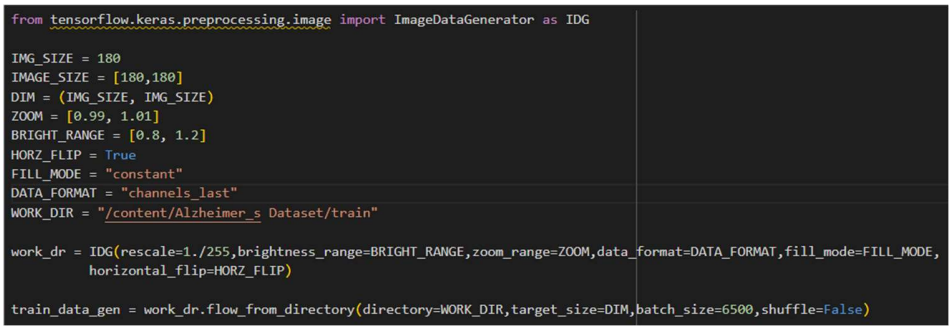
Model Optimization and Tuning Phase Template

Date	10 July 2024
Team ID	SWTID1720426301
Project Title	**Cognitive Care: Early Intervention For Alzheimer's Disease**
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation:

Model	Tuned Hyperparameters
Xception	• **Learning Rate**: Adam optimizer with a default learning rate of 0.001. • **Batch Size**: 6500 training samples per iteration. • **Epochs**: 30 complete passes through the training dataset. • **Dropout Rate**: 0.5 to prevent overfitting. • **Zoom Range**: Random zoom between 0.99 and 1.01. • **Brightness Range**: Random brightness adjustment between 0.8 and 1.2. • **Rescale**: Data normalized by scaling pixel values to 1./255. • **GlobalAveragePooling2D**: A pooling layer to reduce the spatial dimensions of the feature maps. 

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG

IMG_SIZE = 180
IMAGE_SIZE = [180,180]
DIM = (IMG_SIZE, IMG_SIZE)
ZOOM = [0.99, 1.01]
BRIGHT_RANGE = [0.8, 1.2]
HORZ_FLIP = True
FILL_MODE = "constant"
DATA_FORMAT = "channels_last"
WORK_DIR = "/content/Alzheimer_s Dataset/train"

work_dr = IDG(rescale=1./255,brightness_range=BRIGHT_RANGE,zoom_range=ZOOM,data_format=DATA_FORMAT,fill_mode=FILL_MODE,
          horizontal_flip=HORZ_FLIP)

train_data_gen = work_dr.flow_from_directory(directory=WORK_DIR,target_size=DIM,batch_size=6500,shuffle=False)
```

<table>
<tr>
<td></td>
<td>

```
[ ]  xcep_model = Xception(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

```
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")
```

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

```
[ ]  history = model.fit(train_data, train_labels, validation_data=(val_data, val_labels), epochs=30)
```

</td>
</tr>
<tr>
<td>VGG19</td>
<td>

- **Learning Rate**: Adam optimizer with a default learning rate of 0.001.
- **Batch Size**: 6500 training samples per iteration.
- **Epochs**: 30 complete passes through the training dataset.
- **Dropout Rate**: 0.5 to prevent overfitting.
- **Zoom Range**: Random zoom between 0.99 and 1.01.
- **Brightness Range**: Random brightness adjustment between 0.8 and 1.2.
- **Rescale**: Data normalized by scaling pixel values to 1./255.
- **Conv Block**: Multiple convolutional layers with small 3x3 filters.

</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG

IMG_SIZE = 180
IMAGE_SIZE = [180,180]
DIM = (IMG_SIZE, IMG_SIZE)
ZOOM = [0.99, 1.01]
BRIGHT_RANGE = [0.8, 1.2]
HORZ_FLIP = True
FILL_MODE = "constant"
DATA_FORMAT = "channels_last"
WORK_DIR = "/content/Alzheimer_s Dataset/train"

work_dr = IDG(rescale=1./255,brightness_range=BRIGHT_RANGE,zoom_range=ZOOM,data_format=DATA_FORMAT,fill_mode=FILL_MODE,
         horizontal_flip=HORZ_FLIP)

train_data_gen = work_dr.flow_from_directory(directory=WORK_DIR,target_size=DIM,batch_size=6500,shuffle=False)
```

```
from tensorflow.keras.applications import VGG19

vgg_model = VGG19(weights='imagenet', include_top=False, input_shape=(IMG_SIZE, IMG_SIZE, 3))
```

```
model = Sequential([
    vgg_model,
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dense(4, activation='softmax')
])
```

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

[ ] history = model.fit(train_data, train_labels, validation_data=(val_data, val_labels), epochs=30)
```

</td>
</tr>
<tr>
<td>Inception V3</td>
<td>

- **Learning Rate**: Adam optimizer with a default learning rate of 0.001.
- **Batch Size**: 6500 training samples per iteration.
- **Epochs**: 30 complete passes through the training dataset.
- **Dropout Rate**: 0.5 to prevent overfitting.
- **Zoom Range**: Random zoom between 0.99 and 1.01.
- **Brightness Range**: Random brightness adjustment between 0.8 and 1.2.
- **Rescale**: Data normalized by scaling pixel values to 1./255.
- **Factorized Convolutions**: Use of smaller convolutions like 1x7 and 7x1 to reduce computational

</td>
</tr>
</table>

cost.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG

IMG_SIZE = 180
IMAGE_SIZE = [180,180]
DIM = (IMG_SIZE, IMG_SIZE)
ZOOM = [0.99, 1.01]
BRIGHT_RANGE = [0.8, 1.2]
HORZ_FLIP = True
FILL_MODE = "constant"
DATA_FORMAT = "channels_last"
WORK_DIR = "/content/Alzheimer_s Dataset/train"

work_dr = IDG(rescale=1./255,brightness_range=BRIGHT_RANGE,zoom_range=ZOOM,data_format=DATA_FORMAT,fill_mode=FILL_MODE,
          horizontal_flip=HORZ_FLIP)

train_data_gen = work_dr.flow_from_directory(directory=WORK_DIR,target_size=DIM,batch_size=6500,shuffle=False)
```

```
from tensorflow.keras.applications import InceptionV3

inception_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(IMG_SIZE, IMG_SIZE, 3))
```

```
model = Sequential([
    inception_model,
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dense(4, activation='softmax')
])
```

```
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

```
[ ] history = model.fit(train_data, train_labels, validation_data=(val_data, val_labels), epochs=30)
```

Final Model Selection Justification:

Final Model	Reasoning
Xception	The Xception model was chosen as the final optimized model for its consistent improvement in accuracy and validation metrics over 30

	epochs, achieving a final validation accuracy of 85.36%. The model effectively learned to distinguish between different classes of Alzheimer's Disease progression, demonstrating robust performance and convergence during training.