**SWS3009A Deep Learning**

**Assignment Answer Book**

Name 1: \_Chen Siyuan\_\_\_\_

Name 2: \_Pang Bo\_\_\_\_\_\_\_\_

Name 3 (if any): \_\_\_\_\_\_\_\_\_\_\_\_

**IMPORTANT: THIS REPORT IS DUE ON 8 JULY 2025 BUT YOU MUST FINISH EVERYTHING ELSE IN THIS ASSIGNMENT BEFORE YOUR BASELINE EVALUATION!**

Complete this answer book and save to PDF before uploading to Canvas. Deadline is 11.59 pm on 8 July 2025.

1. Fill in the number of images you’ve gotten for each of the species:

Ragdolls: \_\_\_\_723\_\_\_\_\_\_\_\_\_\_\_\_\_ images

Singapura: \_\_\_726\_\_\_\_\_\_\_\_\_\_\_\_\_ images

Persians: \_\_\_\_782\_\_\_\_\_\_\_\_\_\_\_\_\_ images

Sphynx: \_\_\_\_\_721\_\_\_\_\_\_\_\_\_\_\_\_\_ images

Pallas Cats: \_\_\_\_872\_\_\_\_\_\_\_\_\_ images

1. Description of our architecture and justification:

#### **Baseline and Architecture**

Our model is designed for the classification of five distinct cat breeds. The architecture is built upon a **fine-tuned ResNet50** model, pre-trained on ImageNet, which serves as a powerful feature extractor and is especially suitable for the cat-recognizing task.

A key component of our baseline is a unique pre-processing step that leverages OpenCV's **Haar Cascades** for cat face detection. This tackles the problem that CNNs such as Resnet cannot automatically find a bounding box outside of the chaotic environment like YOLO does. However, CNN + extra preprocessing (e.g. our Haar Cascades) attempting to find useful features out of the whole pictures works as well. For each input image, the system first attempts to identify a cat's face.

* If a face is successfully detected, the image is cropped to this region of interest, which is then passed to the model. This focuses the classifier on the most relevant features of the cat.
* If no face is detected, the original, un-cropped image is used for classification.

To effectively adapt the pre-trained model to our specific task, we employ a **three-stage fine-tuning strategy**:

1. **Stage 1: Feature Head Training:** Initially, all layers of the ResNet50 base are frozen. Only the newly added, custom classification layers (the "head") are trained. This allows the head to learn the specifics of our dataset without disrupting the robust, pre-trained weights in the base model, using a relatively high learning rate.
2. **Stage 2: Partial Fine-Tuning:** We then unfreeze the top 10 layers of the ResNet50 model and continue training with a lower learning rate (automatically adjusted by a callback function written by us). This step fine-tunes the higher-level feature representations in the base model to become more relevant to our cat breeds.
3. **Stage 3: Full Model Fine-Tuning:** Finally, the entire model is unfrozen, and we train all layers with a very low learning rate. This allows for small, holistic adjustments across the entire network, achieving optimal performance.

To support this process, we utilize and customize Keras callbacks such as **EarlyStopping** to prevent overfitting by halting training when validation performance ceases to improve, and **ReduceLROnPlateau / LearningRateScheduler** to automatically adjust the learning rate for more stable and effective convergence.

#### **Validation Strategy**

Our validation protocol is designed to provide a robust and detailed assessment of the model's performance. We use a dedicated test set of images that the model has not seen during training.

The core of our evaluation is as follows:

1. **Ground Truth from Filename:** The true label for each image in the test set is encoded directly into its filename.
2. **Prediction and Comparison:** The baseline model processes each image and outputs a predicted class. We then programmatically extract the true label from the filename and compare it to the model's prediction.
3. **Comprehensive Metrics:** To thoroughly evaluate performance and guide future improvements, we calculate and analyze several key metrics:
   * **Per-class accuracy** to understand how well the model performs on each individual cat breed.
   * A **Confusion Matrix** to visualize misclassifications between classes.
   * The **F1-Score** to provide a balanced measure of precision and recall.
4. Results:

Training Accuracy: \_\_\_\_98.54\_\_\_\_\_\_\_\_ (%)

Validation Accuracy: \_\_\_97.63\_\_\_\_\_\_\_\_ (%)

Is there any overfitting? How do you know?

No, our model does not show significant signs of overfitting. We know this by analyzing the learning curves from our training process and through the proactive measures we took to prevent it.

Is there any underfitting? How do you know?

No, our model does not suffer from underfitting. We are confident about this because the model was able to achieve high performance on all the training, validation, and final test sets.