

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 6 worker

```
In [2]: # Set the number of threads
number_of_worker = 6
os.environ['OMP_NUM_THREADS'] = '6' # OpenMP threads for parallelism
os.environ['TF_NUM_INTEROP_THREADS'] = '6' # Threads for inter-operation parallelism
os.environ['TF_NUM_INTRAOP_THREADS'] = '6' # 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(optimizer=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 ————— 5s 157ms/step - accuracy: 0.0890 - loss: 2.3322 - val_accuracy: 0.1000 - val_loss: 2.3011

Epoch 2/200
25/25 ————— 4s 154ms/step - accuracy: 0.1340 - loss: 2.3001 - val_accuracy: 0.1550 - val_loss: 2.2372

Epoch 3/200
25/25 ————— 4s 162ms/step - accuracy: 0.1880 - loss: 2.2027 - val_accuracy: 0.2400 - val_loss: 2.0769

Epoch 4/200
25/25 ————— 4s 160ms/step - accuracy: 0.2299 - loss: 2.1084 - val_accuracy: 0.2600 - val_loss: 2.0161

Epoch 5/200
25/25 ————— 4s 167ms/step - accuracy: 0.2161 - loss: 2.0444 - val_accuracy: 0.2450 - val_loss: 1.9046

Epoch 6/200
25/25 ————— 4s 160ms/step - accuracy: 0.2236 - loss: 2.0755 - val_accuracy: 0.2700 - val_loss: 1.9300

Epoch 7/200
25/25 ————— 4s 157ms/step - accuracy: 0.2998 - loss: 1.9565 - val_accuracy: 0.2900 - val_loss: 1.9200

Epoch 8/200
25/25 ————— 4s 163ms/step - accuracy: 0.2854 - loss: 1.9678 - val_accuracy: 0.2200 - val_loss: 2.0452

Epoch 9/200
25/25 ————— 4s 161ms/step - accuracy: 0.3138 - loss: 1.9735 - val_accuracy: 0.3500 - val_loss: 1.8268

Epoch 10/200
25/25 ————— 4s 166ms/step - accuracy: 0.3174 - loss: 1.8635 - val_accuracy: 0.3950 - val_loss: 1.7900

Epoch 11/200
25/25 ————— 4s 166ms/step - accuracy: 0.3454 - loss: 1.8097 - val_accuracy: 0.4200 - val_loss: 1.6308

Epoch 12/200
25/25 ————— 4s 167ms/step - accuracy: 0.3527 - loss: 1.7176 - val_accuracy: 0.4450 - val_loss: 1.5843

Epoch 13/200
25/25 ————— 4s 164ms/step - accuracy: 0.3828 - loss: 1.6564 - val_accuracy: 0.4500 - val_loss: 1.5476

Epoch 14/200
25/25 ————— 4s 167ms/step - accuracy: 0.3882 - loss: 1.6838 - val_accuracy: 0.4000 - val_loss: 1.6897

Epoch 15/200
25/25 ————— 4s 168ms/step - accuracy: 0.3887 - loss: 1.6812 - val_accuracy: 0.4350 - val_loss: 1.5513

Epoch 16/200
25/25 ————— 4s 169ms/step - accuracy: 0.4168 - loss: 1.6188 - val_accuracy: 0.3950 - val_loss: 1.5964

Epoch 17/200
25/25 ————— 4s 164ms/step - accuracy: 0.3895 - loss: 1.6571 - val_accuracy: 0.4650 - val_loss: 1.5162

Epoch 18/200
25/25 ————— 4s 170ms/step - accuracy: 0.4429 - loss: 1.5384 - val_accuracy: 0.5150 - val_loss: 1.4634

Epoch 19/200
25/25 ————— 4s 165ms/step - accuracy: 0.4582 - loss: 1.4640 - val_accuracy: 0.4400 - val_loss: 1.4517

Epoch 20/200
25/25 ————— 4s 168ms/step - accuracy: 0.4804 - loss: 1.4429 - val_accuracy: 0.4650 - val_loss: 1.4854

Epoch 21/200
25/25 ————— 4s 170ms/step - accuracy: 0.4517 - loss: 1.4612 - val_accuracy: 0.5150 - val_loss: 1.4165

Epoch 22/200
25/25 ————— 4s 170ms/step - accuracy: 0.4987 - loss: 1.3627 - val_accuracy: 0.4500 - val_loss: 1.4091

