

*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]

**When submitting your report, 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. **Upload your report to HAND-IN** 
   * **Each student must upload their own copy**
   * **Use your renamed file**

Contents

[1 Introduction 3](#_Toc116372063)

[2 Problem statement 4](#_Toc116372064)

[3 Data acquisition and exploration 5](#_Toc116372065)

[4 Feedback FROM another team on your preliminary report 6](#_Toc116372066)

[5 Feedback TO another team on THEIR preliminary report 7](#_Toc116372067)

[6 Model selection, training and validation 8](#_Toc116372068)

[7 Deploy and test 9](#_Toc116372069)

[8 Conclusion 10](#_Toc116372070)

[9 References 11](#_Toc116372071)

[Code appendices 12](#_Toc116372072)

# Introduction

|  |  |
| --- | --- |
| Assignment | Introduce your ML portfolio  Explain how it fits in the minor program  Describe the importance of ML in your areas of interest  Show how ML relates to your main project in the minor  List 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. Choose one of these options:    1. Hand gesture classification (Recognize at least 3 gestures)    2. Object classification (Recognize at least 3 objects)    3. ML objective in your EVML project (Replace the classifier in step 5 of the conventional vision train)   Your project must cover these subjects: Data acquisition, Data exploration, Data preparation, ML model selection, Model training, Model fine-tuning, Model deployment, Model testing   1. List and prioritize requirements:    1. Functional requirements: List what your system must do Include measurable criteria, such as:  - gestures or objects to recognize - performance levels to achieve - frame rate needed    2. Non-functional requirements:  List technical constraints, such as:  - camera angles - operating distances - hardware limitations - environmental conditions |
| Acceptance criteria | Problem definition is specific and measurable [1].  Functional and technical requirements are listed and prioritized. |
| Size | Max 1 A4 |

# Data acquisition and exploration

|  |  |
| --- | --- |
| Assignment | 1. Make an image set Set up a controlled test area Take photos Label each photo Make visual representations of your data Assess data quality: - Does the data represent all cases? - Is there enough data? - Are all classes equally represented? - Is the data free from bias? 2. Compute, visualize, and check feature data Choose at least 3 image features Make an algorithm to calculate features   Assess feature data quality: - Are features useful (informative)? - Are features able to tell classes apart (discriminating)? - Are features easy to understand (explainable)?  - Are there outliers (unusual data points) to remove? Analyze feature relationships:  - Find correlations between features - Check if features provide unique information (independent) - Consider creating combined or transformed features if useful Explain your choices   1. Design and implement a preprocessing pipeline: calculate features create algorithm to clean data scale data   Explain each step in your pipeline Show how the pipeline makes data patterns clearer |
| 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 |
| ….. |

# Feedback TO another team on THEIR 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 |
| ….. |

|  |  |
| --- | --- |
| Acceptance criteria | Feedback quality |

# Model selection, training and validation

|  |  |
| --- | --- |
| Assignment | 1. Choose an ML model Compare different ML models Select the best model for your task Explain why you chose this model 2. Prepare your data Split your data into these sets: - Training set - Validation set - Test set   Make sure each set represents all classes equally Check that each set has enough data   1. Train and improve your model Choose suitable performance measures, such as accuracy, precision, recall, F1-score and explain Train your model with the training set Use cross-validation to check model performance Find the best hyperparameters (use a search method, such as grid search, random search) Prevent overfitting (explain which constraints you used and why) Retrain your model if needed 2. 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, such as precision vs. recall or bias vs. 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 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. |

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

|  |  |
| --- | --- |
| 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 |