

*By submitting this portfolio the authors certify that this is their original work, and they have cited all the referenced materials properly.*

Deep Learning report

[name, student number, email address]

[name, student number, email address]

[name, student number, email address]

Minor: [Name of the minor]

Group: [Group nr]

Date: [kies een datum]

# INSTRUCTIONS FOR PROJECT submission

**When submitting your project results, please follow these steps:**

1. **Rename this file** 
   * **Replace {YOUR GROUP\_NUMBER} with your group number**
   * **Replace {YOUR\_NAME} with your name**
   * **Replace {YOUR\_STUDENT\_NUMBER} with your student number**
2. **Delete this instruction page**
3. **Prepare the project results:**
   * **Report as a Word file**
   * **code in a zip file**
   * **deployment video**
4. **Per group, email the project results to the instructor**
5. **Upload your project results to HAND-IN** 
   * **Each student must upload INDIVIDUALLY**
   * **Use your renamed file**
   * **PLEASE DO NOT UPLOAD YOUR DATA AND MODEL.**

Contents

[1 Introduction 3](#_Toc88129256)

[2 Problem statement 4](#_Toc88129257)

[3 Data augmentation and preprocessing 5](#_Toc88129258)

[4 CNN architecture and training 6](#_Toc88129259)

[5 Deploy and test 7](#_Toc88129260)

[6 Conclusion 8](#_Toc88129261)

[7 References 9](#_Toc88129262)

[Code appendices 10](#_Toc88129263)

# Introduction

|  |  |
| --- | --- |
| Assignment | Introduce your DL portfolio  Explain how it fits in the minor program  Describe the importance of DL in your areas of interest  Show how DL relates to your main project in the minor  List your learning objectives |
| Acceptance criteria | DL relation to the minor is discussed.  DL portfolio relation to main project in the minor is discussed. |
| Size | Max 1 A4 |

# Problem statement

|  |  |
| --- | --- |
| Assignment | 1. Update the objective stated in your Machine Learning (ML) portfolio 2. Also update the requirements table and prioritize these requirements. Think about how you would test or prove whether your final result has met a requirement. |
| Acceptance criteria | Problem definition is specific and measurable [1].  Functional and technical requirements are listed and prioritized. |
| Size | Max 1 A4 |

# Data augmentation and preprocessing

|  |  |
| --- | --- |
| Assignment | 1. Increase your data set Start with your ML image set Choose data augmentation methods Apply these methods to create new images Explain why you chose these methods 2. Prepare images for CNNs Choose preprocessing steps to: - Make patterns clearer for CNNs, or - Reduce CNN complexity Consider these factors: - Image size - Color depth - Image enhancement -Normalization   Build a preprocessing pipeline  Explain each step in your pipeline   1. Test your preprocessing Run some images through your pipeline Show before and after examples Explain how each step improves the images |
| Acceptance criteria | Data augmentation methods are used and explained  Preprocessing pipeline is implemented  Each preprocessing step is explained and justified |
| Size | Max 3 A4 |

# CNN architecture, training and validation

|  |  |
| --- | --- |
| Assignment | 1. Design CNN architecture and/or use transfer learning Design a custom CNN:  - Decide number of layers  - Choose number of neurons per layer  - Select pooling methods  - Pick activation functions Use transfer learning:  - Select a pretrained model - Decide which layers to freeze  - Design new top layers for your specific task Explain each choice in your design 2. Train and optimize CNN Choose appropriate performance measures Consider accuracy, precision, recall, F1-score and explain Train your CNN using the training set Use cross-validation to check performance Optimize hyperparameters, e.g.: try different learning rates, adjust batch sizes, experiment with optimizer types   Prevent overfitting (explain which method you used and why)   1. Test your model Use the test set to check model performance Create confusion matrix Compute performance measures Check for overfitting or underfitting Discuss trade-offs, e.g. precision vs. recall or bias vs. variance 2. Visualize CNN learning Show how an input image transforms through your network Visualize at least 3 different layers Explain what features each layer detects |
| Acceptance criteria | Architecture is designed and argued.  Data is split into stratified subsets and checked.  CNN is trained, cross-validated, and fine-tuned.  Performance is evaluated using appropriate methods.  Visualization of network's internal representations is provided |
| Size | Max 5 A4 |

# Deploy and test

|  |  |
| --- | --- |
| Assignment | 1. Deploy your CNN model  Set up preprocessing and prediction pipeline Choose where to run your model: - On your target machine (e.g., Raspberry Pi) - Or on your training machine 2. Make a test plan Review your SMART problem definition List requirements to measure, such as: - Model performance (e.g., accuracy, precision) - Inference speed (e.g., frame rate) - Technical factors (e.g., camera angles, distances, lighting conditions) Set target levels for each measure Explain how you will test each measure 3. Conduct tests Run tests based on your plan Record all test results Compare results to your targets Note any unexpected behaviors or limitations 4. Document your work Write down your test plan Record all test results Explain any differences between results and targets |
| Acceptance criteria | Preprocessing and prediction pipeline deployed.  Test plan present.  Documentation of test results. |
| Size | Max 5 A4 |

# Conclusion

|  |  |
| --- | --- |
| Assignment | 1. Summarize your project List the main steps you took Explain your key decisions 2. Evaluate your results Compare your results to your initial goals Discuss if you met your SMART objectives Explain any differences between goals and results 3. Reflect on generalization performance Discuss how well your model works on new, unseen data Compare performance on training, validation, and test sets Explain any differences in performance across these sets 4. Analyze your approach Identify what worked well Point out areas for improvement Suggest changes for future projects |
| Acceptance criteria | Results are compared to initial goals and SMART objectives  Generalization performance is analyzed |
| Size | Max 1 A4 |

# References

|  |  |
| --- | --- |
| Assignment | Give references to the sources that you have used. |

|  |  |
| --- | --- |
| [1] | „SMART criteria,” 14 05 2020. [Online]. Available: https://en.wikipedia.org/wiki/SMART\_criteria. |
| [2] | A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Sebastopol, Canada.: O’Reilly Media, 2019. |

# Code appendices

|  |  |
| --- | --- |
| Assignment | 1. Select key code snippets Choose important parts of your code Include snippets for: - Data preprocessing - Model creation - Training process - Evaluation methods 2. Explain your code Add comments to each snippet Explain what each part does Describe why you made specific coding choices 3. Show coding best practices  Use clear variable names Structure your code logically Follow Python style guidelines (PEP 8) |
| Acceptance criteria | Code snippets are provided for key parts of the project  Code quality is sufficient |