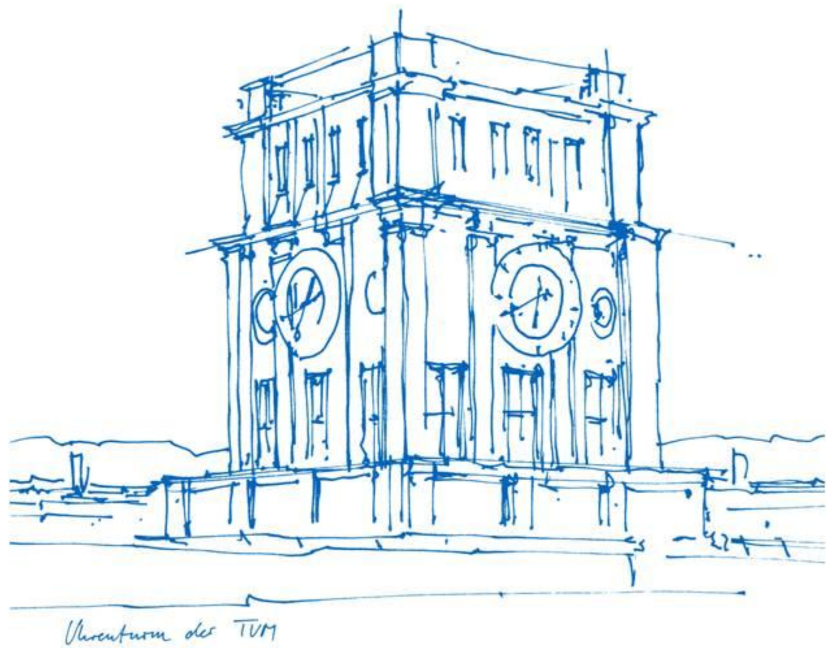
Geometrical Deep Learning on 3D Models: Classification for Additive Manufacturing

Nouhayla Bouziane | Ahmed Ebid | Aditya Sai Srinivas | Felix Bok | Johannes Kiechle

Technical University Munich

Faculty of Mathematics

06. July 2021



Agenda

1 **Overview**

2 **Progress**

- 2.1 **Motivation - Felix**
- 2.2 **Horovod Distributed Training - Ahmed**
- 2.3 **Deep Learning Modelling: Rotated Data - Nouhayla**
- 2.4 **Deep Learning Modelling and Experimentation - Adi**
- 2.5 **Performance Analysis - Johannes, Felix**

3 **Wrap up: Next Steps**

Agenda

1 Overview

2 Progress

2.1 Motivation - Felix

2.2 Horovod Distributed Training - Ahmed

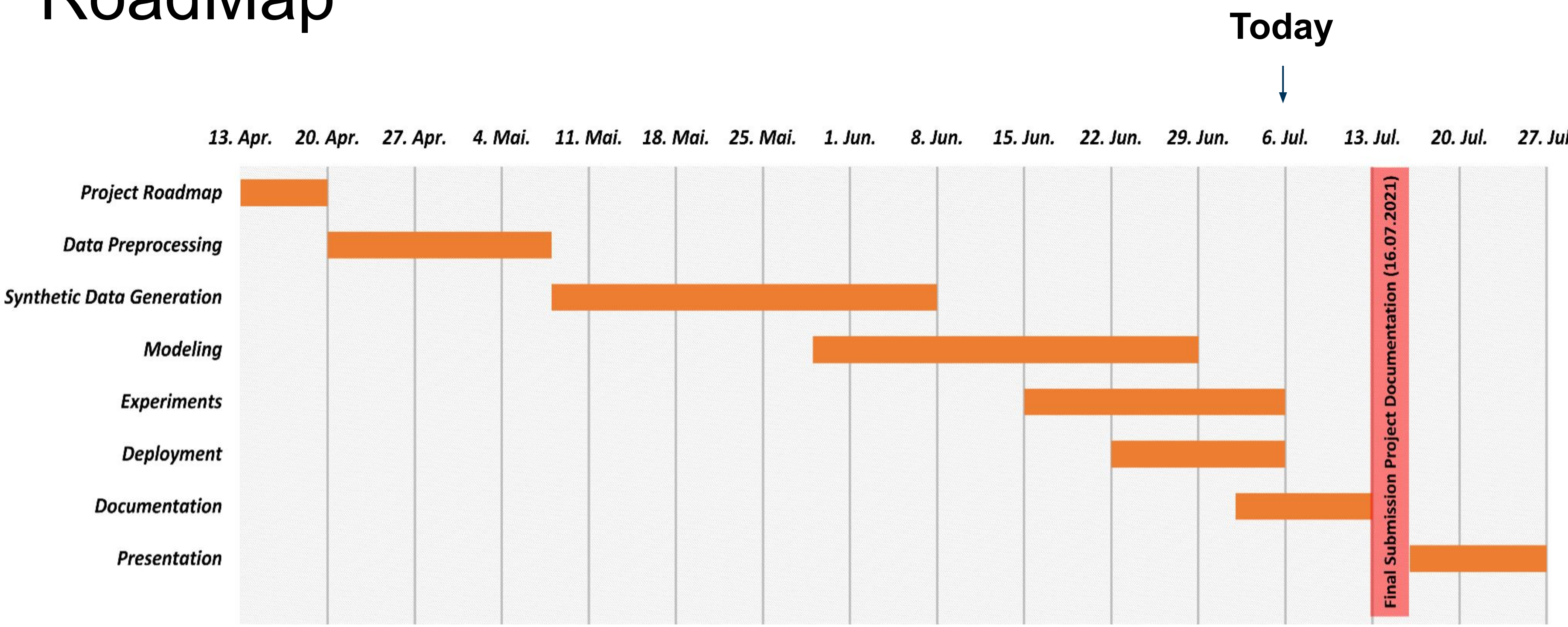
2.3 Deep Learning Modelling: Rotated Data - Nouhayla