Epoch 23/200
25/25 ————— 4s 165ms/step - accuracy: 0.4607 - loss: 1.4274 - val_accuracy: 0.5600 - val_loss: 1.2584

Epoch 24/200
25/25 ————— 4s 163ms/step - accuracy: 0.5193 - loss: 1.3213 - val_accuracy: 0.5650 - val_loss: 1.2483

Epoch 25/200
25/25 ————— 4s 163ms/step - accuracy: 0.4835 - loss: 1.3475 - val_accuracy: 0.5500 - val_loss: 1.2676

Epoch 26/200
25/25 ————— 4s 165ms/step - accuracy: 0.5028 - loss: 1.3596 - val_accuracy: 0.5300 - val_loss: 1.2933

Epoch 27/200
25/25 ————— 4s 168ms/step - accuracy: 0.5224 - loss: 1.3174 - val_accuracy: 0.5300 - val_loss: 1.2774

Epoch 28/200
25/25 ————— 4s 168ms/step - accuracy: 0.5612 - loss: 1.2203 - val_accuracy: 0.5550 - val_loss: 1.2307

Epoch 29/200
25/25 ————— 4s 166ms/step - accuracy: 0.5548 - loss: 1.1674 - val_accuracy: 0.5950 - val_loss: 1.2334

Epoch 30/200
25/25 ————— 4s 167ms/step - accuracy: 0.5841 - loss: 1.1709 - val_accuracy: 0.5500 - val_loss: 1.1717

Epoch 31/200
25/25 ————— 4s 165ms/step - accuracy: 0.6043 - loss: 1.0935 - val_accuracy: 0.5400 - val_loss: 1.2270

Epoch 32/200
25/25 ————— 4s 169ms/step - accuracy: 0.5900 - loss: 1.1565 - val_accuracy: 0.6050 - val_loss: 1.1707

Epoch 33/200
25/25 ————— 4s 166ms/step - accuracy: 0.6057 - loss: 1.0924 - val_accuracy: 0.5700 - val_loss: 1.1845

Epoch 34/200
25/25 ————— 4s 166ms/step - accuracy: 0.6117 - loss: 1.0508 - val_accuracy: 0.5650 - val_loss: 1.2611

Epoch 35/200
25/25 ————— 4s 162ms/step - accuracy: 0.6288 - loss: 1.0134 - val_accuracy: 0.6050 - val_loss: 1.1702

Epoch 36/200
25/25 ————— 4s 167ms/step - accuracy: 0.6574 - loss: 1.0158 - val_accuracy: 0.5050 - val_loss: 1.3790

Epoch 37/200
25/25 ————— 4s 173ms/step - accuracy: 0.6466 - loss: 0.9846 - val_accuracy: 0.4950 - val_loss: 1.3185

Epoch 38/200
25/25 ————— 4s 168ms/step - accuracy: 0.6530 - loss: 0.9448 - val_accuracy: 0.6300 - val_loss: 1.1059

Epoch 39/200
25/25 ————— 4s 169ms/step - accuracy: 0.6636 - loss: 0.9406 - val_accuracy: 0.5450 - val_loss: 1.3051

Epoch 40/200
25/25 ————— 4s 171ms/step - accuracy: 0.7135 - loss: 0.7819 - val_accuracy: 0.5100 - val_loss: 1.4909

Epoch 41/200
25/25 ————— 4s 171ms/step - accuracy: 0.6594 - loss: 0.9155 - val_accuracy: 0.6400 - val_loss: 1.0680

Epoch 42/200
25/25 ————— 4s 165ms/step - accuracy: 0.7064 - loss: 0.8548 - val_accuracy: 0.6250 - val_loss: 1.1288

Epoch 43/200
25/25 ————— 4s 166ms/step - accuracy: 0.7434 - loss: 0.7080 - val_accuracy: 0.6400 - val_loss: 1.0650

Epoch 44/200
25/25 ————— 4s 168ms/step - accuracy: 0.7365 - loss: 0.7392 - val_accuracy: 0.6400 - val_loss: 1.2224

Epoch 45/200
25/25 ————— 4s 165ms/step - accuracy: 0.7206 - loss: 0.7465 - val_accuracy: 0.6050 - val_loss: 1.1413

Epoch 46/200
25/25 ————— 4s 167ms/step - accuracy: 0.7372 - loss: 0.7719 - val_accuracy: 0.6600 - val_loss: 1.0741

Epoch 47/200
25/25 ————— 4s 166ms/step - accuracy: 0.7668 - loss: 0.6776 - val_accuracy: 0.6050 - val_loss: 1.2668

Epoch 48/200
25/25 ————— 4s 172ms/step - accuracy: 0.7406 - loss: 0.6342 - val_accuracy: 0.6450 - val_loss: 1.0855

