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 8 worker

In [2]:
```
# Set the number of threads
number_of_worker = 8
os.environ['OMP_NUM_THREADS'] = '8'  # OpenMP threads for parallelism
os.environ['TF_NUM_INTEROP_THREADS'] = '8'  # Threads for inter-operation parallelism
os.environ['TF_NUM_INTRAOP_THREADS'] = '8'  # 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 126ms/step - accuracy: 0.0908 - loss: 2.3262 - val_accuracy: 0.1000 - val_loss: 2.3015
Epoch 2/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 135ms/step - accuracy: 0.1362 - loss: 2.2996 - val_accuracy: 0.1950 - val_loss: 2.2538
Epoch 3/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 136ms/step - accuracy: 0.1901 - loss: 2.2619 - val_accuracy: 0.1550 - val_loss: 2.1788
Epoch 4/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 136ms/step - accuracy: 0.2647 - loss: 2.0942 - val_accuracy: 0.1650 - val_loss: 2.1146
Epoch 5/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 135ms/step - accuracy: 0.2221 - loss: 2.0821 - val_accuracy: 0.2550 - val_loss: 2.0688
Epoch 6/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.2638 - loss: 1.9729 - val_accuracy: 0.1850 - val_loss: 2.1266
Epoch 7/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.2670 - loss: 2.0135 - val_accuracy: 0.3050 - val_loss: 1.9701
Epoch 8/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.3221 - loss: 1.8983 - val_accuracy: 0.3200 - val_loss: 1.9209
Epoch 9/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.3465 - loss: 1.8112 - val_accuracy: 0.3850 - val_loss: 1.7874
Epoch 10/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 134ms/step - accuracy: 0.3333 - loss: 1.7993 - val_accuracy: 0.3800 - val_loss: 1.7778
Epoch 11/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.3197 - loss: 1.7694 - val_accuracy: 0.3800 - val_loss: 1.8889
Epoch 12/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.4073 - loss: 1.7037 - val_accuracy: 0.3950 - val_loss: 1.7497
Epoch 13/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.4157 - loss: 1.6517 - val_accuracy: 0.4050 - val_loss: 1.6553
Epoch 14/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.4353 - loss: 1.5729 - val_accuracy: 0.4100 - val_loss: 1.5987
Epoch 15/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.4026 - loss: 1.6035 - val_accuracy: 0.4650 - val_loss: 1.5758
Epoch 16/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.4446 - loss: 1.5463 - val_accuracy: 0.4250 - val_loss: 1.5314
Epoch 17/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.4380 - loss: 1.5158 - val_accuracy: 0.3150 - val_loss: 1.9509
Epoch 18/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.4421 - loss: 1.5638 - val_accuracy: 0.5050 - val_loss: 1.5129
Epoch 19/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.4866 - loss: 1.4481 - val_accuracy: 0.4600 - val_loss: 1.4984
Epoch 20/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.5063 - loss: 1.3240 - val_accuracy: 0.5150 - val_loss: 1.4143
Epoch 21/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.4910 - loss: 1.3623 - val_accuracy: 0.5200 - val_loss: 1.4117
Epoch 22/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 144ms/step - accuracy: 0.5047 - loss: 1.3392 - val_accuracy: 0.4800 - val_loss: 1.5531
Epoch 23/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.5424 - loss: 1.2446 - val_accuracy: 0.5400 - val_loss: 1.3136
Epoch 24/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.5654 - loss: 1.2154 - val_accuracy: 0.4050 - val_loss: 1.7990
Epoch 25/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.5409 - loss: 1.2645 - val_accuracy: 0.5100 - val_loss: 1.5017
Epoch 26/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.6027 - loss: 1.1757 - val_accuracy: 0.5600 - val_loss: 1.3537
Epoch 27/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.5567 - loss: 1.1760 - val_accuracy: 0.5450 - val_loss: 1.3311
Epoch 28/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.6314 - loss: 1.0326 - val_accuracy: 0.4900 - val_loss: 1.6634
Epoch 29/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.5486 - loss: 1.2524 - val_accuracy: 0.5700 - val_loss: 1.2929
Epoch 30/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.6574 - loss: 1.0052 - val_accuracy: 0.5450 - val_loss: 1.3549
Epoch 31/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.6195 - loss: 1.0483 - val_accuracy: 0.5400 - val_loss: 1.2411
