**DTS101TC Coursework**

This coursework is designed to assess your understanding of neural networks and machine learning concepts, as well as your ability to implement, analyze, and evaluate models effectively. It consists of two main components: five assignments and an image object detection project. Detailed instructions, marking criteria, and submission requirements are outlined below. AIGC tools are not allowed.

**Part 1: Assignments (50 Marks)**

This section includes five individual assignments, each focusing on different neural network techniques and datasets. The breakdown for each task includes marks for code execution, analysis, evaluation, and reporting quality.

**Submission Requirements**

* Please submit your notebooks to Gradescope. Each assignment must be completed according to the instructions provided in the Python Jupyter Notebook, with all output cells saved alongside the code. You don’t need to write a report for this part. Please put all the analysis and results in your notebook.
* Weekly TA checks during lab sessions and office hours are mandatory. **Assignments will not be graded without TA verification.**

**Question 1: Digit Recognition with Neural Networks**

* **Task**: Implement a basic neural network using TensorFlow/PyTorch to train a digit recognition model on the MNIST dataset.
* **Mark Breakdown**:
  + Code execution by Gradescope: 5 marks
  + Data and model analysis: 2 marks
  + Test cases: 2 marks
  + Report quality (comments and formatting): 1 mark

**Question 2: Logistic Regression for Flower Classification**

* **Task**: Build and implement a Logistic Regression model to classify three types of iris flowers using the dataset in sklearn.
* **Mark Breakdown**:
  + Code execution by Gradescope: 5 marks
  + Data and model analysis: 2 marks
  + Test cases: 2 marks
  + Report quality (comments and formatting): 1 mark

**Question 3: House Price Prediction with ANN/MLP**

* **Task**: Design and implement an ANN/MLP model to predict house prices in California using the dataset in sklearn.
* **Mark Breakdown**:
  + Code execution by Gradescope: 5 marks
  + Data and model analysis: 2 marks
  + Test cases: 2 marks
  + Report quality (comments and formatting): 1 mark

**Question 4: Stock Price Prediction with RNN**

* **Task**: Create an RNN model to predict stock prices for companies like Apple and Amazon from the Nasdaq market using the provided dataset.
* **Mark Breakdown**:
  + Code execution by Gradescope: 5 marks
  + Data and model analysis: 2 marks
  + Model evaluation: 2 marks
  + Report quality (comments and formatting): 1 mark

**Question 5: Image Classification with CNN**

* **Task**: Develop a CNN model to classify images into 10 classes using the CIFAR-10 dataset.
* **Mark Breakdown**:
  + Code execution by Gradescope: 5 marks
  + Data and model analysis: 2 marks
  + Model evaluation: 2 marks
  + Report quality (comments and formatting): 1 mark

**Part 2: Project (50 Marks)**

The project involves building a custom image dataset and implementing an object detection neural network. This is a comprehensive task that evaluates multiple skills, from data preparation to model evaluation.

**Submission Requirements**

* All of your dataset, code (Python files and ipynb files) should be a package in a single ZIP file, with a PDF of your report (notebook with output cells, analysis, and answers). INCLUDE your dataset in the zip file.

**Step 1: Dataset Creation (10 Marks)**

* **Task**: Collect images and use tools like Label Studio or LabelMe to create labeled datasets for object detection. You can add one more class into the provided dataset. The dataset should have up to 10 classes. Each contains at least 200 images.
* **Deliverable**: Include the dataset in the ZIP file submission.
* **Mark Breakdown**:
  + Correct images and labels: 6 marks
  + Data collection and labeling process explanation: 2 marks
  + Dataset information summary: 2 marks

**Step 2: Data Loading and Exploration (10 Marks)**

* **Task**: Organize data into train, validation, and test sets. Display dataset statistics, such as class distributions, image shapes, and random samples with labels. Randomly plot 5 images in the training set with their corresponding labels.
* **Mark Breakdown**:
  + Correct dataset splitting: 6 marks
  + Dataset statistics: 2 marks
  + Sample images and labels visualization: 2 marks

**Step 3: Model Implementation (10 Marks)**

* **Task**: Implement an object detection model, such as YOLOv8. Include a calculation of the total number of parameters in your model. You must include calculation details.
* **Mark Breakdown**:
  + Code and comments: 6 marks
  + Parameter calculation details and result: 4 marks

**Step 4: Model Training (10 Marks)**

* **Task**: Train the model using appropriate hyperparameters (e.g., epoch number, optimizer, learning rate). Visualize training and validation performance through graphs of loss and accuracy.
* **Mark Breakdown**:
  + Code and comments: 6 marks
  + Hyperparameters analysis: 2 marks
  + Performance analysis: 2 marks

**Step 5: Model Evaluation and Testing (10 Marks)**

* **Task**: Evaluate the model on the test set, displaying predictions (visual result) and calculating metrics like mean Average Precision (mAP) and a confusion matrix.
* **Mark Breakdown**:
  + Code and comments: 6 marks
  + Prediction results: 2 marks
  + Evaluation metrics: 2 marks

**Submission Guidelines**

1. **Assignments**: Submit your Jupyter Notebooks via Gradescope. Ensure all output cells are saved and visible.
2. **Project**: Submit your ZIP file containing the dataset, Python files, Jupyter Notebooks, and a PDF report via Learning Mall Core.

**General Notes and Policies**

1. **Plagiarism**: Submissions must be your own work. Avoid copying from external sources without proper attribution. Sharing code is prohibited.
2. **Late Submissions**: Follow the university's policy on late submissions; penalties may apply.
3. **Support**: Utilize lab sessions and TA office hours for guidance.

**Marking Criteria**

**Assignments**

* Code execution by Gradescope: 5 marks
* Data and model analysis: 2 marks
* Test cases or model evaluation: 2 marks
* Report quality (comments and formatting): 1 mark

**Project**

* Code (60%):
  + Fully functional code with clear layout and comments: 6 marks
  + Partially functional code with some outputs: 4 marks
  + Code that partially implements the solution but does not produce outcomes: 2 marks
  + Incomplete or non-functional code: 0 marks
* Analysis (40%):
  + Complete and accurate answers with clear understanding: 4 marks
  + Partial answers showing some understanding: 2 marks
  + Limited understanding or incorrect answers:: 0 marks