Epoch 49/200
25/25 ————— 4s 167ms/step - accuracy: 0.7495 - loss: 0.6687 - val_accuracy: 0.6200 - val_loss: 1.1426

Epoch 50/200
25/25 ————— 4s 174ms/step - accuracy: 0.7787 - loss: 0.5902 - val_accuracy: 0.5900 - val_loss: 1.3110

Epoch 51/200
25/25 ————— 4s 162ms/step - accuracy: 0.7930 - loss: 0.6049 - val_accuracy: 0.6400 - val_loss: 1.2216

Epoch 52/200
25/25 ————— 4s 167ms/step - accuracy: 0.8118 - loss: 0.5545 - val_accuracy: 0.6900 - val_loss: 1.1773

Epoch 53/200
25/25 ————— 4s 162ms/step - accuracy: 0.7822 - loss: 0.5856 - val_accuracy: 0.6450 - val_loss: 1.1829

Epoch 54/200
25/25 ————— 4s 169ms/step - accuracy: 0.7902 - loss: 0.5876 - val_accuracy: 0.6500 - val_loss: 1.1793

Epoch 55/200
25/25 ————— 4s 172ms/step - accuracy: 0.7871 - loss: 0.5644 - val_accuracy: 0.5750 - val_loss: 1.3634

Epoch 56/200
25/25 ————— 4s 174ms/step - accuracy: 0.8249 - loss: 0.4743 - val_accuracy: 0.6400 - val_loss: 1.3440

Epoch 57/200
25/25 ————— 4s 167ms/step - accuracy: 0.7946 - loss: 0.5888 - val_accuracy: 0.6550 - val_loss: 1.2579

Epoch 58/200
25/25 ————— 4s 169ms/step - accuracy: 0.8377 - loss: 0.4342 - val_accuracy: 0.6600 - val_loss: 1.2937

Epoch 59/200
25/25 ————— 4s 167ms/step - accuracy: 0.8479 - loss: 0.4293 - val_accuracy: 0.6600 - val_loss: 1.2260

Epoch 60/200
25/25 ————— 4s 169ms/step - accuracy: 0.8441 - loss: 0.4288 - val_accuracy: 0.6800 - val_loss: 1.1514

Epoch 61/200
25/25 ————— 4s 172ms/step - accuracy: 0.8759 - loss: 0.3626 - val_accuracy: 0.6300 - val_loss: 1.5372

Epoch 62/200
25/25 ————— 4s 166ms/step - accuracy: 0.8100 - loss: 0.5374 - val_accuracy: 0.6400 - val_loss: 1.2935

Epoch 63/200
25/25 ————— 4s 170ms/step - accuracy: 0.8421 - loss: 0.4204 - val_accuracy: 0.6200 - val_loss: 1.5336

Epoch 64/200
25/25 ————— 4s 170ms/step - accuracy: 0.8930 - loss: 0.2955 - val_accuracy: 0.6700 - val_loss: 1.3578

Epoch 65/200
25/25 ————— 4s 167ms/step - accuracy: 0.8889 - loss: 0.3113 - val_accuracy: 0.6150 - val_loss: 1.3991

Epoch 66/200
25/25 ————— 4s 172ms/step - accuracy: 0.8777 - loss: 0.3702 - val_accuracy: 0.6750 - val_loss: 1.3092

Epoch 67/200
25/25 ————— 4s 170ms/step - accuracy: 0.8840 - loss: 0.3203 - val_accuracy: 0.6650 - val_loss: 1.3457

Epoch 68/200
25/25 ————— 4s 171ms/step - accuracy: 0.8963 - loss: 0.2672 - val_accuracy: 0.6700 - val_loss: 1.3466

Epoch 69/200
25/25 ————— 4s 174ms/step - accuracy: 0.8528 - loss: 0.4241 - val_accuracy: 0.6050 - val_loss: 1.6286

Epoch 70/200
25/25 ————— 4s 170ms/step - accuracy: 0.9103 - loss: 0.2695 - val_accuracy: 0.6650 - val_loss: 1.4309

Epoch 71/200
25/25 ————— 4s 170ms/step - accuracy: 0.9254 - loss: 0.2268 - val_accuracy: 0.6350 - val_loss: 1.4934

Epoch 72/200
25/25 ————— 4s 169ms/step - accuracy: 0.9097 - loss: 0.2662 - val_accuracy: 0.6550 - val_loss: 1.5303

Epoch 73/200
25/25 ————— 4s 162ms/step - accuracy: 0.8869 - loss: 0.2817 - val_accuracy: 0.6650 - val_loss: 1.4049

Epoch 74/200