Epoch 32/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.6598 - loss: 0.9453 - val_accuracy: 0.5750 - val_loss: 1.2204
Epoch 33/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.6279 - loss: 1.0213 - val_accuracy: 0.6000 - val_loss: 1.2158
Epoch 34/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 138ms/step - accuracy: 0.6599 - loss: 0.9575 - val_accuracy: 0.6000 - val_loss: 1.2094
Epoch 35/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.6886 - loss: 0.8812 - val_accuracy: 0.5850 - val_loss: 1.1563
Epoch 36/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.6898 - loss: 0.8930 - val_accuracy: 0.5550 - val_loss: 1.2696
Epoch 37/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.6576 - loss: 0.8930 - val_accuracy: 0.5500 - val_loss: 1.2612
Epoch 38/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.6722 - loss: 0.9251 - val_accuracy: 0.5900 - val_loss: 1.2177
Epoch 39/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.6881 - loss: 0.8160 - val_accuracy: 0.6000 - val_loss: 1.1883
Epoch 40/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.7392 - loss: 0.7417 - val_accuracy: 0.5600 - val_loss: 1.1867
Epoch 41/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.6914 - loss: 0.8287 - val_accuracy: 0.5600 - val_loss: 1.3192
```

```
Epoch 42/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.7687 - loss: 0.6921 - val_accuracy: 0.5750 - val_loss: 1.3051
Epoch 43/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.7411 - loss: 0.7306 - val_accuracy: 0.5950 - val_loss: 1.1532
Epoch 44/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.7491 - loss: 0.6981 - val_accuracy: 0.5800 - val_loss: 1.4488
Epoch 45/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.7577 - loss: 0.6257 - val_accuracy: 0.5050 - val_loss: 1.5136
Epoch 46/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.7645 - loss: 0.6471 - val_accuracy: 0.5900 - val_loss: 1.2315
Epoch 47/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.7872 - loss: 0.5983 - val_accuracy: 0.6150 - val_loss: 1.2239
Epoch 48/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.7524 - loss: 0.7278 - val_accuracy: 0.6250 - val_loss: 1.1958
Epoch 49/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.7851 - loss: 0.6215 - val_accuracy: 0.6300 - val_loss: 1.2098
Epoch 50/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.8122 - loss: 0.5169 - val_accuracy: 0.6100 - val_loss: 1.3742
Epoch 51/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.8208 - loss: 0.4952 - val_accuracy: 0.6300 - val_loss: 1.3932
Epoch 52/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.8011 - loss: 0.5288 - val_accuracy: 0.5900 - val_loss: 1.3119
Epoch 53/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.8117 - loss: 0.4985 - val_accuracy: 0.6000 - val_loss: 1.4959
Epoch 54/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.8080 - loss: 0.4890 - val_accuracy: 0.6050 - val_loss: 1.4572
Epoch 55/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.8633 - loss: 0.3825 - val_accuracy: 0.6000 - val_loss: 1.4302
Epoch 56/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.8359 - loss: 0.4682 - val_accuracy: 0.5900 - val_loss: 1.4071
Epoch 57/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.8595 - loss: 0.4078 - val_accuracy: 0.5450 - val_loss: 1.7839
Epoch 58/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.8430 - loss: 0.3921 - val_accuracy: 0.5900 - val_loss: 1.5943
Epoch 59/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.8386 - loss: 0.4653 - val_accuracy: 0.6600 - val_loss: 1.2770
Epoch 60/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.8801 - loss: 0.3475 - val_accuracy: 0.5850 - val_loss: 1.5956
Epoch 61/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.8934 - loss: 0.3076 - val_accuracy: 0.6100 - val_loss: 1.5560
Epoch 62/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.8729 - loss: 0.2943 - val_accuracy: 0.5500 - val_loss: 2.1760
Epoch 63/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 133ms/step - accuracy: 0.8659 - loss: 0.3484 - val_accuracy: 0.6050 - val_loss: 1.3787
Epoch 64/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.8979 - loss: 0.3263 - val_accuracy: 0.6150 - val_loss: 1.2772
Epoch 65/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9240 - loss: 0.2520 - val_accuracy: 0.6150 - val_loss: 1.5082
Epoch 66/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9125 - loss: 0.2537 - val_accuracy: 0.6450 - val_loss: 1.6815
