COMP4318/5318 Assignment 2

Key information

This assignment is worth 25% of your final mark. It is a group assignment to be completed in pairs. Please ensure you have registered your pair on Canvas under the People tab, in either Assignment 2 Groups Section A or Assignment 2 Groups Section B per the announcements on Ed.

Please read the entire specification carefully before beginning the assignment, and refer back to it while working on your project. Please take special note of the information provided on Academic Integrity.

Deadline
11:59pm 20 October 2023 (Friday week 11)

Late submissions are allowed up to 3 days late, with a penalty of 5% of the maximum possible mark per calendar day. Late submissions after 3 calendar days will not be accepted.

Submission information
Three files are required to be submitted in the relevant submission portals on Canvas:

- Your report as a .pdf file
- Your jupyter notebook as a .ipynb file
- Your jupyter notebook as a .pdf file

A pdf of your jupyter notebook can be generated using File>Download as>PDF or Print Preview > Save as PDF.

Name your files with the following format:

- Report:
  - A2-report-SID1-SID2.pdf
- Code:
  - a2-code-SID1-SID2.ipynb
  - a2-code-SID1-SID2.pdf

where SID1 and SID2 are the SIDs of the two students in your pair. Please do not include your names anywhere in your submissions.

Please keep your report to a maximum of 12 pages of size 12 Times New Roman font (additional pages past this limit will not be marked). You may include references and an appendix with supplementary figures which are not included in this limit.

Code information
Your code for this assignment should be written in Python in a Jupyter Notebook environment. Please follow the structure in the template notebook provided. Your implementation of the algorithms should predominantly utilise the same suite of libraries we have introduced in the tutorials (Keras, scikit-learn, numpy, pandas etc.). Other libraries may be utilised for minor functionality such as plotting, however please specify any dependencies at the beginning of your code submission. While most of your explanation and justification

can be included in the report, please ensure your code is well formatted, and that there are sufficient comments or text included in the notebook to explain the cells.

You can choose to run your code locally or on a cloud service such as Google Colaboratory, however your final submission should be able to run on a local machine. Please submit your notebook with the cell output preserved and ensure that all results presented in your report are demonstrated in your submitted notebook.Your code may also be rerun by your marker, so please ensure there are no errors in your submitted code and it can be run in order.

Task Description

In this assignment, you will implement several machine learning algorithms to solve an image classification task, and compare their theoretical properties and experimental results thoroughly.

You will need to demonstrate your understanding of the full machine learning pipeline, including data exploration, preprocessing, model design, hyperparameter tuning, and interpreting results. Moreover, the assignment will require you to consolidate your knowledge from the course so far to effectively discuss the important differences between the algorithms.

While better performance is desirable, it is not the main objective of the assignment. Rather, it is important to fully justify your decisions and analyse the algorithms and the results. Please see the marking criteria at the end of the specification for how you will be assessed.

Code

Data loading, exploration, and preprocessing

The dataset to be used for this task is derived from BloodMNIST. This is a dataset containing 28x28 colour images of normal blood cells on a blood film, with each image containing a cell type of interest. You can read more about this dataset and also find more information about the original source here: https://www.nature.com/articles/s41597-022-01721-8 . This may help you discuss the data in your report.

The dataset is licensed under CC BY 4.0 (attribution is provided below).

We have provided different splits of the dataset than available for download at the MedMNIST site. *Please refer to the provided dataset on Canvas rather than downloading from this source.*

The images have relatively low dimensionality, which is intended to aid in keeping your runtimes short. You can increase/decrease the dimensionality of the data, or use a subset of the training data as required, with justification in the report.

To better understand the task and the preprocessing required, you should perform some exploration of the data. You may like to explore which cells each class corresponds to, the distribution of the data such as the number of examples in each class, and consider the characteristics of the images, such as whether they are centred, the size of different features in the images, pixel intensities across different images etc. You should also explore if there are factors which may make the task more difficult, such as classes with similar features. In your report, you will include anything you feel is relevant from this section.

