

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

Final Machine Learning report

[name, student number]

[name, student number]

[name, student number]

Minor: [Name of the minor]

Group: [Group nr]

Date: [kies een datum]

**Every student must deliver a copy of the report individually using HAND-IN.**

**Before uploading, please rename this template and replace {YOUR GROUP\_NUMBER} with your group number, {YOUR\_NAME}, with your name, and {YOUR\_STUDENT\_NUMBER} with your student number.**

**Then delete this page.**

Contents

[1 Introduction 3](#_Toc111635509)

[2 Problem statement 4](#_Toc111635510)

[3 Data acquisition and exploration 5](#_Toc111635511)

[4 Feedback from another team on your preliminary report 6](#_Toc111635512)

[5 Model selection, training and validation 7](#_Toc111635513)

[6 Deploy and test 8](#_Toc111635514)

[7 Conclusion 9](#_Toc111635515)

[8 References 10](#_Toc111635516)

[Code appendices 11](#_Toc111635517)

# Introduction

|  |  |
| --- | --- |
| Assignment | Introduce your Machine Learning (ML) portfolio and position it within the context of the minor program. What is the relevance of ML to practical domains you are interested in? What is the relation between ML and your main project in the minor? What are your learning objectives? |
| Acceptance criteria | ML relation to the minor is discussed.  ML portfolio relation to main project in the minor is discussed. |
| Size | Max 1 A4 |

# Problem statement

|  |  |
| --- | --- |
| Assignment | 1. Define your objective, e.g. hand gesture classification, recognizing at least 3 gestures, or obstacle classification, recognizing at least 3 different obstacles in front of a mobile robot. Alternatively, combine an ML objective within your main project in the minor, by replacing the classifier in step 5 of the conventional vision train. Note that the subjects for building an ML project should be covered, i.e. data acquisition, exploration, and preparation, ML model selection, training, and finetuning, and model deployment and testing. 2. List and prioritize requirements. Think about measurable parameters, e.g. what performance criterium will you use, what performance levels are you hoping to achieve, what framerate, etc? List and prioritize your functional requirements, think about technical requirements too, e.g. camera angles, distances, etc. |
| Acceptance criteria | Problem definition is specific and measurable (SMART criteria, 2020).  Functional and technical requirements are listed and prioritized. |
| Size | Max 1 A4 |

# Data acquisition and exploration

|  |  |
| --- | --- |
| Assignment | 1. Collect and label an image set using a controlled test set-up. 2. Propose and argue image features and design a preparation algorithm to compute the features. 3. Explore the resulting feature data, and discuss data quality. Is your data set representative, sufficient, balanced, unbiased, etc.? Are your features informative, discriminating, independent, explainable? Look for correlations or combinations. 4. Prepare your data to better expose the underlying data patterns to ML algorithms. Does your data need cleaning? Do outliers need to be detected and removed? Do your features need to be transformed or scaled? |
| Acceptance criteria | Data collected, features engineered and argued.  Feature data is visualized and explored, quality is checked.  Preprocessing pipeline discussed and implemented. |
| Size | Max 5 A4 |

# Feedback from another team on your preliminary report

|  |
| --- |
| Proposed improvement of the problem definition |
| ….. |

|  |
| --- |
| Proposed improvement of the list of requirements |
| ….. |

|  |
| --- |
| Proposed improvement of data collection |
| ….. |

|  |
| --- |
| Proposed improvement of feature engineering and visualization |
| ….. |

# Model selection, training and validation

|  |  |
| --- | --- |
| Assignment | 1. Explore different ML models and select most suitable candidate. 2. Split your data into training, validation, (and possibly test) sets. 3. Train your ML model and optimize hyperparameters. Reduce overfitting by constraining the model. 4. Evaluate model performance by validation and possibly re-train. How should performance be measured? Discuss trade-offs, e.g. precision/recall, and bias/variance. |
| Acceptance criteria | Models are considered and selected model is argued. Data is split into stratified subsets and checked.  Model is trained, cross-validated, and fine-tuned using search methods.  Model performance is evaluated using confusion matrix and/or other performance metrics. Overfitting/underfitting check, etc. |
| Size | Max 5 A4 |

# Deploy and test

|  |  |
| --- | --- |
| Assignment | 1. Deploy your ML model on your target machine (e.g. Raspberry Pi). Note that this step is optional; you can also choose to continue testing on your training machine. 2. Make a system test plan to check your SMART problem definition. Think about measurable parameters, e.g. what performance criterium will you use, what performance levels are you hoping to achieve, what framerate, etc? Think about technical requirements too, e.g. acquisition angles, distances. 3. Run the model and perform tests. |
| Acceptance criteria | Preprocessing and prediction pipeline deployed.  Test plan present.  Documentation of test results. |
| Size | Max 2 A4 |

# Conclusion

|  |  |
| --- | --- |
| Assignment | 1. Discuss results and draw conclusions. 2. Reflect on your work using the ML project checklist (Géron, 2019). |
| Acceptance criteria | Results are concluded.  Generalization performance discussed. |
| Size | Max 1 A4 |

# References

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* Sebastopol, Canada.: O’Reilly Media.

*SMART criteria*. (2020, 05 14). Opgehaald van wikipedia: https://en.wikipedia.org/wiki/SMART\_criteria

# Code appendices