Epoch 67/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9034 - loss: 0.3102 - val_accuracy: 0.6150 - val_loss: 1.6648
Epoch 68/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.8977 - loss: 0.2868 - val_accuracy: 0.6150 - val_loss: 1.5470
Epoch 69/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 144ms/step - accuracy: 0.9120 - loss: 0.2509 - val_accuracy: 0.6050 - val_loss: 1.7457
Epoch 70/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.8936 - loss: 0.2831 - val_accuracy: 0.5750 - val_loss: 2.1970
Epoch 71/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.8854 - loss: 0.3099 - val_accuracy: 0.6350 - val_loss: 1.7903
Epoch 72/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9302 - loss: 0.2035 - val_accuracy: 0.6450 - val_loss: 1.5624
Epoch 73/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.9376 - loss: 0.1596 - val_accuracy: 0.6400 - val_loss: 1.7233
Epoch 74/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9390 - loss: 0.1822 - val_accuracy: 0.6400 - val_loss: 1.7426
Epoch 75/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9161 - loss: 0.2388 - val_accuracy: 0.6150 - val_loss: 1.8065
Epoch 76/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9217 - loss: 0.2217 - val_accuracy: 0.5850 - val_loss: 2.0870
Epoch 77/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9474 - loss: 0.1783 - val_accuracy: 0.5600 - val_loss: 1.9619
Epoch 78/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9547 - loss: 0.1430 - val_accuracy: 0.6400 - val_loss: 1.7234
Epoch 79/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9492 - loss: 0.1798 - val_accuracy: 0.6450 - val_loss: 1.7415
Epoch 80/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.9351 - loss: 0.1791 - val_accuracy: 0.6250 - val_loss: 2.0626
Epoch 81/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9433 - loss: 0.1763 - val_accuracy: 0.6250 - val_loss: 1.8965
Epoch 82/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9281 - loss: 0.2056 - val_accuracy: 0.6350 - val_loss: 1.8599
```

```
Epoch 83/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9496 - loss: 0.1491 - val_accuracy: 0.5900 - val_loss: 2.2457
Epoch 84/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9162 - loss: 0.2485 - val_accuracy: 0.5650 - val_loss: 2.3358
Epoch 85/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 144ms/step - accuracy: 0.9598 - loss: 0.1338 - val_accuracy: 0.6200 - val_loss: 2.0118
Epoch 86/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9405 - loss: 0.1782 - val_accuracy: 0.6250 - val_loss: 1.8883
Epoch 87/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9804 - loss: 0.0627 - val_accuracy: 0.6350 - val_loss: 2.0973
Epoch 88/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9515 - loss: 0.1667 - val_accuracy: 0.6250 - val_loss: 2.0799
Epoch 89/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9499 - loss: 0.1645 - val_accuracy: 0.6100 - val_loss: 2.1547
Epoch 90/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9439 - loss: 0.1523 - val_accuracy: 0.6250 - val_loss: 2.1661
Epoch 91/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9245 - loss: 0.1971 - val_accuracy: 0.5800 - val_loss: 2.1904
Epoch 92/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9467 - loss: 0.1535 - val_accuracy: 0.6200 - val_loss: 2.0070
Epoch 93/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9467 - loss: 0.2095 - val_accuracy: 0.6500 - val_loss: 1.8759
Epoch 94/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 135ms/step - accuracy: 0.9725 - loss: 0.0735 - val_accuracy: 0.6050 - val_loss: 2.1894
Epoch 95/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.9641 - loss: 0.1138 - val_accuracy: 0.6100 - val_loss: 2.1479
Epoch 96/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9625 - loss: 0.1299 - val_accuracy: 0.5650 - val_loss: 1.9899
Epoch 97/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9669 - loss: 0.1340 - val_accuracy: 0.6200 - val_loss: 2.0378
Epoch 98/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9557 - loss: 0.1493 - val_accuracy: 0.5800 - val_loss: 2.0380
Epoch 99/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.9268 - loss: 0.1841 - val_accuracy: 0.6200 - val_loss: 2.1253
Epoch 100/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9663 - loss: 0.1215 - val_accuracy: 0.6300 - val_loss: 2.0655