2.4 Deep Learning Modelling and Experimentation - Adi

2.5 Performance Analysis - Johannes, Felix

3 Wrap up: Next Steps

RoadMap



Today
13. Apr.
20. Apr.
27. Apr.
4. Mai.
11. Mai.
18. Mai.
25. Mai.
1. Jun.
8. Jun.
15. Jun.
22. Jun.
29. Jun.
6. Jul.
13. Jul.
20. Jul.
27. Jul.
Project Roadmap
Data Preprocessing
Synthetic Data Generation
Modeling
Experiments
Deployment
Documentation
Presentation
Final Submission Project Documentation (16.07.2021)

Agenda

1 **Overview**

2 **Progress**

- 2.1 **Motivation - Felix**
- 2.2 **Horovod Distributed Training - Ahmed**
- 2.3 **Deep Learning Modelling: Rotated Data - Nouhayla**
- 2.4 **Deep Learning Modelling and Experimentation - Adi**
- 2.5 **Performance Analysis - Johannes, Felix**

3 **Wrap up: Next Steps**

Motivation

- Adaptive Manufacturing (AM) describes the **process of adding material layer by layer**, where each layer is a slice of a digital 3D model [1].

- AM offers a variety of high-impact benefits compared to classical manufacturing processes: **More complex objects can be produced faster, more sustainable and on demand** [2].

- The AM market has an a**verage growth rate of 27% over the last decade** and is **currently estimated at $12.8 billion** [4]. Some industry experts even estimate that the **market will reach $100-250 billion by 2025** [5].

- Companies are **investing heavily in AM technology** and its application to high volume manufacturing [6, 7, 8].

- **AM still has many limitations**, e.g. lack of design knowledge, imperfections during the printing phase, high cost in mass production....
  → One reason: **It requires a lot of human expertise and supervision** [3, 5]**.**

Agenda

1 **Overview**

2 **Progress**

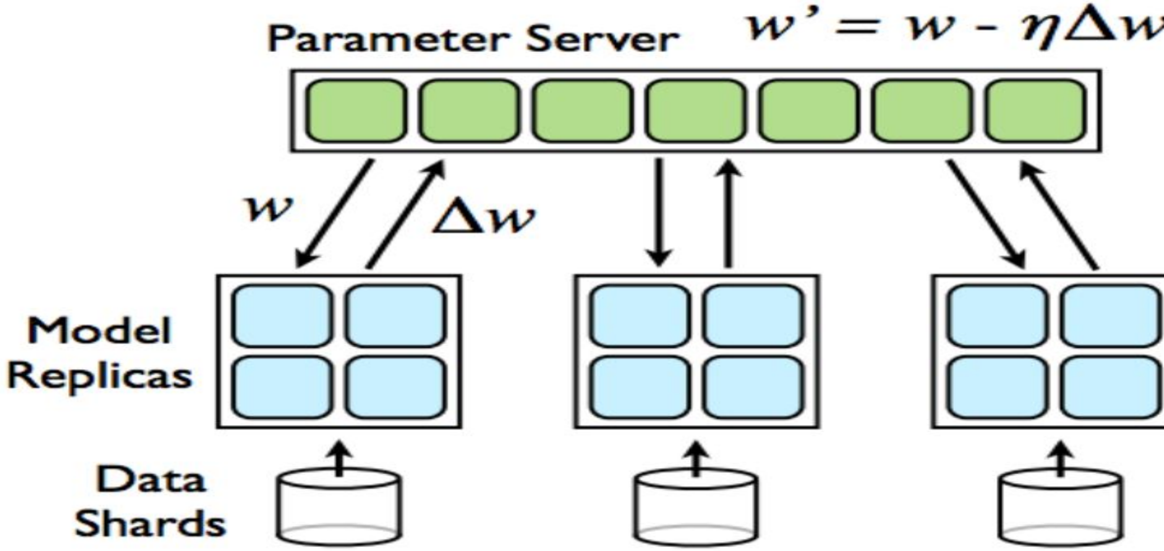
- 2.1 **Motivation - Felix**
- 2.2 **Horovod Distributed Training - Ahmed**
- 2.3 **Deep Learning Modelling: Rotated Data - Nouhayla**
- 2.4 **Deep Learning Modelling and Experimentation - Adi**
- 2.5 **Performance Analysis - Johannes, Felix**

3 **Wrap up: Next Steps**

Horovod Distributed Training

Horovod data-parallel distributed training paradigm works as follows:

1. Run multiple copies of the training script, and for each copy:
   a. read a chunk of the data
   b. run it through the model
   c. compute model updates (gradients)
2. Average gradients among the copies
3. Update the model
4. Repeat (from Step 1)



https://towardsdatascience.com/distributed-deep-learning-with-horovod-2d1eea004cb2