Apply appropriate preprocessing techniques to the data, based on the insights from your data exploration and/or with reference to other sources. You will need to justify your preprocessing choices in the report. You may apply different preprocessing techniques for the different algorithms, with justification in the report. You may also like to use preprocessing techniques that reduce the runtime of your models, but please carefully consider which transformations may be appropriate for particular algorithms.

Consider if you need to make any additional splits of the data and think carefully about how each part of the data should be utilised to evaluate hyperparameter combinations and compare the performance of the different models.

Algorithm design and setup

You will need to design and implement four algorithms primarily using the sklearn and/or keras libraries, in order to investigate their strengths and weaknesses. You will explain your models and justify your choices in the report.

- A fully connected neural network (MLP)
- A convolutional neural network
- 2 other algorithms we have covered in the course (at least one of which should involve an ensemble method)

In this section implement an instance of each model before tuning hyperparameters, and set up any functions you may require to tune hyperparameters in the next section.

Note that it is not feasible to consider every possible neural network architecture when designing your models, but you will need to justify your design decisions in the report, including any sources where relevant. You may like to conduct some rough experimentation to converge on a reasonable design. Consider hardware constraints when designing your models.

Although you may like to reference external sources when designing your algorithms, **you must implement your neural network models yourself**, rather than import prebuilt models from Keras (such as those available in keras.applications).

Hyperparameter tuning

Perform a search over relevant hyperparameters for each algorithm using an appropriate search strategy of your choice (you will need to justify your chosen hyperparameters and search strategy in the report). For the neural network models, you should tune at least 3 hyperparameters. You may need to consider tradeoffs between your search and achieving a feasible runtime. You may use different search algorithms for the different models if appropriate.

Keep a record of results and runtimes with each hyperparameter combination (you may need to consult documentation to see how to extract this information) and use these to produce appropriate visualisations/tables of your trends in your hyperparameter search to aid the discussion in your report.

Please preserve the output of these cells in your submission and keep these hyperparameter search cells independent from the other cells of your notebook to avoid needing to rerun them, ie. ensure the later cells can be run if these cells are skipped.

Final models

Include cells which train the selected models with the best hyperparameters found during your search (remember to make these independent of your hyperparameter search cells). Use these implementations to compare the performance (and other relevant properties) of the different models using the test set.

Report

Introduction

State the aim of the study and outline its importance. You may like to consider the importance of this dataset, but also the importance of comparing algorithms and their suitability for a given task more generally.

Data

Describe the data, including all important characteristics, such as the number of samples, classes, dimensions, and original source of the images. Include anything you feel is relevant from your data exploration as outlined in the Code section above. You may wish to include some sample images to aid this discussion where appropriate.

Justify your chosen preprocessing techniques either through your insights from the data exploration or with reference to other sources. Explain how the preprocessing techniques work, their effect/purpose and any choices in their application. If you have considered but purposefully omitted possible preprocessing techniques, briefly justify these decisions.

Methods

In this section, you should explain the machine learning methods you have chosen. Please include any external references you have utilised.

- *Theory*: For each algorithm, explain the main theoretical ideas. Justify why you chose your 2 algorithms for this task.

- *Strengths and weaknesses*: Describe the relative strengths and weaknesses of the algorithms with reference to their theoretical properties. You may like to consider factors such as performance, overfitting, runtime, interpretability, and anything else you feel is relevant. Explain the reasons behind these properties (e.g. don't simply state that CNNs perform better on images, but explain why this is the case).

- *Architecture and hyperparameters*: State and explain the chosen architectures or other relevant design choices you made in your implementation. Describe the hyperparameters you will tune and outline your search method to be applied and why you chose these. Briefly explain what each hyperparameter controls and the expected effect on the algorithm.

Results and Discussion

Begin by presenting your hyperparameter tuning results. Include appropriate tables or graphs to illustrate the trends (metrics, runtime etc.) across different hyperparameter values. Discuss the trends and provide possible explanations for your observations. Consider if the results aligned with your predictions.