25/25 ————— 4s 172ms/step - accuracy: 0.9100 - loss: 0.2291 - val_accuracy: 0.6550 - val_loss: 1.3215

Epoch 75/200
25/25 ————— 4s 169ms/step - accuracy: 0.9192 - loss: 0.2422 - val_accuracy: 0.6750 - val_loss: 1.2551

Epoch 76/200
25/25 ————— 4s 162ms/step - accuracy: 0.9105 - loss: 0.2636 - val_accuracy: 0.6300 - val_loss: 1.5186

Epoch 77/200
25/25 ————— 4s 166ms/step - accuracy: 0.9350 - loss: 0.1721 - val_accuracy: 0.5850 - val_loss: 1.7975

Epoch 78/200
25/25 ————— 4s 170ms/step - accuracy: 0.9183 - loss: 0.2491 - val_accuracy: 0.6500 - val_loss: 1.4510

Epoch 79/200
25/25 ————— 4s 159ms/step - accuracy: 0.9141 - loss: 0.2119 - val_accuracy: 0.6500 - val_loss: 1.5305

Epoch 80/200
25/25 ————— 4s 168ms/step - accuracy: 0.9163 - loss: 0.2393 - val_accuracy: 0.6850 - val_loss: 1.3357

Epoch 81/200
25/25 ————— 4s 159ms/step - accuracy: 0.9439 - loss: 0.1721 - val_accuracy: 0.6800 - val_loss: 1.5198

Epoch 82/200
25/25 ————— 4s 166ms/step - accuracy: 0.9051 - loss: 0.2558 - val_accuracy: 0.6450 - val_loss: 1.4418

Epoch 83/200
25/25 ————— 4s 172ms/step - accuracy: 0.9331 - loss: 0.1749 - val_accuracy: 0.6750 - val_loss: 1.4238

Epoch 84/200
25/25 ————— 4s 170ms/step - accuracy: 0.9438 - loss: 0.1506 - val_accuracy: 0.6850 - val_loss: 1.3142

Epoch 85/200
25/25 ————— 4s 171ms/step - accuracy: 0.9479 - loss: 0.1356 - val_accuracy: 0.6800 - val_loss: 1.5173

Epoch 86/200
25/25 ————— 4s 171ms/step - accuracy: 0.9248 - loss: 0.2362 - val_accuracy: 0.6600 - val_loss: 1.7096

Epoch 87/200
25/25 ————— 4s 172ms/step - accuracy: 0.9312 - loss: 0.1948 - val_accuracy: 0.6550 - val_loss: 1.7363

Epoch 88/200
25/25 ————— 4s 169ms/step - accuracy: 0.9397 - loss: 0.1930 - val_accuracy: 0.6350 - val_loss: 1.7357

Epoch 89/200
25/25 ————— 4s 171ms/step - accuracy: 0.9558 - loss: 0.1363 - val_accuracy: 0.6750 - val_loss: 1.4225

Epoch 90/200
25/25 ————— 4s 169ms/step - accuracy: 0.9226 - loss: 0.2255 - val_accuracy: 0.6750 - val_loss: 1.5259

Epoch 91/200
25/25 ————— 4s 173ms/step - accuracy: 0.9393 - loss: 0.1695 - val_accuracy: 0.6550 - val_loss: 1.8268

Epoch 92/200
25/25 ————— 4s 161ms/step - accuracy: 0.9457 - loss: 0.1703 - val_accuracy: 0.6550 - val_loss: 1.9608

Epoch 93/200
25/25 ————— 4s 167ms/step - accuracy: 0.9414 - loss: 0.1481 - val_accuracy: 0.6550 - val_loss: 1.5784

Epoch 94/200
25/25 ————— 4s 170ms/step - accuracy: 0.9607 - loss: 0.1032 - val_accuracy: 0.6550 - val_loss: 1.7475

Epoch 95/200
25/25 ————— 4s 165ms/step - accuracy: 0.9778 - loss: 0.1090 - val_accuracy: 0.6700 - val_loss: 1.4802

Epoch 96/200
25/25 ————— 4s 172ms/step - accuracy: 0.9754 - loss: 0.1131 - val_accuracy: 0.6850 - val_loss: 1.4780

Epoch 97/200
25/25 ————— 4s 171ms/step - accuracy: 0.9543 - loss: 0.1551 - val_accuracy: 0.4850 - val_loss: 3.1356

Epoch 98/200
25/25 ————— 4s 163ms/step - accuracy: 0.9005 - loss: 0.3302 - val_accuracy: 0.6600 - val_loss: 1.5196

Epoch 99/200
25/25 ————— 4s 164ms/step - accuracy: 0.9756 - loss: 0.0736 - val_accuracy: 0.6750 - val_loss: 1.6067

