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 7 worker

In [2]:
```
# Set the number of threads
number_of_worker = 7
os.environ['OMP_NUM_THREADS'] = '7'  # OpenMP threads for parallelism
os.environ['TF_NUM_INTEROP_THREADS'] = '7'  # Threads for inter-operation parallelism
os.environ['TF_NUM_INTRAOP_THREADS'] = '7'  # 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 ━━━━━━━━━━━━━━━━━━━━ 4s 123ms/step - accuracy: 0.0818 - loss: 2.3262 - val_accuracy: 0.1050 - val_loss: 2.2680
Epoch 2/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.1573 - loss: 2.2889 - val_accuracy: 0.2100 - val_loss: 2.0738
Epoch 3/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 149ms/step - accuracy: 0.1911 - loss: 2.1370 - val_accuracy: 0.2250 - val_loss: 2.1721
Epoch 4/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 145ms/step - accuracy: 0.2274 - loss: 2.1387 - val_accuracy: 0.2350 - val_loss: 2.0582
Epoch 5/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 149ms/step - accuracy: 0.2377 - loss: 2.0395 - val_accuracy: 0.3100 - val_loss: 1.9587
Epoch 6/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.2534 - loss: 1.9749 - val_accuracy: 0.3600 - val_loss: 2.0144
Epoch 7/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.2613 - loss: 1.9955 - val_accuracy: 0.3100 - val_loss: 1.8442
Epoch 8/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 149ms/step - accuracy: 0.2949 - loss: 1.8974 - val_accuracy: 0.3950 - val_loss: 1.8752
Epoch 9/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.3669 - loss: 1.8716 - val_accuracy: 0.4100 - val_loss: 1.8567
Epoch 10/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.3370 - loss: 1.8430 - val_accuracy: 0.4150 - val_loss: 1.7267
Epoch 11/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.3315 - loss: 1.8672 - val_accuracy: 0.3900 - val_loss: 1.6877
Epoch 12/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.4074 - loss: 1.7262 - val_accuracy: 0.4200 - val_loss: 1.7082
Epoch 13/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.3605 - loss: 1.7425 - val_accuracy: 0.4300 - val_loss: 1.6156
Epoch 14/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.3964 - loss: 1.6583 - val_accuracy: 0.4500 - val_loss: 1.5646
Epoch 15/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.3550 - loss: 1.7073 - val_accuracy: 0.4550 - val_loss: 1.5227
Epoch 16/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.4600 - loss: 1.5362 - val_accuracy: 0.4300 - val_loss: 1.5923
Epoch 17/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.4731 - loss: 1.5131 - val_accuracy: 0.4800 - val_loss: 1.4602
Epoch 18/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.4707 - loss: 1.4784 - val_accuracy: 0.4700 - val_loss: 1.4989
Epoch 19/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 147ms/step - accuracy: 0.5048 - loss: 1.4154 - val_accuracy: 0.4900 - val_loss: 1.4474
Epoch 20/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.5050 - loss: 1.4192 - val_accuracy: 0.4300 - val_loss: 1.5239
Epoch 21/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.4976 - loss: 1.4264 - val_accuracy: 0.5150 - val_loss: 1.3873
Epoch 22/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.5021 - loss: 1.3814 - val_accuracy: 0.5700 - val_loss: 1.3150
Epoch 23/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.5420 - loss: 1.3249 - val_accuracy: 0.5050 - val_loss: 1.3644
Epoch 24/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.5389 - loss: 1.2833 - val_accuracy: 0.5650 - val_loss: 1.2784
Epoch 25/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.5266 - loss: 1.2446 - val_accuracy: 0.5300 - val_loss: 1.3330
Epoch 26/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.5608 - loss: 1.2418 - val_accuracy: 0.5250 - val_loss: 1.3601
Epoch 27/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.6144 - loss: 1.1044 - val_accuracy: 0.5150 - val_loss: 1.3303
Epoch 28/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.5713 - loss: 1.1832 - val_accuracy: 0.5700 - val_loss: 1.2238
Epoch 29/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.5994 - loss: 1.0920 - val_accuracy: 0.5350 - val_loss: 1.3166
Epoch 30/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.6277 - loss: 1.0387 - val_accuracy: 0.5650 - val_loss: 1.3302
Epoch 31/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.6816 - loss: 0.9319 - val_accuracy: 0.5200 - val_loss: 1.3509
