Geometrical Deep Learning on 3D Models: Classification for Additive Manufacturing

Project Road Map

Bouziane, Nouhayla
nouhayla.bouziane@tum.de

Ebid, Ahmed
ge75roz@mytum.de

Srinivas, Aditya Sai
aditya.srinivas@tum.de

Bok, Felix
felix.bok@tum.de

Kiechle, Johannes
johannes.kiechle@tum.de

April 20, 2021

Contents

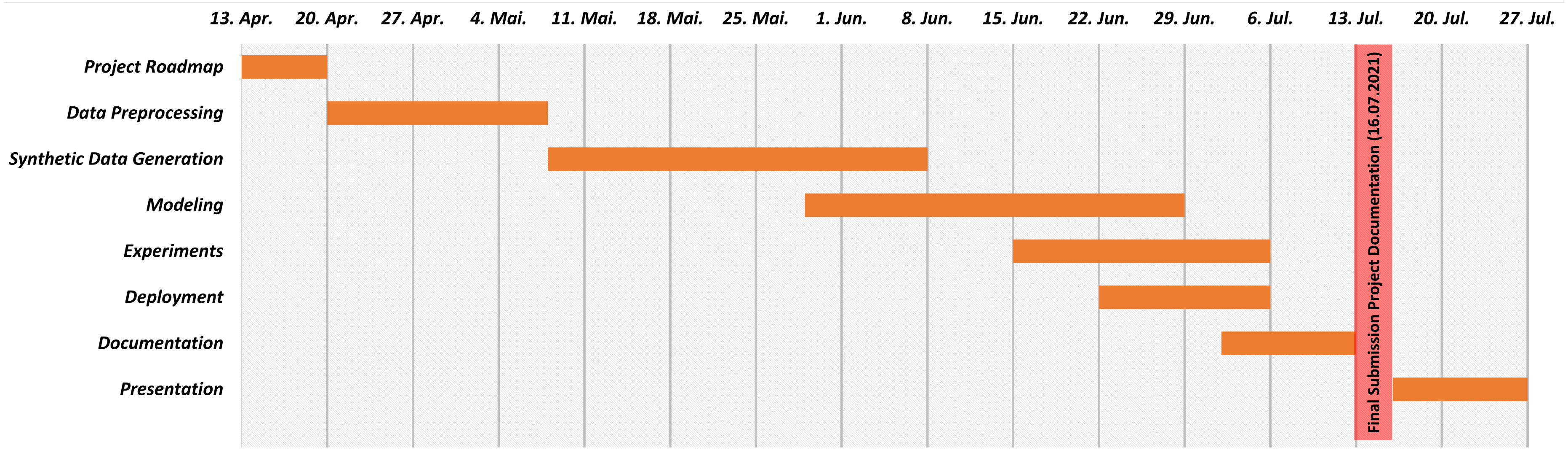
1 Overview & Timetable **2**

2 Project Roles **3**

3 Milestones **3**

3.1 Milestone 1 - Data Preprocessing 4
3.2 Milestone 2 - Synthetic Data Generation 5
3.3 Milestone 3 - Modeling 6
3.4 Milestone 4 - Experiments 7
3.5 Milestone 5 - Deployment 8
3.6 Milestone 6 - Documentation 9

1 Overview & Timetable



2 Project Roles

General responsibility	Person
Project Management	Bok, Felix
DevOps	Kiechle, Johannes
Data	Bouziane, Nouhayla | Ebid, Ahmed
Deep Learning	Srinivas, Aditya Sai
Documentation & Presentation	Bouziane, Nouhayla | Ebid, Ahmed

3 Milestones

For measuring the cape estimate the following scale was used:

Scale	1	2	3	4	5	8	13	20	40	100
Complexity	low									high
Level of detail	high									low
Dependence	no									many
Skills	multiple included in the team									expert needed

