|  |  |
| --- | --- |
| **Module** | Advanced Topics in Computer Vision and Deep Learning Coursework Assignment: EEEM071 |
| **Assessment type** | Coursework assignment |
| **Weighting** | 25% |
| **Deadline** | 4:00PM, Friday, 11 April 2025 (Week 10) |
| **STUDENT URN** |  |
| **STUDENT Name** |  |

**Objectives:**

* Learn and become familiar with the process of sequential hyperparameter tuning of an entire deep learning pipeline.
* Develop intuition via experimentation on how to use such model design and parameter tuning to improve the performance of a pipeline.
* Interpret experimental results of deep learning experiments and their implications.

**Outcomes:**

* Navigate general deep learning codebases and modify according to requirements.
* Empirically suggest reasonable values for hyperparameters and select the ones enabling the best performance.
* Understand the implications of changing the backbone architecture of a pipeline.
* Reason why data augmentation works.
* Understand the implications and importance of selecting the best learning rate and batch size.
* Understand how to report experimental results professionally. This includes using tables, plots, and diagrams to communicate ideas and results.

### Vehicle Re-identification

For this coursework assignment, you are encouraged to apply what you have learnt from the module materials and your participation in collaborative learning activities like lab sessions and discussions.

#### Guidelines

* Submit your assignment as a **Word or PDF document**, along with the supporting evidence as required, via the SurreyLearn submission page before the deadline. Do not submit any other unrelated documents that could flood your submission and mislead the marking process. Only the latest version submitted before the deadline will be assessed if multiple versions exist.
* Label your submission file as: EEEM071-[insert your URN number]-Assignment e.g., EEEM071-1234567-Assignment
* Include a reference list at the end of your assignment, and use the [Harvard](https://library.surrey.ac.uk/subject-resources/referencing) or [IEEE](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf) referencing style when citing sources. Please use the selected option consistently.
* **Word limit:** Each section of the report should not exceed **200 words** (excluding tables, plots, and graphs). Penalties of up to 50% may be applied for unnecessarily long reports.
* This assignment requires time and effort. It is recommended that you start working on it early to minimise the possibility of experiencing stress around it and avoid missing the deadline.
* You will receive both an overall mark and written feedback on your assignment.
* Extensions for this assignment will only be granted under exceptional circumstances. Extension requests must be submitted via the Extenuating Circumstances (ECs) process, as tutors are not permitted to approve ad hoc extension requests.

**Note:**

Please follow Surrey’s guidelines for [avoiding plagiarism](https://exams.surrey.ac.uk/academic-integrity-and-misconduct/plagiarism#:~:text=Avoiding%20and%20detecting%20plagiarism&text=Using%20entirely%20your%20own%20words,writing%20directly%20from%20source%20material).

You must complete this coursework individually. Copying code or text from the internet or another student and including it in your assignment without proper attribution constitutes plagiarism.

Undetected plagiarism undermines the quality of your degree by hindering the assessor’s ability to evaluate your work fairly and prevents you from learning through the coursework. If plagiarism is suspected, you will be referred to an Academic Misconduct Panel, which may result in academic sanctions.

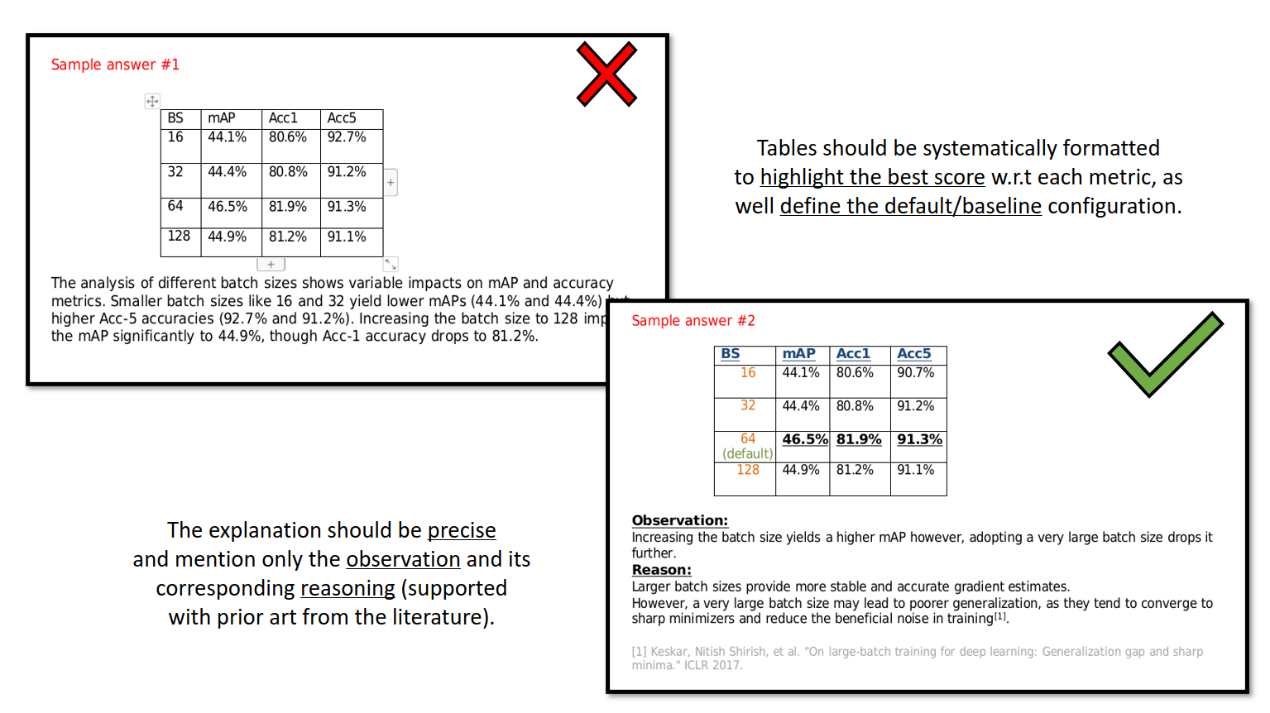
All assignments will be checked for plagiarism using Turnitin. Therefore, it is important that you correctly reference sources that have informed your work where appropriate

