Project-Based Assessment: Image classification using hand-crafted features, neural network and convolutional neural network approaches

Assignment hand-out: 6th January 2025

Hand-in: **Full-time students**: 3rd February 2025 at 8 am

Part-time students (PGCert): 10th February 2025 at 8 am

You will receive feedback within 10 working days of the submission of the assessment.

Contribution towards module: 100% (and scored out of 100 marks). This is an individual assessment. When submitting your assignment, you are agreeing to the following statement:

I certify that the submission is my own work, all sources are correctly attributed, and the contribution of any AI technologies is fully acknowledged.

The university policy on plagiarism is available at:

https://www.qub.ac.uk/directorates/sgc/learning/LearningResources/Plagiarism/

Keep a copy of all submitted coursework – i.e., your computer files. **Remember plagiarism is a serious offense** – ensure that you use your own words even when referencing a source. Also reference libraries, tutorials, or code that you used. This assignment is to be completed individually.

Assignment Brief:

You will work on an image classification project using hand-crafted features, neural network, and convolutional neural network approaches on TinyImageNet100 dataset, which is uploaded to the Canvas. This dataset has 100 classes containing 500 images each.

This coursework aims to assess your ability to build image classification solutions using hand-crafted features, neural network, and convolutional neural network approaches that showcase your proficiency in different computer vision techniques learned and practiced in this module.

Project Phases:

Phase 1: Dataset Preparation

- You can choose 3, 5, 10, or 15 classes from the TinyImageNet100 dataset to build your dataset. Please choose the first 400 images (ordered by alphabet) as the training images and the remaining 100 images as the testing images for each class.

- Provide a function to randomly choose a selected number of classes from the TinyImageNet100 dataset if applicable and prepare training & testing data.

Phase 2: Conduct image classification using the hand-crafted feature approach

- Select one (or two) image local features, e.g., SIFT or ORB

- Build BoW model to extract mid-level features

- (Optional) Build Fisher vector feature encoding approach to extract mid-level features (refer to https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_fisher_vector.html#sphx-glr-auto-examples-features-detection-plot-fisher-vector-py)

- Train linear SVM model using mid-level features

- Evaluate the image classification performance using the trained linear SVM model

Phase 3: Conduct image classification using the neural network approach

- Based on the number of classes you choose, build a proper neural network structure, e.g., select a different number of layers

- Train the neural network

- Evaluate the image classification performance using the trained neural network

Phase 4: Conduct image classification using the convolutional neural network approach

- Based on the number of classes you choose, build a proper convolutional neural network structure, e.g., select a different number and types of layers

- Train the convolutional neural network

- Evaluate the image classification performance using the trained convolutional neural network

Phase 5: Discussion and interpretation

- Compare and discuss the difference between these three approaches, including advantages and disadvantages.

- Record the training procedure of the neural network and convolutional network, and discuss the parameters tuning in the training procedure in order to get better performance.

Project Deliverables:

1. Python script file or Jupyter Notebook file containing all source code and trained modes from three different approaches

Your source code should be well-documented, structured, and include appropriate comments. The source code should be run directly without further unnecessary modification for assessment. The trained modes can be used to obtain the same performance as in the report.

2. Report

You are required to submit a project report not exceeding 3000 words. The project requires you to create a comprehensive report that covers all aspects of the project. This report should include various sections including 1) Introduction, 2) Methodology, 3) Results, 4) Discussions and 5) Conclusion & Word Count. Specifically, elaborate on the outcomes and insights obtained from each Mathematical Foundation Application. You should provide your rationale for selecting the parameters of different approaches. To enhance the report's clarity, incorporate visualisations, tables, and code snippets that effectively support the analysis and findings you present. Throughout the report, please ensure to provide appropriate references and sources to support your methodology, analysis, and findings. Proper citation of relevant literature and online sources is essential to demonstrate the credibility and validity of your work. The word count excludes Figures, Tables, Titles, References, Code and Cover page.

A margin of +10% of the size limit will be allowed before the penalties for exceeding the word count apply.

Penalties for exceeding the word count follow the University Guidance and are as follows:

• +10%: no penalty

• +>10% - 20%: 10% penalty

• +>20% - 30%: 20% penalty

• +>30% - 40%: 30% penalty

• +>40% - 50%: 40% penalty

• +>50%: maximum mark of 50% for the assignment.

• All deductions won't lower the assignment grade below the passing threshold.

3. Video Presentation

You are also required to submit a 5-minute video presentation that summarises the crucial aspects and findings of your project. In the video, ensure you provide a clear and thorough explanation of the dataset used, different approaches for image classification, how to improve the image classification performance of different approaches, and present the significant outcomes you achieved. Your video presentation should effectively communicate the essence of your project's journey and findings.