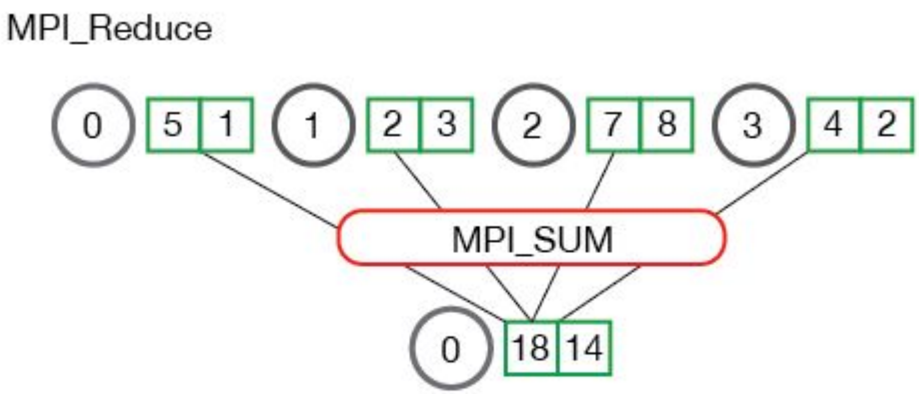
Horovod Distributed Training

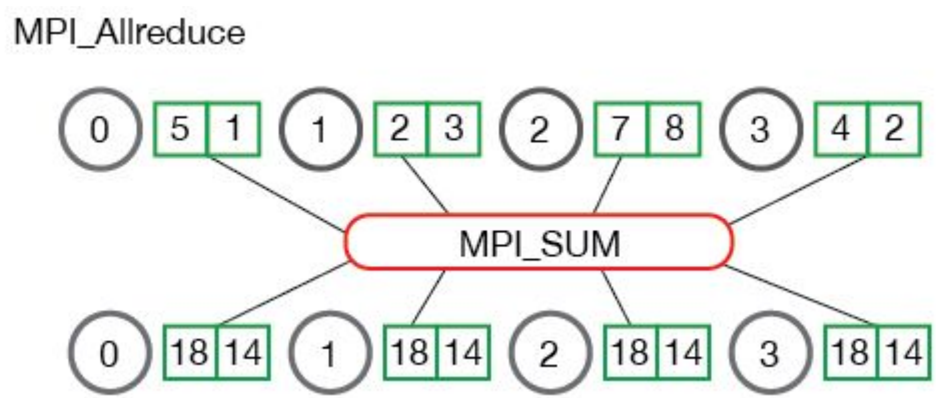
- Horovod core principles are based on MPI concepts:
  - size, rank, local rank, allreduce.

- MPI stands for Message Passing Interface.
- Example: training script launched on 4 servers, each having 4 GPUs. One copy is launched per GPU:
- **Size:** number of processes, 16.
- **Rank:** unique process ID, 0-15.
- **Local rank:** unique process ID within the server, 0-3.
- **Allreduce** is an operation that aggregates data among multiple processes and distributes results back to them.

https://github.com/horovod/horovod/blob/master/docs/concepts.rst

Horovod Distributed Training





https://mpitutorial.com/tutorials/mpi-reduce-and-allreduce/

Horovod Distributed Training: Implementation

1. Each process broadcasts metrics

```
train_loss_red = self.metric_average(self.train_loss, 'avg_loss')
train_acc_red = self.metric_average(self.train_acc, 'avg_acc')
```

2. Allreduce is applied to average the metrics and send it back to each process

```
def metric_average(self, val, name):
    tensor = val.detach().clone()
    avg_tensor = hvd.allreduce(tensor, name=name)
    return avg_tensor.item()
```

3. Only one process logs the averaged metrics

```
if hvd.rank() == 0:
    mlflow.log_metric("train_loss_step", train_loss_red)
    mlflow.log_metric("train_acc_step", train_acc_red)
```

https://github.com/horovod/horovod/blob/master/examples/pytorch/pytorch_lightning_mnist.py

