val_loss: 2.9462
Epoch 105/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 719ms/step - accuracy: 0.9851 - loss: 0.0671 - val_accuracy: 0.6000 - val_loss: 2.5684
Epoch 106/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 726ms/step - accuracy: 0.9438 - loss: 0.2108 - val_accuracy: 0.5700 - val_loss: 2.1696
Epoch 107/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 720ms/step - accuracy: 0.9562 - loss: 0.1139 - val_accuracy: 0.5850 - val_loss: 2.4554
Epoch 108/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 724ms/step - accuracy: 0.9572 - loss: 0.1481 - val_accuracy: 0.6150 - val_loss: 2.1096
Epoch 109/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.9787 - loss: 0.0589 - val_accuracy: 0.5750 - val_loss: 3.2846
Epoch 110/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9564 - loss: 0.1865 - val_accuracy: 0.6050 - val_loss: 2.1813
Epoch 111/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 717ms/step - accuracy: 0.9763 - loss: 0.0739 - val_accuracy: 0.5700 - val_loss: 2.6695
Epoch 112/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 712ms/step - accuracy: 0.9603 - loss: 0.1191 - val_accuracy: 0.6050 - val_loss: 2.3246
Epoch 113/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.9768 - loss: 0.0602 - val_accuracy: 0.6150 - val_loss: 2.4997
Epoch 114/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 696ms/step - accuracy: 0.9840 - loss: 0.0772 - val_accuracy: 0.6000 - val_loss: 2.7861
Epoch 115/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 716ms/step - accuracy: 0.9773 - loss: 0.1082 - val_accuracy: 0.6150 - val_loss: 2.3787
Epoch 116/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 720ms/step - accuracy: 0.9836 - loss: 0.0524 - val_accuracy: 0.6000 - val_loss: 2.9112
Epoch 117/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 721ms/step - accuracy: 0.9603 - loss: 0.1101 - val_accuracy: 0.5900 - val_loss: 2.7486
Epoch 118/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 710ms/step - accuracy: 0.9809 - loss: 0.0785 - val_accuracy: 0.5950 - val_loss: 2.8456
Epoch 119/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 702ms/step - accuracy: 0.9673 - loss: 0.1098 - val_accuracy: 0.5950 - val_loss: 2.6416
Epoch 120/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 701ms/step - accuracy: 0.9882 - loss: 0.0370 - val_accuracy: 0.6100 - val_loss: 2.6769
Epoch 121/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 712ms/step - accuracy: 0.9638 - loss: 0.1044 - val_accuracy: 0.6200 - val_loss: 2.9083
Epoch 122/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 725ms/step - accuracy: 0.9782 - loss: 0.0979 - val_accuracy: 0.6200 - val_loss: 2.5686
Epoch 123/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 729ms/step - accuracy: 0.9812 - loss: 0.0578 - val_accuracy: 0.6250 - val_loss: 2.3114
```