Epoch 101/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9568 - loss: 0.1693 - val_accuracy: 0.6250 - val_loss: 1.7176
Epoch 102/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9743 - loss: 0.0622 - val_accuracy: 0.5800 - val_loss: 2.7079
Epoch 103/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9500 - loss: 0.1408 - val_accuracy: 0.6400 - val_loss: 2.3660
Epoch 104/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9346 - loss: 0.1862 - val_accuracy: 0.6000 - val_loss: 2.5210
Epoch 105/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9648 - loss: 0.0955 - val_accuracy: 0.5850 - val_loss: 2.5043
Epoch 106/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9383 - loss: 0.2160 - val_accuracy: 0.6200 - val_loss: 2.3773
Epoch 107/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.9772 - loss: 0.0925 - val_accuracy: 0.6300 - val_loss: 2.3062
Epoch 108/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9854 - loss: 0.0618 - val_accuracy: 0.5800 - val_loss: 2.2823
Epoch 109/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.9648 - loss: 0.1287 - val_accuracy: 0.6500 - val_loss: 2.2313
Epoch 110/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9834 - loss: 0.0841 - val_accuracy: 0.6150 - val_loss: 2.8426
Epoch 111/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9567 - loss: 0.1696 - val_accuracy: 0.6000 - val_loss: 2.2645
Epoch 112/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9763 - loss: 0.0623 - val_accuracy: 0.5850 - val_loss: 3.1558
Epoch 113/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9779 - loss: 0.0871 - val_accuracy: 0.6000 - val_loss: 2.3316
Epoch 114/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.9771 - loss: 0.0701 - val_accuracy: 0.6100 - val_loss: 2.5346
Epoch 115/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9910 - loss: 0.0450 - val_accuracy: 0.6200 - val_loss: 2.6165
Epoch 116/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 133ms/step - accuracy: 0.9698 - loss: 0.0784 - val_accuracy: 0.6100 - val_loss: 2.5954
Epoch 117/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9836 - loss: 0.0577 - val_accuracy: 0.5850 - val_loss: 2.7277
Epoch 118/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9925 - loss: 0.0346 - val_accuracy: 0.6300 - val_loss: 2.8001
Epoch 119/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.9619 - loss: 0.1764 - val_accuracy: 0.6100 - val_loss: 2.6465
Epoch 120/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9778 - loss: 0.0528 - val_accuracy: 0.5600 - val_loss: 3.3701
Epoch 121/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9516 - loss: 0.1404 - val_accuracy: 0.6050 - val_loss: 2.2734
Epoch 122/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 144ms/step - accuracy: 0.9612 - loss: 0.1199 - val_accuracy: 0.5950 - val_loss: 3.1950
Epoch 123/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9744 - loss: 0.1086 - val_accuracy: 0.6000 - val_loss: 2.6465
```

```
Epoch 124/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9637 - loss: 0.0909 - val_accuracy: 0.6150 - val_loss: 2.5651
Epoch 125/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9712 - loss: 0.0857 - val_accuracy: 0.6150 - val_loss: 2.9187
Epoch 126/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9763 - loss: 0.0642 - val_accuracy: 0.6500 - val_loss: 2.5741
Epoch 127/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9774 - loss: 0.0712 - val_accuracy: 0.6100 - val_loss: 2.8007
Epoch 128/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9647 - loss: 0.1249 - val_accuracy: 0.6250 - val_loss: 2.3439
Epoch 129/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9923 - loss: 0.0272 - val_accuracy: 0.6050 - val_loss: 2.8320
Epoch 130/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9654 - loss: 0.0755 - val_accuracy: 0.6050 - val_loss: 2.5047
Epoch 131/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.9896 - loss: 0.0388 - val_accuracy: 0.6000 - val_loss: 2.9230
Epoch 132/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9675 - loss: 0.1126 - val_accuracy: 0.5800 - val_loss: 2.9205
Epoch 133/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.9697 - loss: 0.1045 - val_accuracy: 0.6300 - val_loss: 2.8700
Epoch 134/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9584 - loss: 0.1553 - val_accuracy: 0.6100 - val_loss: 3.2127
Epoch 135/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9559 - loss: 0.1463 - val_accuracy: 0.6400 - val_loss: 2.4648