VOLKSWAGEN DATA:LAB

Agenda

1 **Overview**

2 **Progress**

2.1 **Motivation - Felix**

2.2 **Horovod Distributed Training - Ahmed**

2.3 **Deep Learning Modelling: Rotated Data - Nouhayla**

2.4 **Deep Learning Modelling and Experimentation - Adi**

2.5 **Performance Analysis - Johannes, Felix**

3 **Wrap up: Next Steps**

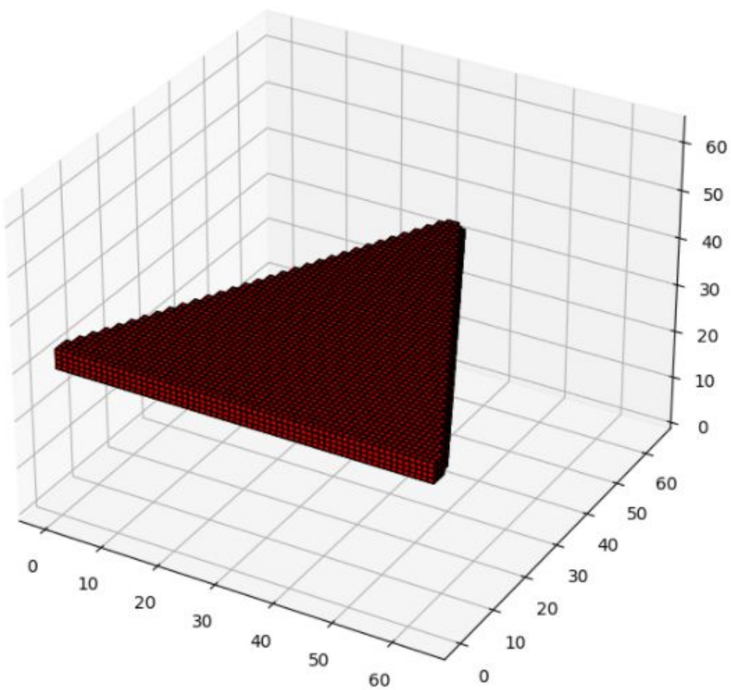
Generating the Data

Rotated holes with random angles: ϕ_x, ϕ_y, ϕ_z

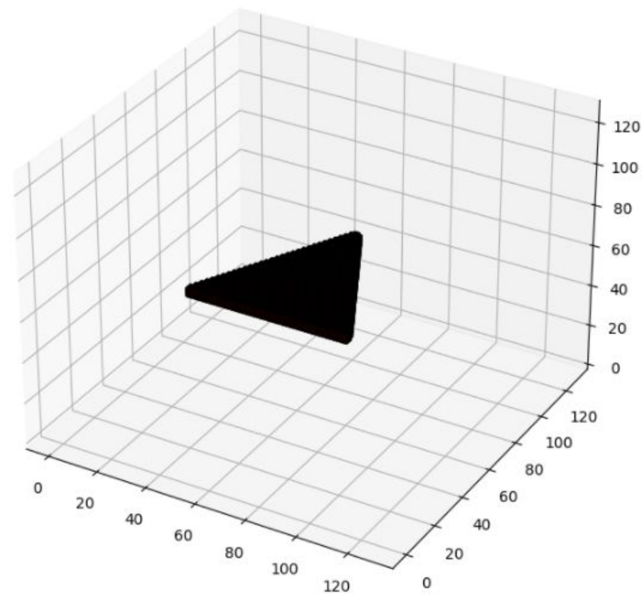
We set the following parameters:

- $r_p = 6$: radius printable
- $r_{np} = 3$: radius non-printable
- $b_p = 2$: border printable
- $b_{np} = 5$: border non-printable.