Next, present a comparison of the results for the four different models you have implemented (with their best hyperparameters). This should include a table with the best hyperparameter combination for each model, relevant performance metrics, and runtime(s). Analyse and discuss the results, referring to the theoretical properties and strengths/weaknesses of the models you discussed above. Consider if the results aligned with your expectations.

As well as performance and runtimes, include anything else that you feel is interesting and/or relevant. For example, you might like to comment on the types of mistakes particular models made etc.

Please do not include screenshots of raw code outputs when presenting your results. Instead tabulate/plot any results in a manner more appropriate for presentation in the report.

Conclusion

Summarise the main findings from your study. Mention any limitations to your study and suggest future work that could be attempted. Please make the future work suggestions specific (rather than eg. "try more algorithms") and justify why they would be appropriate, perhaps with reference to the limitations of your study you described.

Reflection

Write one to two paragraphs outlining your most important learning points from completing the assignment.

References

Include references to any sources you have utilised in completing the code and/or report. You may choose an appropriate referencing style, such as IEEE.

Academic Honesty

While the University is aware that the vast majority of students and staff act ethically and honestly, it is opposed to and will not tolerate academic integrity breaches and will treat all allegations seriously. Further information on academic integrity, and the resources available to all students can be found on the academic integrity pages on the current students website: https://sydney.edu.au/students/academic-integrity.html.

Marking Criteria

Code- 10 Marks (40% of the assignment marks)

Requirement	**Mark**			
Implements preprocessing techniques	Preprocessing does not function or has significant implementation issues **[0]**	Preprocessing codes runs but has moderate implementation issues **[0.5]**	Good; preprocessing techniques implemented with minor issues **[0.75]**	Excellent; preprocessing performed appropriately with no implementation issues **[1]**
Sets up and implements fully connected neural network architecture	No functioning algorithm/major issues with implementation **[0]**	Algorithm somewhat functions, but has serious issues with design or implementation **[0.25]**	Good; algorithm functions well. Minor issues with implementation. **[0.5]**	Excellent; algorithm is appropriate, and there are no issues with implementation **[0.75]**
Sets up and implements convolutional neural network architecture	No functioning algorithm/major issues with implementation **[0]**	Algorithm somewhat functions, but has serious issues with design or implementation **[0.25]**	Good; algorithm functions well. Minor issues with implementation. **[0.5]**	Excellent; algorithm is appropriate, and there are no issues with implementation **[0.75]**
Implements two other appropriate machine learning methods	No functioning algorithms/major issues with implementation **[0]**	Algorithms somewhat functions, but has serious issues with design or implementation **[0.5]**	Good; algorithms function well. Minor issues with implementation. **[0.75]**	Excellent; algorithms are appropriate, and there are no issues with implementation **[1]**
Fully connected neural network – hyperparameter search	No functioning hyperparameter search or completely irrelevant hyperparameters **[0]**	Major issues with search method, missing hyperparameters, or hyperparameter values **[0.5]**	Good; minor issues with search method, hyperparameters or values. Tunes over at least 3 hyperparameters appropriately. **[0.75]**	Excellent; well implemented hyperparameter search **[1]**
CNN – hyperparameter search	No functioning hyperparameter search or	Major issues with search method, missing	Good; minor issues with search method,	Excellent;well implemented