```
Epoch 124/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 716ms/step - accuracy: 0.9868 - loss: 0.0367 - val_accuracy: 0.5950 - val_loss: 2.8347
Epoch 125/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 702ms/step - accuracy: 0.9719 - loss: 0.0724 - val_accuracy: 0.6250 - val_loss: 2.6391
Epoch 126/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9764 - loss: 0.0920 - val_accuracy: 0.5950 - val_loss: 2.5164
Epoch 127/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 709ms/step - accuracy: 0.9781 - loss: 0.0665 - val_accuracy: 0.6250 - val_loss: 2.5423
Epoch 128/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 705ms/step - accuracy: 0.9931 - loss: 0.0285 - val_accuracy: 0.5950 - val_loss: 2.9967
Epoch 129/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.9756 - loss: 0.0772 - val_accuracy: 0.5450 - val_loss: 3.1431
Epoch 130/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 738ms/step - accuracy: 0.9635 - loss: 0.0942 - val_accuracy: 0.6100 - val_loss: 2.8132
Epoch 131/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 711ms/step - accuracy: 0.9839 - loss: 0.0396 - val_accuracy: 0.5900 - val_loss: 2.7533
Epoch 132/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 720ms/step - accuracy: 0.9689 - loss: 0.1105 - val_accuracy: 0.6100 - val_loss: 2.5281
Epoch 133/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 710ms/step - accuracy: 0.9722 - loss: 0.0740 - val_accuracy: 0.5850 - val_loss: 3.1726
Epoch 134/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9883 - loss: 0.0474 - val_accuracy: 0.6200 - val_loss: 2.6395
Epoch 135/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 707ms/step - accuracy: 0.9730 - loss: 0.0793 - val_accuracy: 0.5900 - val_loss: 2.8316
Epoch 136/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 728ms/step - accuracy: 0.9670 - loss: 0.0725 - val_accuracy: 0.6050 - val_loss: 2.2946
Epoch 137/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 715ms/step - accuracy: 0.9915 - loss: 0.0330 - val_accuracy: 0.6050 - val_loss: 2.6592
Epoch 138/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 717ms/step - accuracy: 0.9781 - loss: 0.0669 - val_accuracy: 0.6100 - val_loss: 2.7602
Epoch 139/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 708ms/step - accuracy: 0.9688 - loss: 0.1116 - val_accuracy: 0.6100 - val_loss: 3.3575
Epoch 140/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 691ms/step - accuracy: 0.9746 - loss: 0.1208 - val_accuracy: 0.6200 - val_loss: 2.8369
Epoch 141/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 702ms/step - accuracy: 0.9881 - loss: 0.0348 - val_accuracy: 0.6050 - val_loss: 2.4070
Epoch 142/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 716ms/step - accuracy: 0.9784 - loss: 0.0589 - val_accuracy: 0.5900 - val_loss: 2.8224
Epoch 143/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9804 - loss: 0.0618 - val_accuracy: 0.6000 - val_loss: 2.9480
Epoch 144/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 727ms/step - accuracy: 0.9570 - loss: 0.1407 - val_accuracy: 0.6050 - val_loss: 3.0398
Epoch 145/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.9857 - loss: 0.0526 - val_accuracy: 0.6100 - val_loss: 2.8026
Epoch 146/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 704ms/step - accuracy: 0.9799 - loss: 0.0990 - val_accuracy: 0.5900 - val_loss: 2.9637
Epoch 147/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 711ms/step - accuracy: 0.9758 - loss: 0.0744 - val_accuracy: 0.6050 - val_loss: 2.6072
Epoch 148/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9780 - loss: 0.0572 - val_accuracy: 0.5950 - val_loss: 2.8387
Epoch 149/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 729ms/step - accuracy: 0.9769 - loss: 0.0605 - val_accuracy: 0.5850 - val_loss: 2.7072
Epoch 150/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.9864 - loss: 0.0305 - val_accuracy: 0.6000 - val_loss: 3.7635
Epoch 151/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 726ms/step - accuracy: 0.9730 - loss: 0.1296 - val_accuracy: 0.5750 - val_loss: 2.9057
Epoch 152/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.9956 - loss: 0.0176 - val_accuracy: 0.6300 - val_loss: 2.9177
Epoch 153/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 721ms/step - accuracy: 0.9782 - loss: 0.0799 - val_accuracy: 0.5900 - val_loss: 2.9872
Epoch 154/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 728ms/step - accuracy: 0.9934 - loss: 0.0160 - val_accuracy: 0.5900 - val_loss: 3.0310
Epoch 155/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 19s 762ms/step - accuracy: 0.9743 - loss: 0.0986 - val_accuracy: 0.5950 - val_loss: 2.8097
Epoch 156/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 684ms/step - accuracy: 0.9687 - loss: 0.1124 - val_accuracy: 0.6150 - val_loss: 2.6391
Epoch 157/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.9891 - loss: 0.0306 - val_accuracy: 0.6150 - val_loss: 2.9222
Epoch 158/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 685ms/step - accuracy: 0.9726 - loss: 0.0740 - val_accuracy: 0.6000 - val_loss: 2.8231
Epoch 159/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 706ms/step - accuracy: 0.9903 - loss: 0.0341 - val_accuracy: 0.5800 - val_loss: 3.2725
Epoch 160/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 729ms/step - accuracy: 0.9856 - loss: 0.0550 - val_accuracy: 0.6050 - val_loss: 3.0906
Epoch 161/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 708ms/step - accuracy: 0.9797 - loss: 0.0624 - val_accuracy: 0.5900 - val_loss: 2.7295
Epoch 162/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 731ms/step - accuracy: 0.9947 - loss: 0.0261 - val_accuracy: 0.6050 - val_loss: 2.9180
Epoch 163/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 736ms/step - accuracy: 0.9889 - loss: 0.0323 - val_accuracy: 0.6050 - val_loss: 2.9152
Epoch 164/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 19s 741ms/step - accuracy: 0.9805 - loss: 0.0453 - val_accuracy: 0.6100 - val_loss: 2.8851
```