Generating the Data

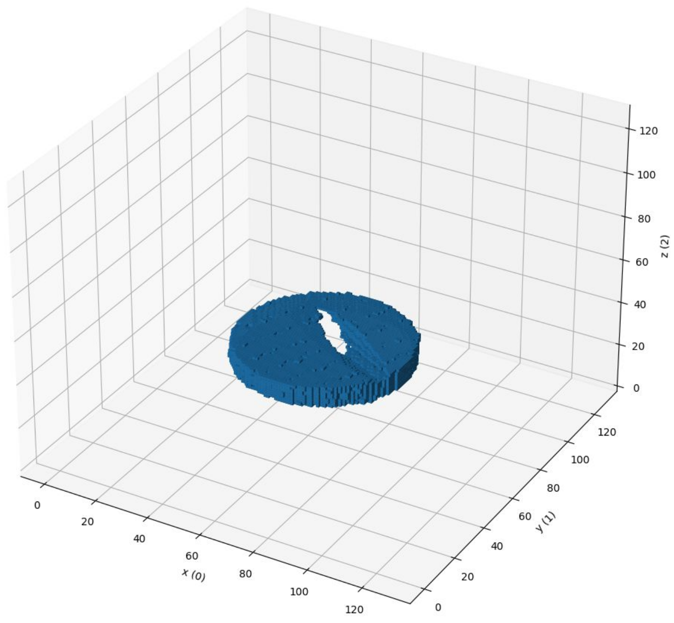


Model with resolution 64

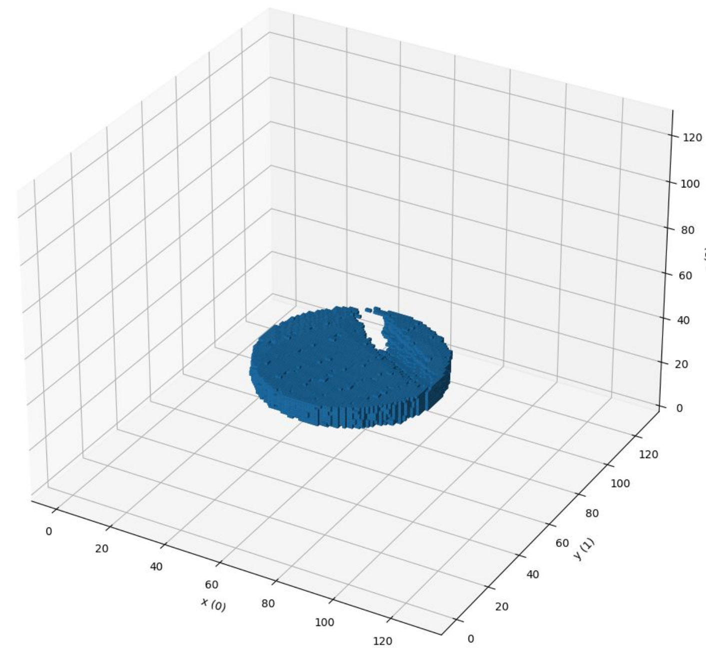


Padded Model with resolution 128

Generating the Data

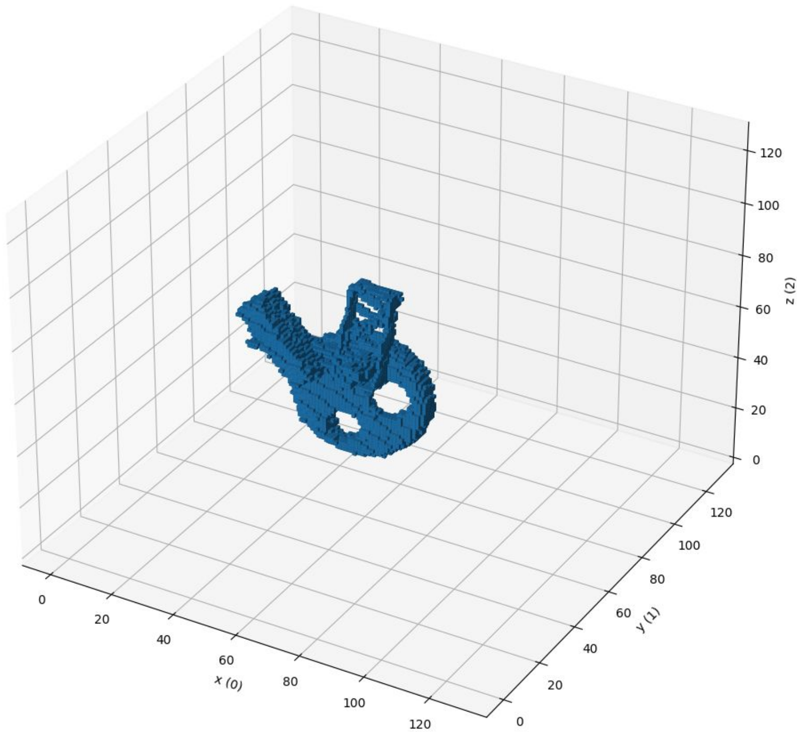


Model printable hole in the middle r_p = 6

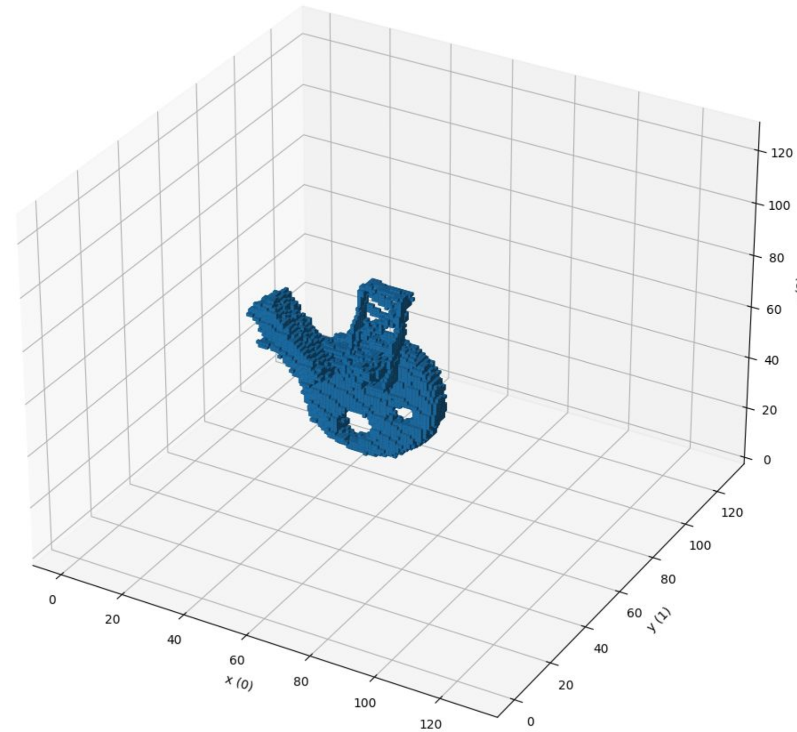


Model non printable hole in the border b_np = 2

Generating the Data



Model printable hole in the middle r_p = 6



Model non printable hole in the middle r_np = 3

Training using Data Rotation Data

Architecture: InceptionNet v1

learning_rate = 0.0003

momentum = 0.5

optimizer = Adam

loss_function = BCE

nb of epochs = 60

Dataset number of examples = 7000

the split Train/Val = 90/10

Results:

Validation accuracy: 0.85

Validation loss: 0.57

Hyperparameter Tuning using Optuna

Optuna is an automatic hyperparameter optimization software framework

- **Study:** optimization based on an objective function
- **Trial:** a single execution of the objective function

The hyperparameter suggestions:

- **lr:** [1e-4, 1e-2]
- **optimizer_name**: ["Adam", "Adadelta"]