Epoch 100/200
25/25 ————— 4s 170ms/step - accuracy: 0.9593 - loss: 0.1224 - val_accuracy: 0.6150 - val_loss: 2.1983

Epoch 101/200
25/25 ————— 4s 173ms/step - accuracy: 0.9357 - loss: 0.2089 - val_accuracy: 0.6800 - val_loss: 1.7204

Epoch 102/200
25/25 ————— 4s 173ms/step - accuracy: 0.9600 - loss: 0.1106 - val_accuracy: 0.6600 - val_loss: 2.1862

Epoch 103/200
25/25 ————— 4s 173ms/step - accuracy: 0.9742 - loss: 0.1175 - val_accuracy: 0.6700 - val_loss: 1.7248

Epoch 104/200
25/25 ————— 4s 171ms/step - accuracy: 0.9673 - loss: 0.1144 - val_accuracy: 0.6800 - val_loss: 1.7310

Epoch 105/200
25/25 ————— 4s 170ms/step - accuracy: 0.9703 - loss: 0.0887 - val_accuracy: 0.6900 - val_loss: 1.7657

Epoch 106/200
25/25 ————— 4s 170ms/step - accuracy: 0.9544 - loss: 0.1503 - val_accuracy: 0.6500 - val_loss: 1.8447

Epoch 107/200
25/25 ————— 4s 171ms/step - accuracy: 0.9348 - loss: 0.1630 - val_accuracy: 0.6800 - val_loss: 1.7305

Epoch 108/200
25/25 ————— 4s 178ms/step - accuracy: 0.9695 - loss: 0.1021 - val_accuracy: 0.6550 - val_loss: 2.1379

Epoch 109/200
25/25 ————— 4s 161ms/step - accuracy: 0.9433 - loss: 0.1890 - val_accuracy: 0.6500 - val_loss: 1.6394

Epoch 110/200
25/25 ————— 4s 166ms/step - accuracy: 0.9777 - loss: 0.1011 - val_accuracy: 0.6550 - val_loss: 1.5713

Epoch 111/200
25/25 ————— 4s 171ms/step - accuracy: 0.9575 - loss: 0.1313 - val_accuracy: 0.6700 - val_loss: 1.6176

Epoch 112/200
25/25 ————— 4s 169ms/step - accuracy: 0.9760 - loss: 0.0796 - val_accuracy: 0.6400 - val_loss: 2.2218

Epoch 113/200
25/25 ————— 4s 155ms/step - accuracy: 0.9694 - loss: 0.0870 - val_accuracy: 0.6450 - val_loss: 2.2486

Epoch 114/200
25/25 ————— 4s 160ms/step - accuracy: 0.9743 - loss: 0.0628 - val_accuracy: 0.5700 - val_loss: 3.4683

Epoch 115/200
25/25 ————— 4s 169ms/step - accuracy: 0.9345 - loss: 0.2422 - val_accuracy: 0.6000 - val_loss: 2.5655

Epoch 116/200
25/25 ————— 4s 168ms/step - accuracy: 0.9648 - loss: 0.1994 - val_accuracy: 0.6300 - val_loss: 2.0663

Epoch 117/200
25/25 ————— 4s 174ms/step - accuracy: 0.9458 - loss: 0.1140 - val_accuracy: 0.6750 - val_loss: 1.9091

Epoch 118/200
25/25 ————— 4s 174ms/step - accuracy: 0.9762 - loss: 0.0969 - val_accuracy: 0.7050 - val_loss: 1.7098

Epoch 119/200
25/25 ————— 4s 169ms/step - accuracy: 0.9849 - loss: 0.0512 - val_accuracy: 0.6600 - val_loss: 2.0054

Epoch 120/200
25/25 ————— 4s 169ms/step - accuracy: 0.9551 - loss: 0.1360 - val_accuracy: 0.6650 - val_loss: 1.9079

Epoch 121/200
25/25 ————— 4s 170ms/step - accuracy: 0.9772 - loss: 0.0688 - val_accuracy: 0.6650 - val_loss: 1.8168

Epoch 122/200
25/25 ————— 4s 169ms/step - accuracy: 0.9620 - loss: 0.1115 - val_accuracy: 0.6750 - val_loss: 1.8947

Epoch 123/200
25/25 ————— 4s 179ms/step - accuracy: 0.9783 - loss: 0.0906 - val_accuracy: 0.6550 - val_loss: 1.8924

Epoch 124/200
25/25 ————— 4s 163ms/step - accuracy: 0.9669 - loss: 0.1133 - val_accuracy: 0.6750 - val_loss: 1.7190
Epoch 125/200