Epoch 32/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.6372 - loss: 1.0041 - val_accuracy: 0.5250 - val_loss: 1.3733
Epoch 33/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.6570 - loss: 0.9495 - val_accuracy: 0.6100 - val_loss: 1.1951
Epoch 34/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.7004 - loss: 0.8949 - val_accuracy: 0.5900 - val_loss: 1.2032
Epoch 35/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.7214 - loss: 0.7897 - val_accuracy: 0.5300 - val_loss: 1.3989
Epoch 36/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.6723 - loss: 0.9051 - val_accuracy: 0.5700 - val_loss: 1.3947
Epoch 37/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.7025 - loss: 0.8452 - val_accuracy: 0.5950 - val_loss: 1.4084
Epoch 38/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.7058 - loss: 0.8214 - val_accuracy: 0.6000 - val_loss: 1.3120
Epoch 39/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.7523 - loss: 0.7262 - val_accuracy: 0.6000 - val_loss: 1.3909
Epoch 40/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.7442 - loss: 0.6782 - val_accuracy: 0.6300 - val_loss: 1.2302
Epoch 41/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.7542 - loss: 0.6868 - val_accuracy: 0.5750 - val_loss: 1.3111
```

```
Epoch 42/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.7372 - loss: 0.7429 - val_accuracy: 0.6000 - val_loss: 1.3485
Epoch 43/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.7675 - loss: 0.6476 - val_accuracy: 0.5900 - val_loss: 1.5109
Epoch 44/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.7961 - loss: 0.6001 - val_accuracy: 0.6050 - val_loss: 1.4317
Epoch 45/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.8014 - loss: 0.5804 - val_accuracy: 0.6100 - val_loss: 1.2404
Epoch 46/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.7928 - loss: 0.5734 - val_accuracy: 0.6250 - val_loss: 1.4807
Epoch 47/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.8146 - loss: 0.5881 - val_accuracy: 0.6300 - val_loss: 1.3544
Epoch 48/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.8335 - loss: 0.4861 - val_accuracy: 0.6150 - val_loss: 1.5126
Epoch 49/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.8099 - loss: 0.5352 - val_accuracy: 0.5750 - val_loss: 1.4304
Epoch 50/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.8612 - loss: 0.4229 - val_accuracy: 0.5750 - val_loss: 1.5456
Epoch 51/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.8352 - loss: 0.5294 - val_accuracy: 0.5600 - val_loss: 1.8977
Epoch 52/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.8287 - loss: 0.4984 - val_accuracy: 0.6050 - val_loss: 1.4981
Epoch 53/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.8452 - loss: 0.4254 - val_accuracy: 0.6400 - val_loss: 1.5920
Epoch 54/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.8350 - loss: 0.4264 - val_accuracy: 0.6150 - val_loss: 1.4441
Epoch 55/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.9038 - loss: 0.3091 - val_accuracy: 0.6400 - val_loss: 1.4488
Epoch 56/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.8716 - loss: 0.3435 - val_accuracy: 0.6200 - val_loss: 1.4544
Epoch 57/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.8840 - loss: 0.3251 - val_accuracy: 0.6000 - val_loss: 1.4671
Epoch 58/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.8853 - loss: 0.3201 - val_accuracy: 0.6200 - val_loss: 1.4904
Epoch 59/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.8687 - loss: 0.3920 - val_accuracy: 0.5850 - val_loss: 1.6867
Epoch 60/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.8751 - loss: 0.3240 - val_accuracy: 0.6350 - val_loss: 1.5783
Epoch 61/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.8843 - loss: 0.3223 - val_accuracy: 0.6150 - val_loss: 1.7564
Epoch 62/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.8901 - loss: 0.3179 - val_accuracy: 0.6050 - val_loss: 1.6935
Epoch 63/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9084 - loss: 0.2651 - val_accuracy: 0.6450 - val_loss: 1.5367
Epoch 64/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 159ms/step - accuracy: 0.9011 - loss: 0.2681 - val_accuracy: 0.6550 - val_loss: 1.5078
Epoch 65/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9111 - loss: 0.2012 - val_accuracy: 0.6550 - val_loss: 1.7399
Epoch 66/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9056 - loss: 0.2551 - val_accuracy: 0.6450 - val_loss: 1.7057