Agenda

1 **Overview**

2 **Progress**

2.1 **Motivation - Felix**

2.2 **Horovod Distributed Training - Ahmed**

2.3 **Deep Learning Modelling: Rotated Data - Nouhayla**

2.4 **Deep Learning Modelling and Experimentation - Adi**

2.5 **Performance Analysis - Johannes, Felix**

3 **Wrap up: Next Steps**

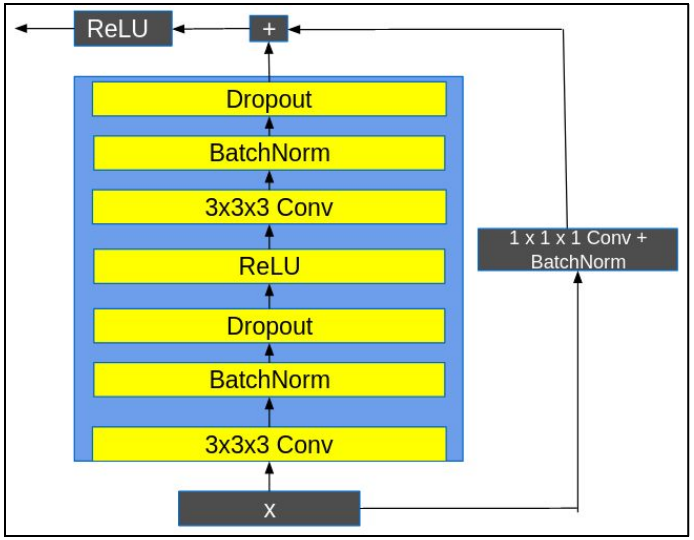
Neural Network Architectures

- ResNet_small
- InceptionNet_v1
- InceptionNet_v3

Neural Network Architectures

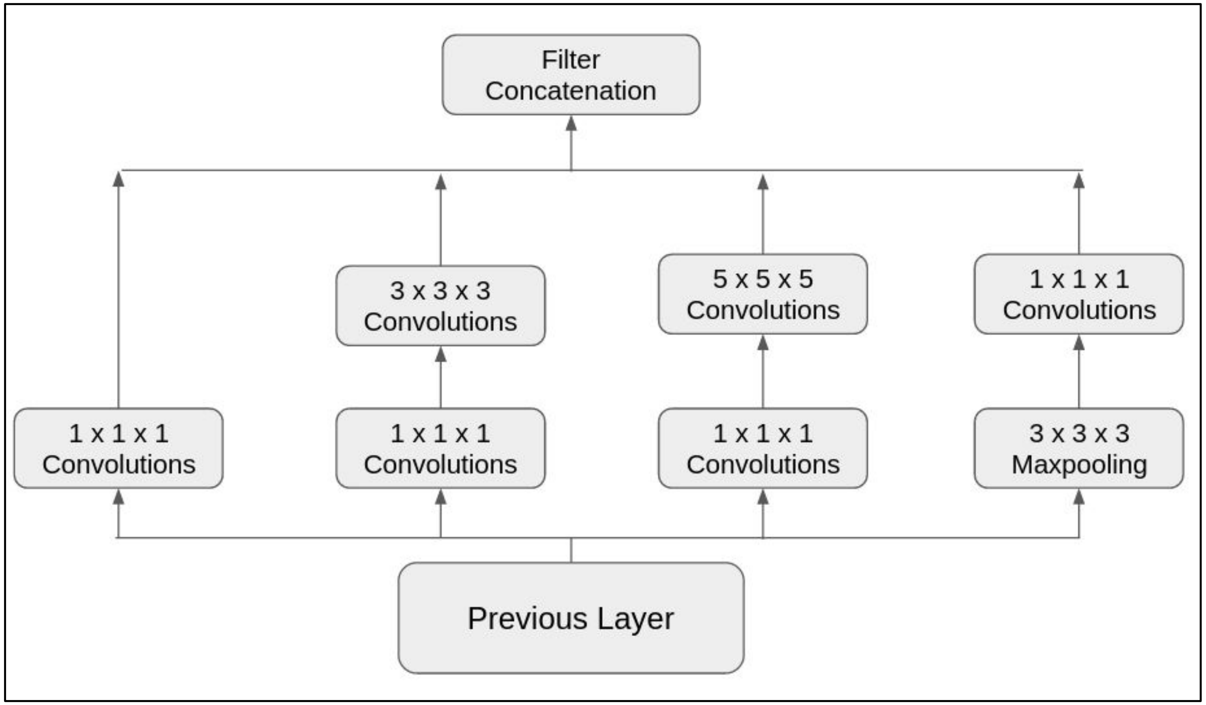
ResNet_small:

- Number of trainable parameters: 8.3M.
- Kernel sizes : 7 x 7 x 7, 3 x 3 x 3 kernel.
- Basic ResNet block similar to the standard ResNet block.
- Optimizers tried out : Adam and SGD