25/25 ————— 5s 180ms/step - accuracy: 0.9892 - loss: 0.0311 - val_accuracy: 0.6550 - val_loss: 2.1296
Epoch 126/200
25/25 ————— 4s 169ms/step - accuracy: 0.9659 - loss: 0.0754 - val_accuracy: 0.6700 - val_loss: 1.9735
Epoch 127/200
25/25 ————— 4s 177ms/step - accuracy: 0.9602 - loss: 0.1226 - val_accuracy: 0.6650 - val_loss: 2.0814
Epoch 128/200
25/25 ————— 4s 173ms/step - accuracy: 0.9858 - loss: 0.0539 - val_accuracy: 0.6800 - val_loss: 2.1406
Epoch 129/200
25/25 ————— 4s 169ms/step - accuracy: 0.9681 - loss: 0.1102 - val_accuracy: 0.6750 - val_loss: 1.9733
Epoch 130/200
25/25 ————— 4s 174ms/step - accuracy: 0.9774 - loss: 0.0680 - val_accuracy: 0.6700 - val_loss: 1.7611
Epoch 131/200
25/25 ————— 4s 174ms/step - accuracy: 0.9811 - loss: 0.0621 - val_accuracy: 0.6500 - val_loss: 1.8567
Epoch 132/200
25/25 ————— 4s 168ms/step - accuracy: 0.9886 - loss: 0.0347 - val_accuracy: 0.6050 - val_loss: 2.9189
Epoch 133/200
25/25 ————— 4s 171ms/step - accuracy: 0.9355 - loss: 0.3112 - val_accuracy: 0.6600 - val_loss: 1.9797
Epoch 134/200
25/25 ————— 4s 172ms/step - accuracy: 0.9788 - loss: 0.0517 - val_accuracy: 0.6500 - val_loss: 2.3874
Epoch 135/200
25/25 ————— 4s 173ms/step - accuracy: 0.9656 - loss: 0.0846 - val_accuracy: 0.6650 - val_loss: 2.0185
Epoch 136/200
25/25 ————— 4s 171ms/step - accuracy: 0.9800 - loss: 0.0580 - val_accuracy: 0.6900 - val_loss: 1.9009
Epoch 137/200
25/25 ————— 4s 159ms/step - accuracy: 0.9758 - loss: 0.0743 - val_accuracy: 0.6350 - val_loss: 2.5351
Epoch 138/200
25/25 ————— 4s 161ms/step - accuracy: 0.9412 - loss: 0.1937 - val_accuracy: 0.6750 - val_loss: 2.3021
Epoch 139/200
25/25 ————— 4s 172ms/step - accuracy: 0.9578 - loss: 0.0986 - val_accuracy: 0.6600 - val_loss: 2.4281
Epoch 140/200
25/25 ————— 4s 171ms/step - accuracy: 0.9918 - loss: 0.0322 - val_accuracy: 0.6400 - val_loss: 2.8026
Epoch 141/200
25/25 ————— 4s 167ms/step - accuracy: 0.9650 - loss: 0.0813 - val_accuracy: 0.6900 - val_loss: 2.1600
Epoch 142/200
25/25 ————— 4s 173ms/step - accuracy: 0.9815 - loss: 0.0500 - val_accuracy: 0.6800 - val_loss: 2.1603
Epoch 143/200
25/25 ————— 4s 173ms/step - accuracy: 0.9883 - loss: 0.0346 - val_accuracy: 0.6850 - val_loss: 2.0507
Epoch 144/200
25/25 ————— 4s 169ms/step - accuracy: 0.9832 - loss: 0.0488 - val_accuracy: 0.6600 - val_loss: 2.2622
Epoch 145/200
25/25 ————— 4s 171ms/step - accuracy: 0.9690 - loss: 0.1287 - val_accuracy: 0.6600 - val_loss: 2.0851
Epoch 146/200
25/25 ————— 4s 172ms/step - accuracy: 0.9703 - loss: 0.0971 - val_accuracy: 0.6600 - val_loss: 2.2161
Epoch 147/200
25/25 ————— 4s 172ms/step - accuracy: 0.9647 - loss: 0.1197 - val_accuracy: 0.6900 - val_loss: 1.9852
Epoch 148/200
25/25 ————— 4s 170ms/step - accuracy: 0.9848 - loss: 0.0470 - val_accuracy: 0.6800 - val_loss: 2.1950
Epoch 149/200
25/25 ————— 4s 168ms/step - accuracy: 0.9646 - loss: 0.1005 - val_accuracy: 0.6750 - val_loss: 2.2800
Epoch 150/200
25/25 ————— 4s 171ms/step - accuracy: 0.9792 - loss: 0.0724 - val_accuracy: 0.6750 - val_loss: 2.0512