Epoch 67/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9123 - loss: 0.2318 - val_accuracy: 0.6200 - val_loss: 1.7583
Epoch 68/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.9229 - loss: 0.2184 - val_accuracy: 0.6200 - val_loss: 1.7297
Epoch 69/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9322 - loss: 0.1998 - val_accuracy: 0.6450 - val_loss: 1.6918
Epoch 70/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9346 - loss: 0.1579 - val_accuracy: 0.6300 - val_loss: 1.9413
Epoch 71/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 149ms/step - accuracy: 0.9307 - loss: 0.1901 - val_accuracy: 0.6100 - val_loss: 1.6549
Epoch 72/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.8986 - loss: 0.2894 - val_accuracy: 0.6350 - val_loss: 1.7343
Epoch 73/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9419 - loss: 0.1680 - val_accuracy: 0.6350 - val_loss: 1.8229
Epoch 74/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9522 - loss: 0.1810 - val_accuracy: 0.6350 - val_loss: 1.8504
Epoch 75/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9320 - loss: 0.2007 - val_accuracy: 0.5950 - val_loss: 1.8805
Epoch 76/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9319 - loss: 0.1726 - val_accuracy: 0.6050 - val_loss: 2.3080
Epoch 77/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9378 - loss: 0.1729 - val_accuracy: 0.5850 - val_loss: 2.5177
Epoch 78/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9300 - loss: 0.2347 - val_accuracy: 0.6300 - val_loss: 1.8858
Epoch 79/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.9570 - loss: 0.1181 - val_accuracy: 0.6500 - val_loss: 1.7856
Epoch 80/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9564 - loss: 0.1307 - val_accuracy: 0.6600 - val_loss: 1.9016
Epoch 81/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9573 - loss: 0.1220 - val_accuracy: 0.6350 - val_loss: 2.2473
Epoch 82/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 147ms/step - accuracy: 0.9390 - loss: 0.1833 - val_accuracy: 0.6300 - val_loss: 1.9236
```

```
Epoch 83/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.9190 - loss: 0.2433 - val_accuracy: 0.6200 - val_loss: 1.9956
Epoch 84/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.9646 - loss: 0.0943 - val_accuracy: 0.6400 - val_loss: 1.9079
Epoch 85/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9253 - loss: 0.2035 - val_accuracy: 0.6650 - val_loss: 1.9023
Epoch 86/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9570 - loss: 0.1210 - val_accuracy: 0.6300 - val_loss: 2.2414
Epoch 87/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9634 - loss: 0.1130 - val_accuracy: 0.6800 - val_loss: 1.7807
Epoch 88/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9449 - loss: 0.1721 - val_accuracy: 0.5950 - val_loss: 2.3276
Epoch 89/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 158ms/step - accuracy: 0.9632 - loss: 0.1476 - val_accuracy: 0.6400 - val_loss: 1.9755
Epoch 90/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 146ms/step - accuracy: 0.9655 - loss: 0.1029 - val_accuracy: 0.6450 - val_loss: 2.1367
Epoch 91/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 147ms/step - accuracy: 0.9527 - loss: 0.1291 - val_accuracy: 0.6400 - val_loss: 2.1940
Epoch 92/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.9660 - loss: 0.0966 - val_accuracy: 0.6100 - val_loss: 2.1825
Epoch 93/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.9590 - loss: 0.1117 - val_accuracy: 0.6500 - val_loss: 2.4047
Epoch 94/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9492 - loss: 0.1480 - val_accuracy: 0.6200 - val_loss: 2.1937
Epoch 95/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9725 - loss: 0.0931 - val_accuracy: 0.5950 - val_loss: 2.4867
Epoch 96/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9345 - loss: 0.1980 - val_accuracy: 0.5950 - val_loss: 2.5017
Epoch 97/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9536 - loss: 0.1214 - val_accuracy: 0.6600 - val_loss: 2.3244
Epoch 98/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9634 - loss: 0.1116 - val_accuracy: 0.6050 - val_loss: 2.7636
Epoch 99/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9595 - loss: 0.1351 - val_accuracy: 0.6100 - val_loss: 2.1780
Epoch 100/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9683 - loss: 0.0903 - val_accuracy: 0.6450 - val_loss: 2.2073