3.1 Milestone 1 - Data Preprocessing

- **Work Packages + Cape Estimates**
  - Download dataset - 1
  - Cleaning data - 5
  - Normalization of the data - 8
  - Align all models - 13
  - Find the right voxelization algorithm - 40
  - Conversion to voxel representation - 40
  - Evaluate resulting representations - 20
- **Deliverables**
  - Possible to download all data from ABC dataset
  - Tested voxelization algorithm
  - Voxelized ABC dataset all normalized (same size, same resolution)
  - 3D plot function for models in voxel representation
- **Risk Analysis**

Risk	**Mitigation**
Understanding of the given format	Read more documentation,
Difficulty to align models	Group data with the same alignments into subsets
Integrity of the data	Filter dataset
Objects not appearing in the same orientation	Re-orientation of data models
Size of the data needed	Set up enough storage on LRZ Data Science Storage, Use only a few models locally
Voxel model has defects	Use another algorithm for voxelization, extensive (manual) testing

3.2 Milestone 2 - Synthetic Data Generation

- **Work Packages + Cape Estimates**
  - Create basic defects - 5
  - Control the position and orientation of the defect - 8
  - Control the size of the defects - 40
  - Identify different defect shapes - 40
  - Generate/ store labels for models - 100
  - Scale code to work on large datasets - 2
- **Deliverables**
  - For each model in the ABC dataset a printable and non-printable version can be generated
  - Different kind of defects can be added
  - Labeled dataset
- **Risk Analysis**

Risk	**Mitigation**
Inability to determine whether an object is printable or not	Hypothesise a rule, exclude critical models
Models are too diverse	Divide dataset into subsets of similar objects
Defects created do not look as expected	Find alternative approach to introducing defects or apply re-meshing
Inconsistent hole sizes with different objects	Split up faces into smaller sub-faces
Defect created is not uniform in shape	Re-meshing

3.3 Milestone 3 - Modeling

- **Work Packages + Cape Estimates**
  - Setup torch pipeline framework - 100
    * Create on-the-fly dataloader
    * Getting data and models aligned
    * Setup MLflow
    * Implement performance metrics
    * Integrate Weights and Biases (WB) for real-time monitoring of training
  - Collection and implementation of different model architectures - 20
- **Deliverables**
  - Runnable training pipeline
  - Different models architectures to train
- **Risk Analysis**

Risk	**Mitigation**
Do not find suitable model	Good research
Mismatch between behaviour of loss curves and performance metrics	Use different performance metrics, manual
Undetected bug in the model implementation	Test implementation, code reviews, start small in the model and the training data and slowly increase complexity
Overutilization of compute resoures	Implementation of on-the-fly dataloader

3.4 Milestone 4 - Experiments

- **Work Packages + Cape Estimates**
  - Perform runs of different - 5
  - Evaluate performance - 40
    - Re-evaluate procedure of standardize/normalize data if performance is not satisfying
    - Investigate 3D shape candidates for which prediction fails and try to find reasoning
- **Deliverables**
  - One model with a decent performance
  - Final model package
- **Risk Analysis**

Risk	**Mitigation**
All deployed models fail to achieve descent performance and we do not know why	Consult additional literature, get feedback of supervisors, test code, manual checks
The performance of the best model is worse than expected/desired	Run comprehensive hyper-parameter optimization

3.5 Milestone 5 - Deployment

- **Work Packages + Cape Estimates**
  - Write complete inference pipeline - 5
  - Get Field data - 5
  - Test with Field data - 20
  - Benchmark inference pipeline - 5
- **Deliverables**
  - Field tested inference pipeline
- **Risk Analysis**

Risk	Mitigation
Good model is too large	Know inference hardware limitations and use it when building a model
Inference takes too much time to run	Know inference hardware limitations, optimize performance
Inference pipeline does not work with field data	Get field data early on and use it as an orientation
Performance on field data not as good as on synthetic dataset	Get field data early on and use it as an orientation, add performance metric using field data in the testing phase

3.6 Milestone 6 - Documentation

- **Work Packages + Cape Estimates**
  - Write documentation - 40
  - Prepare final presentation - 13
- **Deliverables**
  - Final documentation
  - Final presentation
- **Risk Analysis**

Risk	Mitigation
Too little time left	Start early on writing, good planning