###### Hyperparameters

For the parameters that are not learnable, you can set them before starting the training.

###### Format

* Ensure your responses to each question are on the spot as shown in Figure 1. Focus on the specific question and not move away.



**Figure 1:** Response example.

###### Log files

* Submit your own log files from your codebase. You are required to submit a separate log file for each experiment-based question.
* Do not change the log file structure. Each log file is watermarked and unique to each run.
* Name each log file according to the section number and question number: (log\_sec{Section\_num}\_q{Question\_num}.txt). For example, the log file generated for an experiment that corresponds to Section 1 Question 2, should be named as “log\_sec1\_q2.txt”.

###### Presentation and clarity

In addition to the technical content, the presentation and clarity of your writing will be assessed.

###### Model performance metrics

While different metrics are available, it is recommended that you prioritise mAP when selecting the best-performance models.

##### Start your assignment

Write your assignment in the designated space provided below. Each of the sections outlines the tasks required and specifies the information you must include in your report for that section.

##### Section 1: Familiarity with the code provided (30 marks)

1. Run the code using the default settings.Discuss the training and evaluation process (*mention the loss functions used*) followed by implications of the observed performance using an appropriate metric. (10 marks)
2. Apply another CNN variant (that is not provided in the default settings). Critically discuss and contrast the results with what observed in question 1 above. (10 marks)
3. Apply one more neural network architecture not from the same family as the above questions. Critically discuss and contrast the results with what observed questions 1 and 2 above. (10 marks)

**Note:**

For Questions 2 and 3, ensure the new architecture is explicitly specified in your shell script command if you are using one from the codebase.

*Start writing here:*

##### Section 2: Dataset preparation and augmentation experiment (25 marks)

Begin with the default data augmentation setting with random, horizontal flip and Random2DTranslation.

1. Further append on top two additional data augmentation techniques, one at a time e.g., Default + “crop”, Default + “horizontal flip”, and Default + “blurring”. Compare the results with the default configuration in the provided code and discuss the differences in performance. (20 marks)
2. Combine the augmentation techniques from Question 1 to find the best-performing combination. Highlight any improvement or drop in the overall score. (5 marks)

*Start writing here:*

##### Section 3: Exploration of Hyperparameters (25 marks)

Start with the default learning rate (LR) and batch size (BS).

1. Exploration of Learning Rate (LR). (10 marks)
   1. Experiment with 4 values of LR (in addition to the default value).
   2. Discuss the observed impact of each value on overall performance.
2. Exploration Batch sizes. (10 marks)
   1. Fixing the best LR value from the experiments in question 1 above, test 4 different values of the BS in addition to the default value.
   2. Discuss the impact observed on overall performance.
3. Exploration of the optimizer. (5 marks)
4. Fixing the best Learning Rate value and best Batch Size value from the experiments in Questions 1 and 2, respectively, test with changing the optimizer to SGD, using PyTorch’s internal class.
   1. Discuss the impact observed on overall performance.

*Start writing here:*

### **Section 4: Summary on overall hyper-parameter tuning, only need to fill up the table below, no more text. (10 marks)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Hyper-parameter** | **Best configuration (obtained)** | **mAP** | **Rank-1** |
| 1 | Architecture |  |  |  |
| 2 | Data augmentation |  |  |  |
| 3.1 | Learning rate |  |  |  |
| 3.2 | Batch size |  |  |  |
| 3.3 | Optimizer |  |  |  |

### Rubric

#### Section 1: Familiarity with the code provided (30 marks)

##### Task 1.1: Running and understanding the code (10 marks)

* **Understanding the code execution** 
  + *Criteria*: Clear understanding and explanation of the code flow and processes.
  + *Indicators*: Detailed description of data loading, model initialisation, training loop, and evaluation steps.

**(5 marks)**

* **Metric explanation and performance analysis** 
  + *Criteria*: Appropriate choice and understanding of the evaluation metric(s) (e.g., accuracy, mAP).
  + *Indicators*: Correct calculation and interpretation of the metric values. Discussion on what these values imply about the model's performance.