	completely irrelevant hyperparameters **[0]**	hyperparameters, or hyperparameter values **[0.5]**	hyperparameters or values. Tunes over at least 3 hyperparameters appropriately. **[0.75]**	hyperparameter search **[1]**
Algorithm of choice 1 – hyperparameter search	No functioning hyperparameter search or completely irrelevant hyperparameters **[0]**	Major issues with search method, missing hyperparameters, or hyperparameter values **[0.25]**	Good; minor issues with search method, hyperparameters or values. **[0.5]**	Excellent; well implemented hyperparameter search **[0.75]**
Algorithm of choice 2 – hyperparameter search	No functioning hyperparameter search or completely irrelevant hyperparameters **[0]**	Major issues with search method, missing hyperparameters, or hyperparameter values **[0.25]**	Good; minor issues with search method, hyperparameters or values. **[0.5]**	Excellent; well implemented hyperparameter search **[0.75]**
Best hyperparameter combination of each model trained and evaluated in separate cell	Not completed, or significant issues **[0]**		Completed with no/minimal issues **[0.5]**	
Code quality	Very poor code quality throughout. eg. some code does not run, no comments or markdown text, very poor variable names **[0]**	Poor code quality, eg. poor comments or not enough text to easily read the notebook, poor variable names **[0.75]**	Good code quality; minor issues with one aspect such as comments, not enough text, or variable names **[1.5]**	Excellent, readable code and overall notebook **[2.5]**

Report – 15 marks (60% of the assignment marks)

Requirement	**Mark**			
Introduction - 1 marks				
Aim	Not discussed or very poor **[0]**	Good; minor issues eg. not highlighting all aspects of study (comparison, hyperparameter tuning etc.) **[0.5]**	Excellent; aim of study is well discussed with no issues **[1]**	
Importance	Not discussed or very poor **[0]**	Good; importance partially discussed but missing some aspects, such as importance of comparing classifiers, or other issues **[0.5]**	Excellent; importance well justified and related to practical use **[1]**	
Data - 1.5 marks				
Data description and exploration	Dataset not described or very poor **[0]**	Limited dataset description with missing information and/or no data exploration **[0.25]**	Minor issues with dataset description and/or exploration **[0.5]**	Thorough data description and exploration, including discussion of important features and challenges as mentioned in the assignment specification, with sample images where relevant **[0.75]**
Preprocessing description and justification	Preprocessing not mentioned or very poor **[0]**	Preprocessing mentioned but not described well and/or missing/poor justification **[0.25]**	Good; Minor issues with either preprocessing description or justification of choices **[0.75]**	Excellent description of preprocessing techniques and their effect/purpose. Techniques used are justified from lectures, labs, or other sources.Brief discussion of which pre-processing techniques were considered but not necessary. **[0.75]**
Methods - 4.5 marks				
Fully connected neural network - description	Description missing or very poor **[0]**	Major issues with description **[0.1]**	Good description, with minor issues or missing detail **[0.25]**	Excellent description with sufficient detail to explain the advantages and disadvantages of

				the algorithms later. References included where appropriate. **[0.5]**
Convolutional neural network - description	Description missing or very poor **[0]**	Major issues with description **[0.1]**	Good description, with minor issues or missing detail **[0.25]**	Excellent description with sufficient detail to explain the advantages and disadvantages of the algorithms later. References included where appropriate. **[0.5]**
Algorithm of choice 1 - description	Description missing or very poor **[0]**	Major issues with description or justification of inclusion, including poor design decisions **[0.1]**	Good description and justification of inclusion, with minor issues or missing detail **[0.25]**	Excellent description and justification of inclusion with sufficient detail to explain the advantages and disadvantages of the algorithms later. References included where appropriate. **[0.5]**
Algorithm of choice 2 - description	Description missing or very poor **[0]**	Major issues with description or justification of inclusion, including poor design decisions **[0.1]**	Good description and justification of inclusion, with minor issues or missing detail **[0.25]**	Excellent description and justification of inclusion with sufficient detail to explain the advantages and disadvantages of the algorithms later. References included where appropriate. **[0.5]**
Comparison of strengths and weaknesses	Not included or very poor **[0]**	Major issues or omissions **[0.25]**	Good; minor issues including some relevant points of comparison missed **[0.75]**	Excellent comparison of the relative strengths and weaknesses of the classifiers from a theory perspective, and considering this particular dataset in the comparison. References included where appropriate. **[1]**
Architecture and hyperparameter tuning description	Not included or very poor **[0]**	Major issues or omissions in description and justification of design choices **[0.5]**	Good; architecture choices and search methods are well described and justified, and hyperparameters chosen to search over are explained.	Excellent description and explanation/justification of architecture design choices, search methods, and chosen