Epoch 101/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.9749 - loss: 0.0874 - val_accuracy: 0.6300 - val_loss: 2.6755
Epoch 102/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.9796 - loss: 0.0895 - val_accuracy: 0.6300 - val_loss: 2.2189
Epoch 103/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 149ms/step - accuracy: 0.9510 - loss: 0.1476 - val_accuracy: 0.5800 - val_loss: 2.4473
Epoch 104/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.9651 - loss: 0.1142 - val_accuracy: 0.6400 - val_loss: 2.1473
Epoch 105/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9715 - loss: 0.1023 - val_accuracy: 0.5900 - val_loss: 2.3329
Epoch 106/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9530 - loss: 0.1820 - val_accuracy: 0.6000 - val_loss: 2.7490
Epoch 107/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9564 - loss: 0.1376 - val_accuracy: 0.6500 - val_loss: 2.3251
Epoch 108/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9596 - loss: 0.1519 - val_accuracy: 0.6550 - val_loss: 2.4093
Epoch 109/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9653 - loss: 0.0969 - val_accuracy: 0.5850 - val_loss: 2.8712
Epoch 110/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.9487 - loss: 0.1415 - val_accuracy: 0.6350 - val_loss: 2.0791
Epoch 111/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9632 - loss: 0.1062 - val_accuracy: 0.6350 - val_loss: 2.1070
Epoch 112/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9727 - loss: 0.0791 - val_accuracy: 0.6250 - val_loss: 2.1753
Epoch 113/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.9769 - loss: 0.0716 - val_accuracy: 0.6500 - val_loss: 2.0458
Epoch 114/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9788 - loss: 0.0582 - val_accuracy: 0.6400 - val_loss: 2.2886
Epoch 115/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9330 - loss: 0.2581 - val_accuracy: 0.6450 - val_loss: 2.3485
Epoch 116/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9665 - loss: 0.1004 - val_accuracy: 0.6450 - val_loss: 2.5679
Epoch 117/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.9841 - loss: 0.0572 - val_accuracy: 0.6350 - val_loss: 2.4322
Epoch 118/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9653 - loss: 0.0872 - val_accuracy: 0.6350 - val_loss: 2.3860
Epoch 119/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9739 - loss: 0.0839 - val_accuracy: 0.6250 - val_loss: 2.6356
Epoch 120/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 147ms/step - accuracy: 0.9751 - loss: 0.0728 - val_accuracy: 0.6400 - val_loss: 2.5431
Epoch 121/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.9766 - loss: 0.0772 - val_accuracy: 0.6450 - val_loss: 2.2422
Epoch 122/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9807 - loss: 0.0660 - val_accuracy: 0.6250 - val_loss: 2.4464
Epoch 123/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9760 - loss: 0.0668 - val_accuracy: 0.6100 - val_loss: 2.8307
```

```
Epoch 124/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9725 - loss: 0.0801 - val_accuracy: 0.6250 - val_loss: 2.3358
Epoch 125/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9702 - loss: 0.1228 - val_accuracy: 0.6500 - val_loss: 2.3098
Epoch 126/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9603 - loss: 0.1196 - val_accuracy: 0.6550 - val_loss: 2.4535
Epoch 127/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9828 - loss: 0.0520 - val_accuracy: 0.6400 - val_loss: 2.5397
Epoch 128/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9824 - loss: 0.0516 - val_accuracy: 0.6250 - val_loss: 2.7237
Epoch 129/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9866 - loss: 0.0471 - val_accuracy: 0.6250 - val_loss: 2.4941
Epoch 130/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9766 - loss: 0.0721 - val_accuracy: 0.6350 - val_loss: 2.6602
Epoch 131/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9856 - loss: 0.0327 - val_accuracy: 0.6400 - val_loss: 3.0325
Epoch 132/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9794 - loss: 0.0530 - val_accuracy: 0.6200 - val_loss: 2.9336
Epoch 133/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9736 - loss: 0.0823 - val_accuracy: 0.6100 - val_loss: 2.8365
Epoch 134/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9675 - loss: 0.1192 - val_accuracy: 0.6150 - val_loss: 2.5089