**(3 marks)**

* **Discussion of observed performance** 
  + *Criteria*: Insightful discussion on the observed results.
  + *Indicators*: Analysis of performance patterns, potential overfitting/underfitting issues, and suggestions for improvement.

**(2 marks)**

##### Task 1.2: Applying a different CNN variant (10 marks)

* **Implementation of a new CNN variant** 
  + *Criteria*: Correctly modifying the code to implement a different CNN architecture.
  + *Indicators*: Model architecture is different from the provided one, and code runs without errors.

**(4 marks)**

* **Comparison and critical analysis** 
  + *Criteria*: Critical comparison between the original and new CNN results.
  + *Indicators*: Use of quantitative results to support comparison, discussion on potential reasons for performance differences.

**(6 marks)**

##### Task 1.3: Applying another neural network architecture (10 marks)

* **Implementation of a new neural network** 
  + *Criteria*: Correctly implementing a neural network architecture different from those in Tasks 1.1 and 1.2.
  + *Indicators*: Model architecture is distinct, and code runs correctly.

**(4 marks)**

* **Comprehensive comparison and analysis** 
  + *Criteria*: Deep comparative analysis between all three architectures.
  + *Indicators*: Insightful discussion on the strengths and weaknesses of each model, based on quantitative results and qualitative observations.

**(6 marks)**

#### Section 2: Dataset preparation and augmentation experiment (25 marks)

##### Task 2.1: Augmentation techniques (20 marks)

* **Implementation of additional augmentations** 
  + *Criteria*: Successful addition of two new augmentation techniques.
  + *Indicators*: Correct application and explanation of chosen techniques, code runs without errors.

**(8 marks)**

* **Performance comparison and analysis** 
  + *Criteria*: Comparative analysis of the impact of each augmentation setting.
  + *Indicators*: Clear explanation of performance differences between default and new settings, supported by quantitative results.

**(8 marks)**

* **Discussion on results and insights** 
  + *Criteria*: Insightful interpretation of how augmentations affected model performance.
  + *Indicators*: Discussion on potential reasons for performance changes and implications for model robustness.

**(4 marks)**

##### Task 2.2: Best-performing augmentation combination (5 marks)

* **Combining techniques and experimentation** 
  + *Criteria*: Correct implementation of combined augmentation techniques.
  + *Indicators*: Code successfully combines the techniques and executes without errors.

**(3 marks)**

* **Analysis of improvement or decline** 
  + *Criteria*: Detailed discussion on whether the combined augmentation improved performance.
  + *Indicators*: Use of appropriate metrics to highlight any performance change, supported by logical reasoning.

**(2 marks)**

#### Section 3: Exploration of hyperparameters (25 marks)

##### Task 3.1: Learning rate exploration (10 marks)

* **Experimentation with different LR values** 
  + *Criteria*: Correct experimentation with three additional LR values.
  + *Indicators*: Clear documentation of the chosen LR values and implementation.

**(4 marks)**

* **Performance analysis of each LR** 
  + *Criteria*: Thorough analysis of the effect of each LR value on performance.
  + *Indicators*: Use of metrics to compare performance across LR values, discussion on the reasons behind observed changes.

**(6 marks)**

##### Task 3.2: Batch size exploration (10 marks)

* **Experimentation with different BS values** 
  + *Criteria*: Correct experimentation with three additional BS values.
  + *Indicators*: Clear documentation of the chosen BS values and implementation using the best-performing LR from Task 3.1.

**(4 marks)**

* **Performance analysis of each BS** 
  + *Criteria*: Detailed analysis of the impact of each BS value on performance.
  + *Indicators*: Use of metrics to compare performance across BS values, discussion on the reasons behind observed changes.

**(6 marks)**

##### Task 3.2: Optimizer exploration (5 marks)

* **Experimentation with different optimizers** 
  + *Criteria*: Correct experimentation with an additional optimizer.
  + *Indicators*: Clear documentation of the chosen optimizer and implementation using the best-performing LR and BS.

**(2 marks)**

* **Performance analysis of each BS** 
  + *Criteria*: Detailed analysis of the impact of optimizer on performance.
  + *Indicators*: Use of metrics to compare performance across optimizers, discussion on the reasons behind observed changes.

**(3 marks)**

#### Section 4: Completing the summary Table (10 marks)

* **Consistency with Experimental Results**
  + Criteria: The hyperparameters listed must align with findings from previous sections. Each for **2 marks**.
  + Indicators: No contradictions between reported values and earlier discussion.

##### Report writing and presentation (10 marks)

* **Figures should be well-presented, with readable axes and labels.**

**(2 marks)**

* **The writing should be clear, grammatically correct, and easy to follow.**

**(6 marks)**

* **Tables are used when discussing results involving multiple hyperparameters values** 
  + *Criteria*: Clarity, structure, and depth of the written report.
  + *Indicators*: Logical flow, clear explanation of methods and results, proper use of figures/tables to support the analysis.

**(2 marks)**

**Total: 100 marks**