**Model Development Phase Template**

|  |  |
| --- | --- |
| Date | 20 June 2025 |
| Team ID | SWTID1749906821 |
| Project Title | neural networks ahoy: cutting-edge ship classification for maritime |
| 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** |
| Model 1 | * Use **Depthwise Separable Convolution** layers (DepthwiseConv2D + SeparableConv2D) — this is a lightweight and efficient CNN architecture pattern, inspired by MobileNet. * Include **Average Pooling** after convolution layers to reduce spatial dimensions. * Use a **fully connected (Dense)** layer of 128 neurons followed by a **Softmax output layer** for multi-class classification. * Use **Global Average Pooling** before the Dense layers. * Use **Adam optimizer** with a learning rate of 0.0001. * Are trained with the same model structure but for a different number of epochs. * Trained for **15 epochs** * Contains **Dropout layers** after **each convolutional block** (after every AveragePooling2D layer). * Dropout rate: 0.5 * Purpose: Helps in **regularization** and **prevents overfitting** during shorter training. * May **generalize better** on small datasets or when training for fewer epochs. |
| Model 2 | * Use **Depthwise Separable Convolution** layers (DepthwiseConv2D + SeparableConv2D) — this is a lightweight and efficient CNN architecture pattern, inspired by MobileNet. * Include **Average Pooling** after convolution layers to reduce spatial dimensions. * Use a **fully connected (Dense)** layer of 128 neurons followed by a **Softmax output layer** for multi-class classification. * Use **Global Average Pooling** before the Dense layers. * Use **Adam optimizer** with a learning rate of 0.0001. * Are trained with the same model structure but for a different number of epochs. * Trained for **35 epochs** * **Dropout after first block is commented out**, only included after the second block. * Less dropout overall compared to Model 1. * Potentially better at **learning finer details** (due to fewer dropped connections), especially over a **longer training time**. |
| Model 3 | * Input Preprocessing (Rescales pixel values from [0, 255] to [0, 1].) * Conv2D Layers (4 convolutional layers with increasing filters: 16 → 32 → 64 → 128, all using ReLU activation and kernel size (3, 3).) * MaxPooling2D (Follows each Conv2D layer to reduce spatial dimensions.) * Dropout (Dropout after 2nd and 3rd conv blocks (rate: 0.5) for regularization.) * Flatten + Dense (Final layers: Flatten → Dense(128, relu) → Dense(5, softmax). ) * Loss Function (SparseCategoricalCrossentropy — used when labels are integers (not one-hot encoded). ) * Callbacks Includes ReduceLROnPlateau and EarlyStopping for better convergence and training control. * Training Epochs (Trained for up to 10 epochs (with early stopping)) |