Epoch 135/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9666 - loss: 0.1056 - val_accuracy: 0.6550 - val_loss: 2.5584
Epoch 136/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9863 - loss: 0.0400 - val_accuracy: 0.6100 - val_loss: 2.6453
Epoch 137/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.9780 - loss: 0.0677 - val_accuracy: 0.6500 - val_loss: 2.3392
Epoch 138/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9820 - loss: 0.0461 - val_accuracy: 0.6400 - val_loss: 2.6747
Epoch 139/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9669 - loss: 0.1188 - val_accuracy: 0.6150 - val_loss: 2.7924
Epoch 140/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9767 - loss: 0.0923 - val_accuracy: 0.6300 - val_loss: 2.4425
Epoch 141/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9895 - loss: 0.0273 - val_accuracy: 0.6000 - val_loss: 2.8783
Epoch 142/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 159ms/step - accuracy: 0.9818 - loss: 0.0588 - val_accuracy: 0.6200 - val_loss: 3.2892
Epoch 143/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9920 - loss: 0.0224 - val_accuracy: 0.6350 - val_loss: 2.8748
Epoch 144/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9902 - loss: 0.0372 - val_accuracy: 0.6450 - val_loss: 2.9861
Epoch 145/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 149ms/step - accuracy: 0.9819 - loss: 0.0749 - val_accuracy: 0.6250 - val_loss: 2.6517
Epoch 146/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9738 - loss: 0.0858 - val_accuracy: 0.6450 - val_loss: 2.7020
Epoch 147/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9814 - loss: 0.0641 - val_accuracy: 0.6250 - val_loss: 2.7762
Epoch 148/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9870 - loss: 0.0443 - val_accuracy: 0.5850 - val_loss: 3.0334
Epoch 149/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9825 - loss: 0.0449 - val_accuracy: 0.6450 - val_loss: 2.9334
Epoch 150/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9744 - loss: 0.1010 - val_accuracy: 0.6250 - val_loss: 2.8921
Epoch 151/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9817 - loss: 0.0701 - val_accuracy: 0.6300 - val_loss: 2.8880
Epoch 152/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9804 - loss: 0.0619 - val_accuracy: 0.6450 - val_loss: 2.9605
Epoch 153/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9932 - loss: 0.0343 - val_accuracy: 0.6550 - val_loss: 2.7758
Epoch 154/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.9638 - loss: 0.1059 - val_accuracy: 0.6250 - val_loss: 3.0840
Epoch 155/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9836 - loss: 0.0825 - val_accuracy: 0.5850 - val_loss: 3.1181
Epoch 156/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9827 - loss: 0.1365 - val_accuracy: 0.6250 - val_loss: 2.3661
Epoch 157/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9843 - loss: 0.0540 - val_accuracy: 0.6200 - val_loss: 3.2481
Epoch 158/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 157ms/step - accuracy: 0.9880 - loss: 0.0368 - val_accuracy: 0.6100 - val_loss: 2.9156
Epoch 159/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9795 - loss: 0.0625 - val_accuracy: 0.6350 - val_loss: 2.9362
Epoch 160/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9871 - loss: 0.0501 - val_accuracy: 0.6100 - val_loss: 2.6436
Epoch 161/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9893 - loss: 0.0645 - val_accuracy: 0.6000 - val_loss: 3.0421
Epoch 162/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9654 - loss: 0.0766 - val_accuracy: 0.6750 - val_loss: 2.8712
Epoch 163/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 161ms/step - accuracy: 0.9946 - loss: 0.0277 - val_accuracy: 0.6450 - val_loss: 3.0525
Epoch 164/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9867 - loss: 0.0390 - val_accuracy: 0.6050 - val_loss: 3.7059
```

```
Epoch 165/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9782 - loss: 0.0850 - val_accuracy: 0.6400 - val_loss: 3.2004
Epoch 166/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 150ms/step - accuracy: 0.9659 - loss: 0.1274 - val_accuracy: 0.6400 - val_loss: 2.7282
Epoch 167/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9869 - loss: 0.0290 - val_accuracy: 0.6350 - val_loss: 2.9333
Epoch 168/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9765 - loss: 0.0736 - val_accuracy: 0.6200 - val_loss: 3.0869