Epoch 136/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9850 - loss: 0.0413 - val_accuracy: 0.6400 - val_loss: 2.4898
Epoch 137/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9781 - loss: 0.0685 - val_accuracy: 0.6100 - val_loss: 3.1533
Epoch 138/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9778 - loss: 0.0885 - val_accuracy: 0.6400 - val_loss: 2.9685
Epoch 139/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.9756 - loss: 0.0563 - val_accuracy: 0.5950 - val_loss: 3.0030
Epoch 140/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 140ms/step - accuracy: 0.9734 - loss: 0.1123 - val_accuracy: 0.6200 - val_loss: 2.6312
Epoch 141/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 144ms/step - accuracy: 0.9894 - loss: 0.0447 - val_accuracy: 0.6000 - val_loss: 2.7892
Epoch 142/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 136ms/step - accuracy: 0.9839 - loss: 0.0571 - val_accuracy: 0.6100 - val_loss: 3.5329
Epoch 143/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9584 - loss: 0.2007 - val_accuracy: 0.6000 - val_loss: 3.1208
Epoch 144/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9645 - loss: 0.0972 - val_accuracy: 0.6100 - val_loss: 3.0806
Epoch 145/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9682 - loss: 0.1397 - val_accuracy: 0.6200 - val_loss: 2.8241
Epoch 146/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9820 - loss: 0.0489 - val_accuracy: 0.6100 - val_loss: 2.8053
Epoch 147/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9878 - loss: 0.0593 - val_accuracy: 0.6150 - val_loss: 2.6795
Epoch 148/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 145ms/step - accuracy: 0.9806 - loss: 0.0387 - val_accuracy: 0.6450 - val_loss: 3.2969
Epoch 149/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9720 - loss: 0.0867 - val_accuracy: 0.5750 - val_loss: 3.8260
Epoch 150/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 144ms/step - accuracy: 0.9751 - loss: 0.0840 - val_accuracy: 0.6100 - val_loss: 3.1916
Epoch 151/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9577 - loss: 0.2086 - val_accuracy: 0.6250 - val_loss: 2.7172
Epoch 152/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9896 - loss: 0.0274 - val_accuracy: 0.6300 - val_loss: 2.7414
Epoch 153/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 144ms/step - accuracy: 0.9864 - loss: 0.0404 - val_accuracy: 0.6200 - val_loss: 2.8900
Epoch 154/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9674 - loss: 0.1702 - val_accuracy: 0.6150 - val_loss: 2.7966
Epoch 155/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9900 - loss: 0.0298 - val_accuracy: 0.6150 - val_loss: 3.0291
Epoch 156/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9665 - loss: 0.1125 - val_accuracy: 0.6450 - val_loss: 2.6484
Epoch 157/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 137ms/step - accuracy: 0.9833 - loss: 0.0632 - val_accuracy: 0.6200 - val_loss: 2.6650
Epoch 158/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9877 - loss: 0.0304 - val_accuracy: 0.5750 - val_loss: 3.2498
Epoch 159/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9653 - loss: 0.1523 - val_accuracy: 0.6350 - val_loss: 2.5158
Epoch 160/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9975 - loss: 0.0103 - val_accuracy: 0.6350 - val_loss: 2.9519
Epoch 161/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9862 - loss: 0.0531 - val_accuracy: 0.6300 - val_loss: 2.7204
Epoch 162/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9858 - loss: 0.0496 - val_accuracy: 0.6450 - val_loss: 3.1435
Epoch 163/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9840 - loss: 0.0454 - val_accuracy: 0.6350 - val_loss: 3.5224
Epoch 164/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9805 - loss: 0.0615 - val_accuracy: 0.6050 - val_loss: 3.2094
```

```
Epoch 165/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9871 - loss: 0.0618 - val_accuracy: 0.6350 - val_loss: 3.2888
Epoch 166/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9696 - loss: 0.1007 - val_accuracy: 0.6400 - val_loss: 2.6891
Epoch 167/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 145ms/step - accuracy: 0.9872 - loss: 0.0435 - val_accuracy: 0.5900 - val_loss: 3.3853
Epoch 168/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9914 - loss: 0.0244 - val_accuracy: 0.6200 - val_loss: 3.1923
