

## Model Optimization and Tuning Phase Template

Date	27 October 2024
Team ID	739755
Project Title	Bird Species Classification
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 (8 Marks):

Model	Tuned Hyperparameters
Model 1	<pre> history = model.fit(     train_generator,     validation_data=val_generator,     epochs=10 # Adjust the number of epochs as needed ) </pre> <p>Epoch 1/10    /usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your "PyDatasetAdapter" class is not a subclass of "PyDatasetAdapter".    self._warn_if_super_not_called()</p> <p>112/112 ————— 83s 632ms/step - accuracy: 0.0166 - loss: 5.6958 - val_accuracy: 0.0166 - val_loss: 4.5540    Epoch 2/10    112/112 ————— 69s 585ms/step - accuracy: 0.0130 - loss: 4.6233 - val_accuracy: 0.0144 - val_loss: 4.3269    Epoch 3/10    112/112 ————— 73s 619ms/step - accuracy: 0.0160 - loss: 4.4720 - val_accuracy: 0.0354 - val_loss: 4.2400    Epoch 4/10    112/112 ————— 68s 579ms/step - accuracy: 0.0275 - loss: 4.3779 - val_accuracy: 0.0542 - val_loss: 4.1719    Epoch 5/10    112/112 ————— 81s 570ms/step - accuracy: 0.0286 - loss: 4.2953 - val_accuracy: 0.0619 - val_loss: 4.0097    Epoch 6/10    112/112 ————— 69s 574ms/step - accuracy: 0.0332 - loss: 4.2088 - val_accuracy: 0.0564 - val_loss: 3.9751    Epoch 7/10    112/112 ————— 68s 573ms/step - accuracy: 0.0370 - loss: 4.1482 - val_accuracy: 0.0531 - val_loss: 3.9572    Epoch 8/10</p>

```
from tensorflow.keras import layers, models
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(210, activation='softmax'))
model.summary()
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential_2"
```

Layer (type)	Output Shape	Params #
conv2d_3 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_2 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_4 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_3 (MaxPooling2D)	(None, 54, 54, 64)	0
flatten_1 (Flatten)	(None, 186624)	0
dense_3 (Dense)	(None, 128)	23,888,000
dropout_1 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 210)	27,000

```
import os

# Path to the train and validation directories
train_dir = '/content/drive/MyDrive/Bird Species Classification/manasa/train'
val_dir = '/content/drive/MyDrive/Bird Species Classification/manasa/val'

# Check the structure
print("Training Classes:", os.listdir(train_dir))
print("Validation Classes:", os.listdir(val_dir))

Training Classes: ['002.Laysan_Albatross', '001.Black_footed_Albatross', '003.Sooty_Albatross', '004.Gl
Validation Classes: ['002.Laysan_Albatross', '001.Black_footed_Albatross', '003.Sooty_Albatross', '004.
```

### Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Model	<p>Preprocessing: Resizing and Normalization</p> <p>Data Augmentation: Random rotation, flipping, zooming</p> <p>Loss Function: Categorical Crossentropy</p>

	<p>Optimizer: Adam Optimizer</p> <p>Learning Rate Scheduler: ReduceLROnPlateau</p> <p>Epochs 10</p> <p>Batch Size : 32</p> <p>Evaluation Metric: Accuracy</p>
--	---