Epoch 151/200
25/25 ————— 4s 170ms/step - accuracy: 0.9825 - loss: 0.0804 - val_accuracy: 0.6900 - val_loss: 2.4662
Epoch 152/200
25/25 ————— 4s 171ms/step - accuracy: 0.9782 - loss: 0.0690 - val_accuracy: 0.6450 - val_loss: 2.5868
Epoch 153/200
25/25 ————— 4s 171ms/step - accuracy: 0.9741 - loss: 0.0586 - val_accuracy: 0.6700 - val_loss: 2.2021
Epoch 154/200
25/25 ————— 4s 173ms/step - accuracy: 0.9772 - loss: 0.0831 - val_accuracy: 0.6850 - val_loss: 2.1732
Epoch 155/200
25/25 ————— 4s 174ms/step - accuracy: 0.9813 - loss: 0.0443 - val_accuracy: 0.6250 - val_loss: 2.8519
Epoch 156/200
25/25 ————— 4s 168ms/step - accuracy: 0.9704 - loss: 0.0755 - val_accuracy: 0.6450 - val_loss: 2.3332
Epoch 157/200
25/25 ————— 4s 173ms/step - accuracy: 0.9592 - loss: 0.1600 - val_accuracy: 0.6850 - val_loss: 2.0250
Epoch 158/200
25/25 ————— 4s 172ms/step - accuracy: 0.9815 - loss: 0.0503 - val_accuracy: 0.6800 - val_loss: 2.1698
Epoch 159/200
25/25 ————— 4s 169ms/step - accuracy: 0.9784 - loss: 0.0438 - val_accuracy: 0.7050 - val_loss: 2.1671
Epoch 160/200
25/25 ————— 4s 169ms/step - accuracy: 0.9854 - loss: 0.0685 - val_accuracy: 0.6750 - val_loss: 2.2213
Epoch 161/200
25/25 ————— 4s 175ms/step - accuracy: 0.9817 - loss: 0.0532 - val_accuracy: 0.6800 - val_loss: 2.1180
Epoch 162/200
25/25 ————— 5s 179ms/step - accuracy: 0.9793 - loss: 0.0800 - val_accuracy: 0.6650 - val_loss: 2.0562
Epoch 163/200
25/25 ————— 4s 168ms/step - accuracy: 0.9934 - loss: 0.0224 - val_accuracy: 0.6800 - val_loss: 2.3128
Epoch 164/200
25/25 ————— 4s 172ms/step - accuracy: 0.9781 - loss: 0.0955 - val_accuracy: 0.6850 - val_loss: 2.1782


```

Epoch 165/200
25/25 ————— 4s 161ms/step - accuracy: 0.9667 - loss: 0.1743 - val_accuracy: 0.6650 - val_loss: 2.3627
Epoch 166/200
25/25 ————— 4s 171ms/step - accuracy: 0.9924 - loss: 0.0248 - val_accuracy: 0.6650 - val_loss: 2.4867
Epoch 167/200
25/25 ————— 4s 171ms/step - accuracy: 0.9799 - loss: 0.0824 - val_accuracy: 0.6800 - val_loss: 2.2316
Epoch 168/200
25/25 ————— 4s 169ms/step - accuracy: 0.9718 - loss: 0.0875 - val_accuracy: 0.6750 - val_loss: 2.1804
Epoch 169/200
25/25 ————— 4s 174ms/step - accuracy: 0.9887 - loss: 0.0427 - val_accuracy: 0.6750 - val_loss: 2.0417
Epoch 170/200
25/25 ————— 4s 167ms/step - accuracy: 0.9892 - loss: 0.0395 - val_accuracy: 0.6750 - val_loss: 2.4242
Epoch 171/200
25/25 ————— 4s 168ms/step - accuracy: 0.9685 - loss: 0.1369 - val_accuracy: 0.6550 - val_loss: 2.5412
Epoch 172/200
25/25 ————— 4s 175ms/step - accuracy: 0.9850 - loss: 0.0492 - val_accuracy: 0.7000 - val_loss: 1.9487
Epoch 173/200
25/25 ————— 4s 170ms/step - accuracy: 0.9841 - loss: 0.0448 - val_accuracy: 0.6200 - val_loss: 2.5740
Epoch 174/200
25/25 ————— 4s 177ms/step - accuracy: 0.9827 - loss: 0.0601 - val_accuracy: 0.6850 - val_loss: 1.9880
Epoch 175/200
25/25 ————— 4s 172ms/step - accuracy: 0.9967 - loss: 0.0140 - val_accuracy: 0.6900 - val_loss: 2.1357
Epoch 176/200
25/25 ————— 4s 168ms/step - accuracy: 0.9829 - loss: 0.0802 - val_accuracy: 0.6800 - val_loss: 2.0864