4. Submission

The report will be electronically submitted online using Canvas. Your submission should consist of a .zip file containing the report along with all other relevant files. Python code must be copied directly from your Python script file or Jupyter Notebook file as an appendix of the report for plagiarism check. In addition, please submit the Python script file or Jupyter Notebook file separately as part of your submission. You will be given an individual mark based on the assessment criteria. After submission, please double-check that all submitted work has been uploaded correctly on Canvas.

Standard university penalties apply for late submission -- penalised at the rate of 5% of the total mark awarded for that component for each calendar day late up to a maximum of five calendar days, after which a mark of zero shall be awarded, i.e., up to one calendar day is -5%; up to two calendar days is -10%; up to three calendar days is - 15%, etc. This excludes University closure days.

https://www.qub.ac.uk/directorates/AcademicStudentAffairs/AcademicAffairs/GeneralRegulations/StudyRegulations/StudyRegulationsforPostgraduateTaughtProgrammes/

Assessment Criteria	Marks
Dataset selection and preparation Outstanding (9-10 marks): Randomly choose 15 classes and load the dataset correctly for the experiments Good (7-8 marks): Randomly choose 10 classes and load the dataset correctly for the experiments. Acceptable (5-6 marks): Randomly choose 5 classes and load the dataset correctly for the experiments.	10 marks

Poor (3-4 marks): Randomly choose 3 classes and load the dataset correctly for the experiments. Very Poor (0-2 marks): Fail to load the dataset properly for the experiments, e.g., mix the training data and testing data	
Hand-crafted features approach Outstanding (17-20 marks): Use both two local features, i.e., SIFT and ORB, as two approaches. Build the BoW model and the Fisher vector feature encoding approach correctly. Train and evaluate the linear SVM model correctly, and achieve good performance. Good (13-16 marks): Use both two local features, i.e., SIFT and ORB, as two approaches. Build the BoW model correctly. Train and evaluate the linear SVM model correctly, and achieve reasonable performance. Acceptable (9-12 marks): Use one local feature, i.e., SIFT or ORB, as one approach. Build the BOW model correctly. Train and evaluate the linear SVM model correctly, and achieve acceptable performance. Poor (5-8 marks): Use one local feature, i.e., SIFT or ORB, as one approach. Build the BoW model with some minor mistakes. Train and evaluate the linear SVM model with some minor mistakes. Very Poor (0-4 marks): Use one local feature, i.e., SIFT or ORB, as one approach. Build the BoW model with some major mistakes. Train and evaluate the linear SVM model with some major mistakes.	20 marks
Neural network approach Outstanding (13-15 marks): Select the different number of layers to build a proper neural network. Apply suitable parameters and training tricks, such as suitable learning rate and data augmentation methods, etc, during training. Achieve good performance using this neural network. Good (10-12 marks): Select the different number of layers to build a proper neural network. Apply suitable parameters, such as suitable learning rate, etc, during	15 marks

training. Achieve reasonable performance using this neural network. Acceptable (7-9 marks): Build a neural network correctly. Apply suitable parameters, such as suitable learning rate, etc, during training. Achieve acceptable performance using this neural network. Poor (4-6 marks): Build a neural network with some minor mistakes. Train this neural network with some minor mistakes. Very Poor (0-3 marks): Fail to build a neural network or with some major mistakes. Fail to train this neural network or with some major mistakes.	
Convolutional neural network approach Outstanding (29-35 marks): Select the different number of layers with different types to build a proper convolutional neural network. Apply suitable parameters and training tricks, such as suitable learning rate and data augmentation methods, etc, during training. Achieve good performance using this convolutional neural network. Good (22-28 marks): Select the different number of layers with different types to build a proper convolutional neural network. Apply suitable parameters, suitable learning rate, etc, during training. Achieve acceptable performance using this convolutional neural network. Acceptable (15-21 marks): Build a convolutional neural network correctly. Apply suitable parameters, suitable learning rate, etc, during training. Achieve acceptable performance using this convolutional neural network. Poor (8-14 marks): Build a convolutional neural network with some minor mistakes. Train this neural network with some minor mistakes. Very Poor (0-7 marks): Fail to build a convolutional neural network or with some major mistakes. Fail to train this neural network or with some major mistakes.	35 marks
Discussion and interpretation	10 marks

Outstanding (9-10 marks): Profound insights derived; comprehensive analysis presented. Good (7-8 marks): Valuable insights drawn with thorough analysis. Acceptable (5-6 marks): Meaningful insights highlighted; analysis is sufficient. Poor (3-4 marks): Basic insights provided; analysis lacks depth. Very poor (0-2 marks): Insights minimal or missing; analysis insufficient.	
Project report quality and organisation Outstanding (9-10 marks): Exceptional organization with clear sections and logical flow. Good (7-8 marks): Well-structured report with coherent sections. Acceptable (5-6 marks): Adequate report structure with sections mostly organized. Poor (3-4 marks): Basic organization with some inconsistencies. Very poor (0-2 marks): Poorly organized report: sections lack coherence.	10 marks