Model Development Phase Template

Date	19 July 2024
Team ID	SWTID1720110092
Project Title	Beneath the Waves: Unraveling Coral Mysteries through Deep Learning
Maximum Marks	5 Marks

Model Selection Report

In the model selection report for future deep learning and computer vision projects, various architectures, such as CNNs or RNNs, will be evaluated. Factors such as performance, complexity, and computational requirements will be considered to determine the most suitable model for the task at hand.

Model Selection Report:

Model	Description
Main Model	• **EfficientNetB0 Backbone**: ◦ **Architecture**: EfficientNetB0 is a convolutional neural network (CNN) that balances network depth, width, and resolution for efficient performance. ◦ **Pre-trained Weights**: The model uses pre-trained weights from ImageNet, which helps leverage learned features from a large, diverse dataset. ◦ **Excluding Top Layers**: The top (fully connected) layers are excluded (`include_top=False`) to allow custom layers to be added for the specific classification task. • **Fine-Tuning**:

- **Partial Freezing**: The first 100 layers of EfficientNetB0 are frozen to retain their pre-trained weights, while the remaining layers are trainable to adapt to the new dataset.

- **Custom Layers**:
  - **Input Layer**: Defines the shape of the input images.
  - **Base Model**: The EfficientNetB0 base model processes the input images.
  - **Global Average Pooling**: Reduces the spatial dimensions of the feature maps, summarizing the presence of features.
  - **Dropout Layers**: Adds regularization by randomly setting a fraction of input units to 0 at each update during training to prevent overfitting.

Dense Layer: Fully connected layer with 128 units and ReLU activation.

  - **Another Dropout Layer**: Adds further regularization.
  - **Output Layer**: Dense layer with 2 units (for binary classification) and softmax activation to output class probabilities.

- **Model Compilation**:
  - **Optimizer**: Adam optimizer with a learning rate of 0.0001, known for its adaptive learning rate capabilities.
  - **Loss Function**: Categorical cross-entropy, suitable for multi-class classification tasks.
  - **Metrics**: Accuracy, to monitor the fraction of correctly classified samples.

- **Data Augmentation and Generators**:
  - **ImageDataGenerator**: Used for data augmentation to improve model generalization by creating variations of the training images.
  - **Data Generators**: Generate batches of tensor image data with real-time data augmentation for training and testing.

- **Model Training**:

	• **Training**: The model is trained for a specified number of epochs with training and validation data generators. • **Model Evaluation and Saving**: • **Saving the Model**: The trained model is saved to a file. • **Evaluation**: The model's performance is evaluated on the test set, and test accuracy is printed.