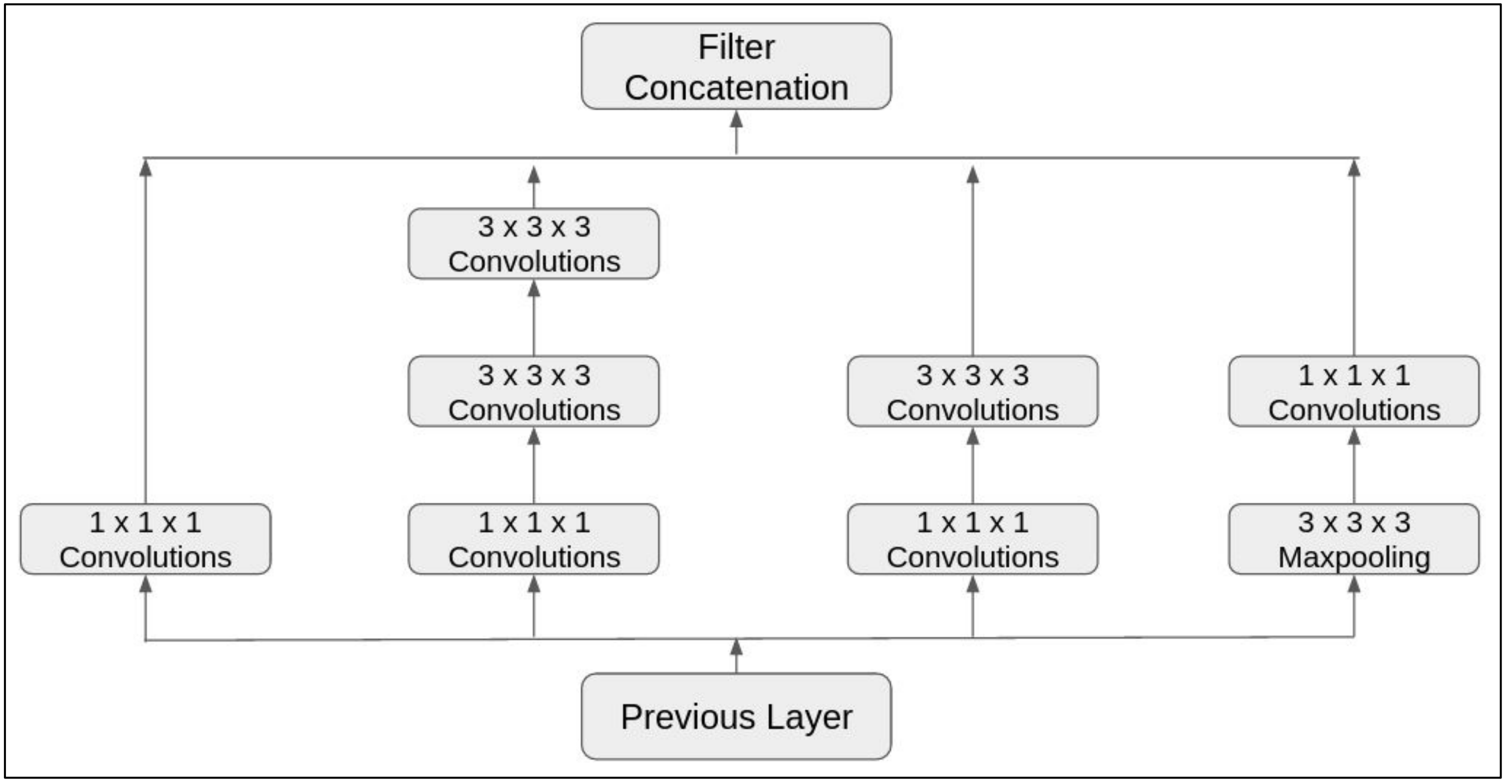
ResNet module

Neural Network Architectures



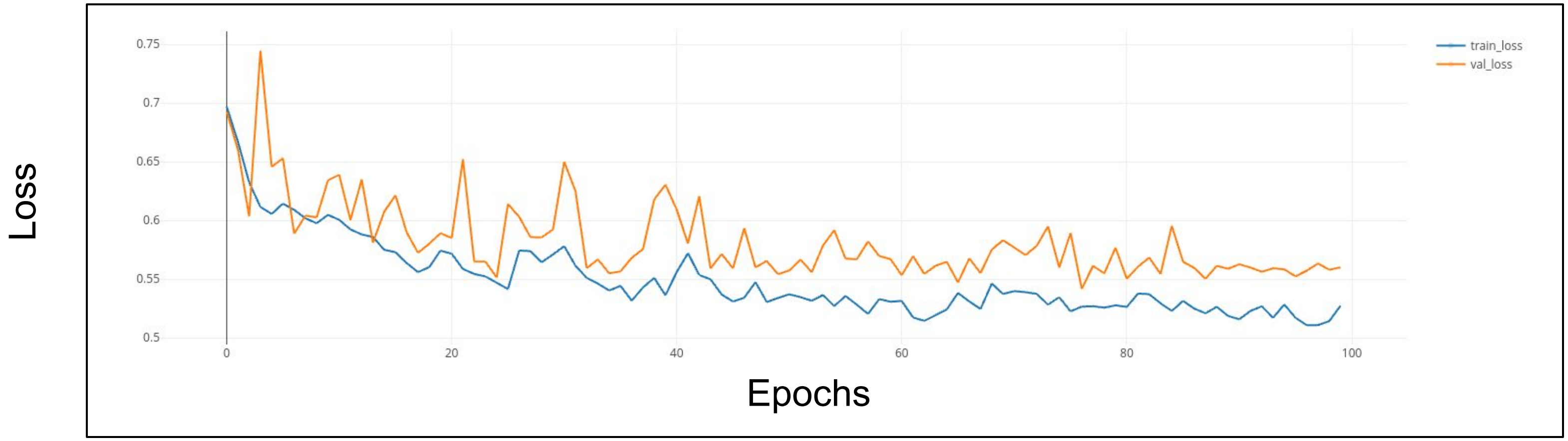
InceptionNet_v1 block

Neural Network Architectures

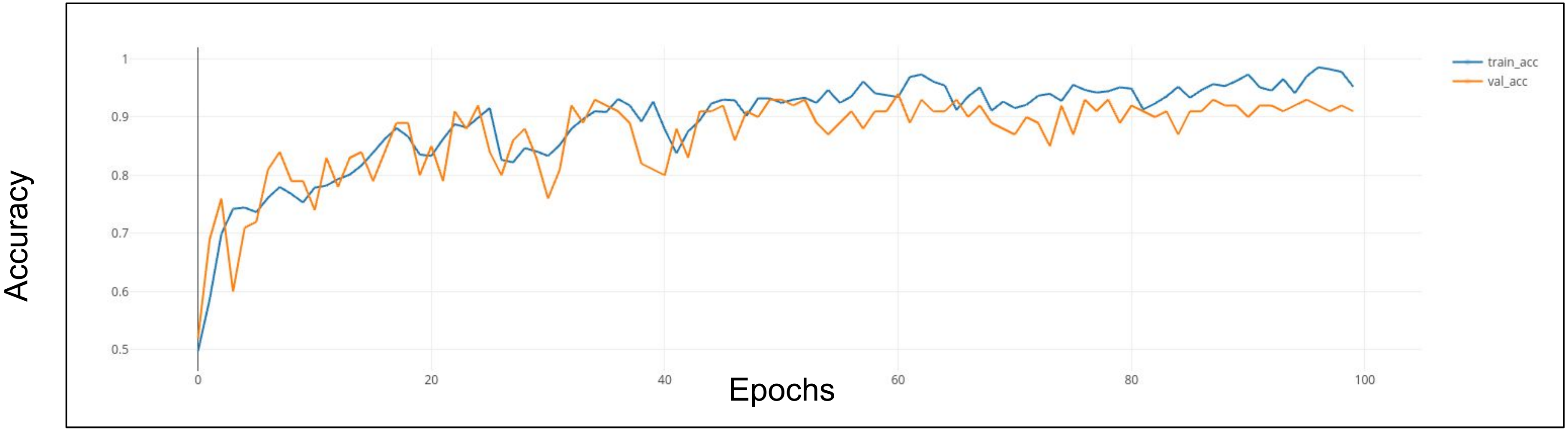


InceptionNet_v3 block

Experiments



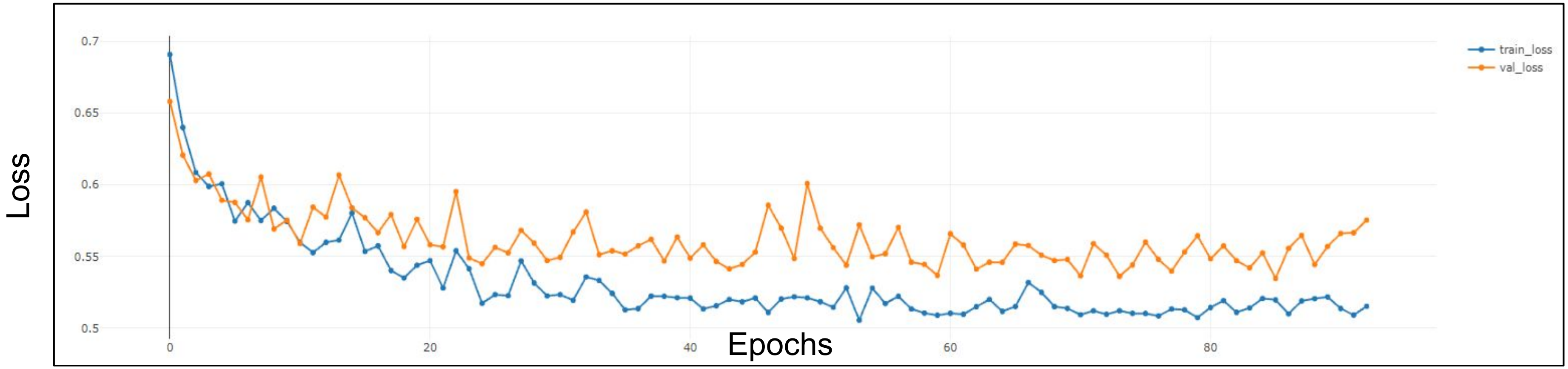