```
Epoch 165/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 726ms/step - accuracy: 0.9907 - loss: 0.0331 - val_accuracy: 0.6150 - val_loss: 3.0317
Epoch 166/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 688ms/step - accuracy: 0.9721 - loss: 0.0709 - val_accuracy: 0.6100 - val_loss: 3.0954
Epoch 167/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 713ms/step - accuracy: 0.9757 - loss: 0.0530 - val_accuracy: 0.6050 - val_loss: 3.2853
Epoch 168/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 706ms/step - accuracy: 0.9767 - loss: 0.0868 - val_accuracy: 0.6100 - val_loss: 2.8212
Epoch 169/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 721ms/step - accuracy: 0.9894 - loss: 0.0469 - val_accuracy: 0.6000 - val_loss: 3.0674
Epoch 170/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.9883 - loss: 0.0524 - val_accuracy: 0.5900 - val_loss: 2.7765
Epoch 171/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9719 - loss: 0.0858 - val_accuracy: 0.5950 - val_loss: 2.9619
Epoch 172/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 723ms/step - accuracy: 0.9967 - loss: 0.0159 - val_accuracy: 0.5950 - val_loss: 3.2934
Epoch 173/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 702ms/step - accuracy: 0.9748 - loss: 0.0891 - val_accuracy: 0.5900 - val_loss: 2.9908
Epoch 174/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 694ms/step - accuracy: 0.9945 - loss: 0.0353 - val_accuracy: 0.6050 - val_loss: 3.2003
Epoch 175/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 710ms/step - accuracy: 0.9945 - loss: 0.0238 - val_accuracy: 0.5550 - val_loss: 5.2057
Epoch 176/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 728ms/step - accuracy: 0.9718 - loss: 0.1247 - val_accuracy: 0.5800 - val_loss: 3.1216
Epoch 177/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.9788 - loss: 0.0614 - val_accuracy: 0.6100 - val_loss: 3.3939
Epoch 178/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 729ms/step - accuracy: 0.9917 - loss: 0.0176 - val_accuracy: 0.5800 - val_loss: 3.5900
Epoch 179/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 720ms/step - accuracy: 0.9924 - loss: 0.0369 - val_accuracy: 0.6100 - val_loss: 3.5684
Epoch 180/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 716ms/step - accuracy: 0.9919 - loss: 0.0242 - val_accuracy: 0.6100 - val_loss: 3.8176
Epoch 181/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 703ms/step - accuracy: 0.9974 - loss: 0.0112 - val_accuracy: 0.6200 - val_loss: 3.5469
Epoch 182/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 722ms/step - accuracy: 0.9864 - loss: 0.0510 - val_accuracy: 0.6000 - val_loss: 3.8091
Epoch 183/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 723ms/step - accuracy: 0.9632 - loss: 0.1131 - val_accuracy: 0.5950 - val_loss: 3.4164
Epoch 184/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 732ms/step - accuracy: 0.9814 - loss: 0.1048 - val_accuracy: 0.5850 - val_loss: 2.7835
Epoch 185/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.9960 - loss: 0.0176 - val_accuracy: 0.5900 - val_loss: 3.4109
Epoch 186/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 710ms/step - accuracy: 0.9937 - loss: 0.0333 - val_accuracy: 0.5950 - val_loss: 3.6571
Epoch 187/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 681ms/step - accuracy: 0.9916 - loss: 0.0364 - val_accuracy: 0.6050 - val_loss: 3.6589
Epoch 188/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 714ms/step - accuracy: 0.9962 - loss: 0.0138 - val_accuracy: 0.6000 - val_loss: 3.2003
Epoch 189/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 723ms/step - accuracy: 0.9920 - loss: 0.0247 - val_accuracy: 0.5800 - val_loss: 3.8158
Epoch 190/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 718ms/step - accuracy: 0.9831 - loss: 0.0737 - val_accuracy: 0.6250 - val_loss: 3.4735
Epoch 191/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 719ms/step - accuracy: 0.9852 - loss: 0.0763 - val_accuracy: 0.6000 - val_loss: 3.5289
Epoch 192/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 706ms/step - accuracy: 0.9861 - loss: 0.0350 - val_accuracy: 0.6000 - val_loss: 3.7007
Epoch 193/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 697ms/step - accuracy: 0.9921 - loss: 0.0248 - val_accuracy: 0.6050 - val_loss: 3.6667
Epoch 194/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 708ms/step - accuracy: 0.9772 - loss: 0.0782 - val_accuracy: 0.6200 - val_loss: 4.1763
Epoch 195/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 20s 703ms/step - accuracy: 0.9855 - loss: 0.0633 - val_accuracy: 0.6000 - val_loss: 3.3390
Epoch 196/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 710ms/step - accuracy: 0.9920 - loss: 0.0385 - val_accuracy: 0.5800 - val_loss: 5.2956
Epoch 197/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 703ms/step - accuracy: 0.9785 - loss: 0.0978 - val_accuracy: 0.5950 - val_loss: 4.0987
Epoch 198/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 729ms/step - accuracy: 0.9750 - loss: 0.0530 - val_accuracy: 0.6100 - val_loss: 4.0447
Epoch 199/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 18s 705ms/step - accuracy: 0.9891 - loss: 0.0457 - val_accuracy: 0.5850 - val_loss: 3.7655
Epoch 200/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 17s 695ms/step - accuracy: 0.9887 - loss: 0.0432 - val_accuracy: 0.5750 - val_loss: 4.1484
```

In [8]:
```
print(f"Execution time: {elapsed_time:.2f} seconds")
```

```
Execution time: 3590.61 seconds
```

In [9]:
```
def append_core_data(score_path, num_cores, elapsed_time):
    # Check if the file already exists
    file_exists = os.path.exists(score_path)

    # Open the file in append mode
    with open(score_path, mode='a', newline='') as file:
```

```
        writer = csv.writer(file)

        # If the file is new, write the header
        if not file_exists:
            writer.writerow(["Number of Cores", "Elapsed Time"])

        # Write the new data
        writer.writerow([num_cores, elapsed_time])
```

In [10]:
```
score_path = r"C:\Users\nikhi\OneDrive\Desktop\Final Project\DEEP LEARNING WITH HPSC\core_data.txt"
append_core_data(score_path, number_of_worker, elapsed_time)
```