Model Development Phase Template

Date	11 December 2024
Team ID	739876
Project Title	Alzheimer Disease Prediction
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):

```
[ ]: ###Train the model

[31]: history = custom_inception_model.fit(train_data, train_labels, validation_data=(val_data, val_labels), epochs=30)

Epoch 1/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 201s 1s/step - accuracy: 0.3037 - loss: 1.7156 - val_accuracy: 0.5682 - val_loss: 1.1073
Epoch 2/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 203s 2s/step - accuracy: 0.4725 - loss: 1.2089 - val_accuracy: 0.6435 - val_loss: 0.8306
Epoch 3/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 218s 2s/step - accuracy: 0.5581 - loss: 0.9786 - val_accuracy: 0.6520 - val_loss: 0.7255
Epoch 4/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 224s 2s/step - accuracy: 0.6274 - loss: 0.8366 - val_accuracy: 0.6787 - val_loss: 0.6818
Epoch 5/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 195s 1s/step - accuracy: 0.6401 - loss: 0.7899 - val_accuracy: 0.6835 - val_loss: 0.6505
Epoch 6/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 218s 2s/step - accuracy: 0.6715 - loss: 0.7307 - val_accuracy: 0.6921 - val_loss: 0.6275
Epoch 7/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 224s 2s/step - accuracy: 0.6784 - loss: 0.7119 - val_accuracy: 0.7083 - val_loss: 0.6237
Epoch 8/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 187s 1s/step - accuracy: 0.6856 - loss: 0.6975 - val_accuracy: 0.7150 - val_loss: 0.5954
Epoch 9/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 220s 2s/step - accuracy: 0.6981 - loss: 0.6644 - val_accuracy: 0.7245 - val_loss: 0.5875
Epoch 10/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 177s 1s/step - accuracy: 0.7145 - loss: 0.6496 - val_accuracy: 0.7264 - val_loss: 0.5770
Epoch 11/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 283s 2s/step - accuracy: 0.7255 - loss: 0.6235 - val_accuracy: 0.7512 - val_loss: 0.5515
Epoch 12/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 218s 2s/step - accuracy: 0.7395 - loss: 0.6131 - val_accuracy: 0.7388 - val_loss: 0.5598
Epoch 13/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 242s 2s/step - accuracy: 0.7480 - loss: 0.5948 - val_accuracy: 0.7579 - val_loss: 0.5637
Epoch 14/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 320s 2s/step - accuracy: 0.7571 - loss: 0.5644 - val_accuracy: 0.7474 - val_loss: 0.5377
Epoch 15/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 319s 2s/step - accuracy: 0.7847 - loss: 0.5487 - val_accuracy: 0.7588 - val_loss: 0.5322
Epoch 16/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 216s 2s/step - accuracy: 0.7637 - loss: 0.5567 - val_accuracy: 0.7722 - val_loss: 0.5249
```

```
Epoch 17/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 205s 2s/step - accuracy: 0.7970 - loss: 0.5059 - val_accuracy: 0.7760 - val_loss: 0.5181
Epoch 18/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 215s 2s/step - accuracy: 0.8021 - loss: 0.4882 - val_accuracy: 0.7817 - val_loss: 0.5271
Epoch 19/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 199s 2s/step - accuracy: 0.8069 - loss: 0.4862 - val_accuracy: 0.7874 - val_loss: 0.5103
Epoch 20/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 196s 1s/step - accuracy: 0.8205 - loss: 0.4619 - val_accuracy: 0.7941 - val_loss: 0.5028
Epoch 21/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 210s 2s/step - accuracy: 0.8034 - loss: 0.4848 - val_accuracy: 0.7960 - val_loss: 0.5196
Epoch 22/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 211s 2s/step - accuracy: 0.8226 - loss: 0.4550 - val_accuracy: 0.7969 - val_loss: 0.4727
Epoch 23/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 276s 2s/step - accuracy: 0.8372 - loss: 0.4069 - val_accuracy: 0.8084 - val_loss: 0.4595
Epoch 24/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 302s 2s/step - accuracy: 0.8449 - loss: 0.3985 - val_accuracy: 0.8093 - val_loss: 0.4812
Epoch 25/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 226s 2s/step - accuracy: 0.8501 - loss: 0.4005 - val_accuracy: 0.7941 - val_loss: 0.4838
Epoch 26/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 253s 2s/step - accuracy: 0.8621 - loss: 0.3795 - val_accuracy: 0.8112 - val_loss: 0.4992
Epoch 27/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 231s 2s/step - accuracy: 0.8557 - loss: 0.3902 - val_accuracy: 0.8074 - val_loss: 0.4792
Epoch 28/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 224s 2s/step - accuracy: 0.8739 - loss: 0.3416 - val_accuracy: 0.8141 - val_loss: 0.4822
Epoch 29/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 238s 2s/step - accuracy: 0.8709 - loss: 0.3506 - val_accuracy: 0.8132 - val_loss: 0.4807
Epoch 30/30
132/132 ━━━━━━━━━━━━━━━━━━━━ 227s 2s/step - accuracy: 0.8684 - loss: 0.3752 - val_accuracy: 0.8208 - val_loss: 0.4703
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics

Xception Model

```
[ ]: ###Model Building & Testing

[ ]: ###Pre-trained CNN model as a Feature Extractor

[19]: IMAGE_SIZE = [180, 180]
      xcep_model = Xception(input_shape=IMAGE_SIZE+[3], weights= 'imagenet',include_top=False)
      for layer in xcep_model.layers:
          layer.trainable=False

[ ]: ###Create Sequential Layers

[27]: from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import (
          SeparableConv2D, BatchNormalization, GlobalAveragePooling2D,
          Dropout, Flatten, Dense
      )
      custom_inception_model = Sequential([
          xcep_model,
          Dropout(0.5),
          GlobalAveragePooling2D(),
          Flatten(),
          BatchNormalization(),
          Dense(512, activation='relu'),
          BatchNormalization(),
          Dropout(0.5),
          Dense(256, activation='relu'),
          BatchNormalization(),
          Dropout(0.5),
          Dense(128, activation='relu'),
          BatchNormalization(),
          Dropout(0.5),
          Dense(64, activation='relu'),
          Dropout(0.5),
          BatchNormalization(),
          Dense(4, activation='softmax')
      ], name="inception_cnn_model")
```

```
[ ]: ###Configure the Learning Process

[28]: custom_inception_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

[ ]: ###Test The Model

[57]: from tensorflow.keras.models import load_model
      from tensorflow.keras.preprocessing import image
      import numpy as np
      model = load_model('adp.h5')
      img = image.load_img(r"extract_files/Alzheimer_s Dataset/test/ModerateDemented/32.jpg", target_size=(180, 180))
      x = image.img_to_array(img)
      x = x / 255.0
      x = np.expand_dims(x, axis=0)
      preds = model.predict(x)
      pred = np.argmax(preds, axis=1)
      index = ['MildDemented', 'ModerateDemented', 'NonDemented', 'VeryMildDemented']
      result = str(index[pred[0]])
      print(result)

      WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you
      the model.
      1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
      ModerateDemented
```