Epoch 169/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9709 - loss: 0.0626 - val_accuracy: 0.6200 - val_loss: 3.1077
Epoch 170/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9798 - loss: 0.0398 - val_accuracy: 0.6150 - val_loss: 3.1138
Epoch 171/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9849 - loss: 0.0470 - val_accuracy: 0.6300 - val_loss: 3.1697
Epoch 172/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9905 - loss: 0.0274 - val_accuracy: 0.6200 - val_loss: 2.9240
Epoch 173/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9799 - loss: 0.0688 - val_accuracy: 0.6250 - val_loss: 3.0502
Epoch 174/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9785 - loss: 0.0580 - val_accuracy: 0.5950 - val_loss: 3.1900
Epoch 175/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9910 - loss: 0.0383 - val_accuracy: 0.6300 - val_loss: 3.1558
Epoch 176/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9687 - loss: 0.0718 - val_accuracy: 0.6350 - val_loss: 3.1910
Epoch 177/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9696 - loss: 0.0787 - val_accuracy: 0.6450 - val_loss: 2.9201
Epoch 178/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9831 - loss: 0.0540 - val_accuracy: 0.6400 - val_loss: 2.9415
Epoch 179/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9862 - loss: 0.0781 - val_accuracy: 0.6650 - val_loss: 3.0149
Epoch 180/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9834 - loss: 0.0509 - val_accuracy: 0.6450 - val_loss: 2.8482
Epoch 181/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9862 - loss: 0.0647 - val_accuracy: 0.6500 - val_loss: 3.4474
Epoch 182/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9794 - loss: 0.1101 - val_accuracy: 0.6500 - val_loss: 2.7705
Epoch 183/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9850 - loss: 0.0286 - val_accuracy: 0.6600 - val_loss: 3.5766
Epoch 184/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9922 - loss: 0.0405 - val_accuracy: 0.6650 - val_loss: 2.8988
Epoch 185/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9823 - loss: 0.0602 - val_accuracy: 0.6500 - val_loss: 2.8050
Epoch 186/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9909 - loss: 0.0293 - val_accuracy: 0.6450 - val_loss: 2.7298
Epoch 187/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9860 - loss: 0.0411 - val_accuracy: 0.6450 - val_loss: 3.0310
Epoch 188/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9940 - loss: 0.0242 - val_accuracy: 0.6600 - val_loss: 3.1644
Epoch 189/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9837 - loss: 0.0473 - val_accuracy: 0.6550 - val_loss: 2.8926
Epoch 190/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 156ms/step - accuracy: 0.9790 - loss: 0.1066 - val_accuracy: 0.6300 - val_loss: 2.9581
Epoch 191/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9876 - loss: 0.0275 - val_accuracy: 0.6350 - val_loss: 3.0488
Epoch 192/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9851 - loss: 0.0696 - val_accuracy: 0.6200 - val_loss: 3.1620
Epoch 193/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 154ms/step - accuracy: 0.9921 - loss: 0.0154 - val_accuracy: 0.6500 - val_loss: 3.8708
Epoch 194/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 153ms/step - accuracy: 0.9761 - loss: 0.1093 - val_accuracy: 0.6750 - val_loss: 2.9113
Epoch 195/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 151ms/step - accuracy: 0.9773 - loss: 0.0973 - val_accuracy: 0.6550 - val_loss: 3.0089
Epoch 196/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 158ms/step - accuracy: 0.9829 - loss: 0.0561 - val_accuracy: 0.6550 - val_loss: 2.9303
Epoch 197/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 152ms/step - accuracy: 0.9827 - loss: 0.0508 - val_accuracy: 0.6300 - val_loss: 3.1155
Epoch 198/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9807 - loss: 0.0829 - val_accuracy: 0.6350 - val_loss: 2.8135
Epoch 199/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 155ms/step - accuracy: 0.9841 - loss: 0.0454 - val_accuracy: 0.6450 - val_loss: 3.5844
Epoch 200/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 148ms/step - accuracy: 0.9732 - loss: 0.0892 - val_accuracy: 0.6450 - val_loss: 2.8868
```

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

```
Execution time: 770.62 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)
```