			Minor issues or lacking detail. **[1]**	hyperparameters. **[1.5]**
Results and discussion - 4.5 marks				
Hyperparameter tuning results presentation	No figures/tables or only screenshots of code output **[0]**	Figures or tables have major issues or omissions **[0.5]**	Good; figures or tables are appropriate and show trends/results from hyperparameter tuning. Minor issues with presentation. **[0.75]**	Excellent presentation of hyperparameter tuning results in appropriate figures or tables, with no presentation issues. If there are any relevant differences in runtime, these are presented. **[1]**
Hyperparameter tuning discussion	Not included or very poor **[0]**	Discussion has major issues or omissions **[0.5]**	Most important hyperparameter tuning results/trends are discussed. Includes comment on how the results aligned with predictions. Minor issues and/or lack of detail. **[1]**	Excellent discussion of hyperparameter results/trends, including possible explanations or reflections on how the results aligned with predictions. **[1.5]**
Results table	Not included or very poor, including screenshots of code output **[0]**	Major issues with formatting or omission of multiple results **[0.1]**	Minor issues with formatting or omission of one important result **[0.25]**	Excellent table with all required results and appropriate formatting **[0.5]**
Results discussion and analysis	Not included or very poor **[0]**	Discussion has major issues or omissions **[0.5]**	Most important trends in the results discussed, and compared to expectations based on theoretical properties. Minor omissions and/or lack of detail. **[1]**	Excellent analysis of the trends in the results, with comparison to expectation based on theoretical properties. Differences in runtime are discussed and justified. Possible exploration of further trends beyond the tabulated results (e.g. differences by class accuracy, precision vs recall etc.) **[1.5]**
Conclusion and future work - 1 mark				
Summary of main findings and identification of study limitations	Not included or very poor **[0]**	Major omissions or issues in summary and/or limitations **[0.1]**	Minor issues with summary (eg. does not consider runtime, or misses some relevant limitation(s)) **[0.25]**	Excellent summary which considers factors such as runtime and practicality of the algorithms for this particular task.

				Limitations identified are relevant and appropriate. **[0.5]**
Future work suggestions	Not included or very poor **[0]**	Suggestions not specific enough or do not address study limitations **[0.1]**	Minor issues with suggestions **[0.25]**	Suggestions are concrete and directly address the study limitations **[0.5]**
Reflection - 0.5 marks				
Reflection	Not included or very poor **[0]**	Reflection is lacking in depth or detail **[0.25]**	Excellent, relevant reflection with sufficient depth **[0.5]**	
Report presentation - 2 marks				
Formatting, presentation and structure	Serious issues with formatting or structure that make the report difficult to read **[0]**	Unclear structure or formatting issues, but report is still readable **[0.5]**	Minor issues with structure or formatting **[0.75]**	No issues with report structure or formatting. Sections are clearly delineated and formatting is clean and legible. Code snippets are not included inappropriately in the report. **[1]**
Academic writing	Serious spelling or grammatical issues in all aspects of report that make the report difficult to read **[0]**	Many minor spelling or grammatical issues that hinder the overall readability of the report, and/or non-academic language in many sections **[0.5]**	Several minor spelling or grammar mistakes that do not hinder the overall readability of the report, and/or non-academic language in some sections **[0.75]**	Very few minor spelling or grammar mistakes. Language is academic in style with clear sentences. **[1]**

Dataset Attribution

Yang, J., Shi, R., Wei, D. *et al.* MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Sci Data* 10, 41 (2023). https://doi.org/10.1038/s41597-022-01721-8

Andrea Acevedo, Anna Merino, et al., "A dataset of microscopic peripheral blood cell images for development of automatic recognition systems," Data in Brief, vol. 30, pp. 105474, 2020.