Epoch 169/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9791 - loss: 0.0804 - val_accuracy: 0.6300 - val_loss: 2.9425
Epoch 170/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9858 - loss: 0.0313 - val_accuracy: 0.6400 - val_loss: 3.1421
Epoch 171/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9780 - loss: 0.1068 - val_accuracy: 0.6050 - val_loss: 2.7188
Epoch 172/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 145ms/step - accuracy: 0.9848 - loss: 0.0615 - val_accuracy: 0.6200 - val_loss: 2.4824
Epoch 173/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9863 - loss: 0.0414 - val_accuracy: 0.6400 - val_loss: 2.7057
Epoch 174/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9884 - loss: 0.0382 - val_accuracy: 0.6700 - val_loss: 2.7627
Epoch 175/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9889 - loss: 0.0458 - val_accuracy: 0.6450 - val_loss: 3.2791
Epoch 176/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9926 - loss: 0.0161 - val_accuracy: 0.6350 - val_loss: 3.3782
Epoch 177/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9737 - loss: 0.0997 - val_accuracy: 0.5950 - val_loss: 2.9673
Epoch 178/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9740 - loss: 0.0582 - val_accuracy: 0.6250 - val_loss: 2.7655
Epoch 179/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9769 - loss: 0.0929 - val_accuracy: 0.6100 - val_loss: 3.2484
Epoch 180/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9844 - loss: 0.0512 - val_accuracy: 0.6450 - val_loss: 3.1637
Epoch 181/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9837 - loss: 0.0645 - val_accuracy: 0.6100 - val_loss: 3.5123
Epoch 182/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 138ms/step - accuracy: 0.9856 - loss: 0.0575 - val_accuracy: 0.6300 - val_loss: 3.5459
Epoch 183/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9792 - loss: 0.0598 - val_accuracy: 0.6250 - val_loss: 3.2964
Epoch 184/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9887 - loss: 0.0531 - val_accuracy: 0.6150 - val_loss: 3.4861
Epoch 185/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9784 - loss: 0.0492 - val_accuracy: 0.6350 - val_loss: 3.3608
Epoch 186/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9810 - loss: 0.0573 - val_accuracy: 0.6200 - val_loss: 3.4178
Epoch 187/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 139ms/step - accuracy: 0.9845 - loss: 0.0397 - val_accuracy: 0.6050 - val_loss: 3.8684
Epoch 188/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9640 - loss: 0.1283 - val_accuracy: 0.6350 - val_loss: 3.2243
Epoch 189/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9854 - loss: 0.0466 - val_accuracy: 0.6200 - val_loss: 4.0521
Epoch 190/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 136ms/step - accuracy: 0.9810 - loss: 0.0668 - val_accuracy: 0.6050 - val_loss: 3.4468
Epoch 191/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 138ms/step - accuracy: 0.9773 - loss: 0.0749 - val_accuracy: 0.6200 - val_loss: 3.5472
Epoch 192/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 145ms/step - accuracy: 0.9875 - loss: 0.0332 - val_accuracy: 0.5800 - val_loss: 4.8018
Epoch 193/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9606 - loss: 0.1226 - val_accuracy: 0.6250 - val_loss: 3.5109
Epoch 194/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 139ms/step - accuracy: 0.9761 - loss: 0.0790 - val_accuracy: 0.6300 - val_loss: 4.0045
Epoch 195/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 140ms/step - accuracy: 0.9658 - loss: 0.1862 - val_accuracy: 0.6150 - val_loss: 3.4841
Epoch 196/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9821 - loss: 0.0515 - val_accuracy: 0.5900 - val_loss: 3.7746
Epoch 197/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 142ms/step - accuracy: 0.9773 - loss: 0.0986 - val_accuracy: 0.6100 - val_loss: 3.8439
Epoch 198/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9911 - loss: 0.0289 - val_accuracy: 0.6600 - val_loss: 3.3768
Epoch 199/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 141ms/step - accuracy: 0.9782 - loss: 0.0690 - val_accuracy: 0.6250 - val_loss: 3.4474
Epoch 200/200
25/25 ━━━━━━━━━━━━━━━━━━━━ 4s 143ms/step - accuracy: 0.9942 - loss: 0.0190 - val_accuracy: 0.6150 - val_loss: 4.5955
```

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

```
Execution time: 707.66 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)
```