Epoch 177/200
25/25 ————— 4s 167ms/step - accuracy: 0.9862 - loss: 0.0423 - val_accuracy: 0.6900 - val_loss: 2.1834
Epoch 178/200
25/25 ————— 4s 171ms/step - accuracy: 0.9574 - loss: 0.1224 - val_accuracy: 0.6650 - val_loss: 2.3492
Epoch 179/200
25/25 ————— 4s 172ms/step - accuracy: 0.9945 - loss: 0.0241 - val_accuracy: 0.6250 - val_loss: 3.3972
Epoch 180/200
25/25 ————— 4s 171ms/step - accuracy: 0.9831 - loss: 0.0915 - val_accuracy: 0.6700 - val_loss: 3.0905
Epoch 181/200
25/25 ————— 4s 162ms/step - accuracy: 0.9797 - loss: 0.0585 - val_accuracy: 0.6150 - val_loss: 3.0855
Epoch 182/200
25/25 ————— 4s 166ms/step - accuracy: 0.9878 - loss: 0.0432 - val_accuracy: 0.6750 - val_loss: 2.1840
Epoch 183/200
25/25 ————— 4s 173ms/step - accuracy: 0.9706 - loss: 0.1064 - val_accuracy: 0.6550 - val_loss: 2.7121
Epoch 184/200
25/25 ————— 4s 168ms/step - accuracy: 0.9745 - loss: 0.0734 - val_accuracy: 0.6900 - val_loss: 2.2102
Epoch 185/200
25/25 ————— 4s 172ms/step - accuracy: 0.9871 - loss: 0.0432 - val_accuracy: 0.6950 - val_loss: 2.0016
Epoch 186/200
25/25 ————— 4s 165ms/step - accuracy: 0.9841 - loss: 0.0458 - val_accuracy: 0.6650 - val_loss: 2.3818
Epoch 187/200
25/25 ————— 4s 171ms/step - accuracy: 0.9763 - loss: 0.0931 - val_accuracy: 0.6950 - val_loss: 2.1277
Epoch 188/200
25/25 ————— 4s 168ms/step - accuracy: 0.9905 - loss: 0.0339 - val_accuracy: 0.6950 - val_loss: 2.1814
Epoch 189/200
25/25 ————— 4s 172ms/step - accuracy: 0.9924 - loss: 0.0474 - val_accuracy: 0.6600 - val_loss: 2.8345
Epoch 190/200
25/25 ————— 4s 169ms/step - accuracy: 0.9727 - loss: 0.0834 - val_accuracy: 0.6600 - val_loss: 2.1733
Epoch 191/200
25/25 ————— 4s 171ms/step - accuracy: 0.9894 - loss: 0.0371 - val_accuracy: 0.6700 - val_loss: 2.3174
Epoch 192/200
25/25 ————— 4s 170ms/step - accuracy: 0.9936 - loss: 0.0223 - val_accuracy: 0.6850 - val_loss: 2.5772
Epoch 193/200
25/25 ————— 4s 160ms/step - accuracy: 0.9917 - loss: 0.0226 - val_accuracy: 0.6650 - val_loss: 2.3985
Epoch 194/200
25/25 ————— 4s 167ms/step - accuracy: 0.9880 - loss: 0.0382 - val_accuracy: 0.6750 - val_loss: 2.5574
Epoch 195/200
25/25 ————— 4s 170ms/step - accuracy: 0.9776 - loss: 0.1042 - val_accuracy: 0.6600 - val_loss: 2.7016
Epoch 196/200
25/25 ————— 4s 175ms/step - accuracy: 0.9847 - loss: 0.0479 - val_accuracy: 0.6950 - val_loss: 2.6320
Epoch 197/200
25/25 ————— 4s 170ms/step - accuracy: 0.9786 - loss: 0.0357 - val_accuracy: 0.6550 - val_loss: 2.3906
Epoch 198/200
25/25 ————— 4s 174ms/step - accuracy: 0.9901 - loss: 0.0448 - val_accuracy: 0.6600 - val_loss: 2.7241
Epoch 199/200
25/25 ————— 4s 167ms/step - accuracy: 0.9870 - loss: 0.0350 - val_accuracy: 0.6250 - val_loss: 2.9793
Epoch 200/200
25/25 ————— 4s 171ms/step - accuracy: 0.9906 - loss: 0.0226 - val_accuracy: 0.6500 - val_loss: 2.4609

```

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

Execution time: 850.77 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)
```