InceptionNet_v1 training and validation loss curves



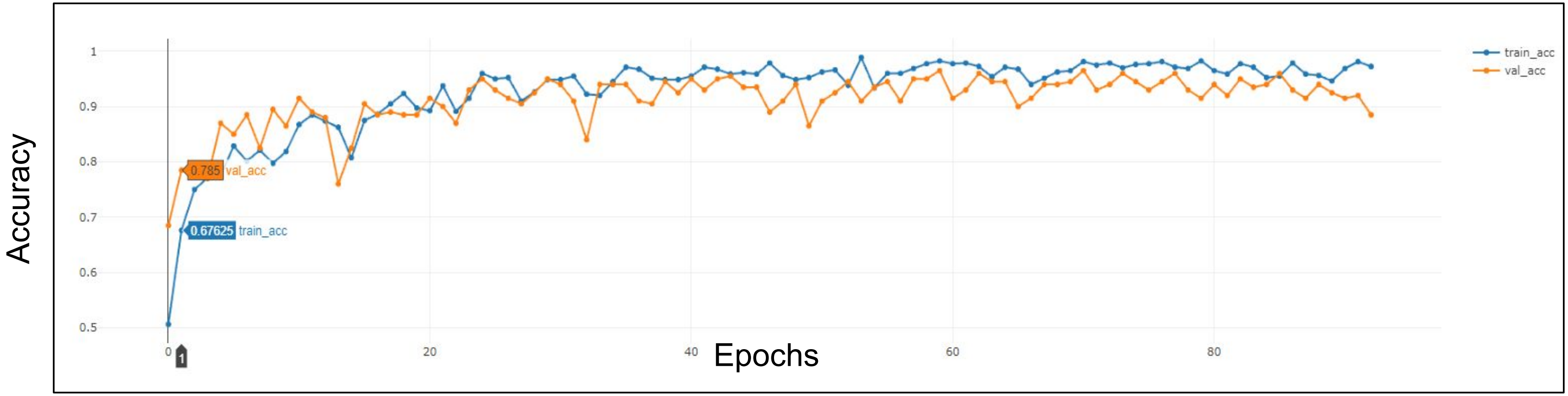
InceptionNet_v1 training and validation accuracy curves

VOLKSWAGEN DATA:LAB

Experiments



InceptionNet_v3 training and accuracy loss curves



InceptionNet_v3 training and validation accuracy curves

Agenda

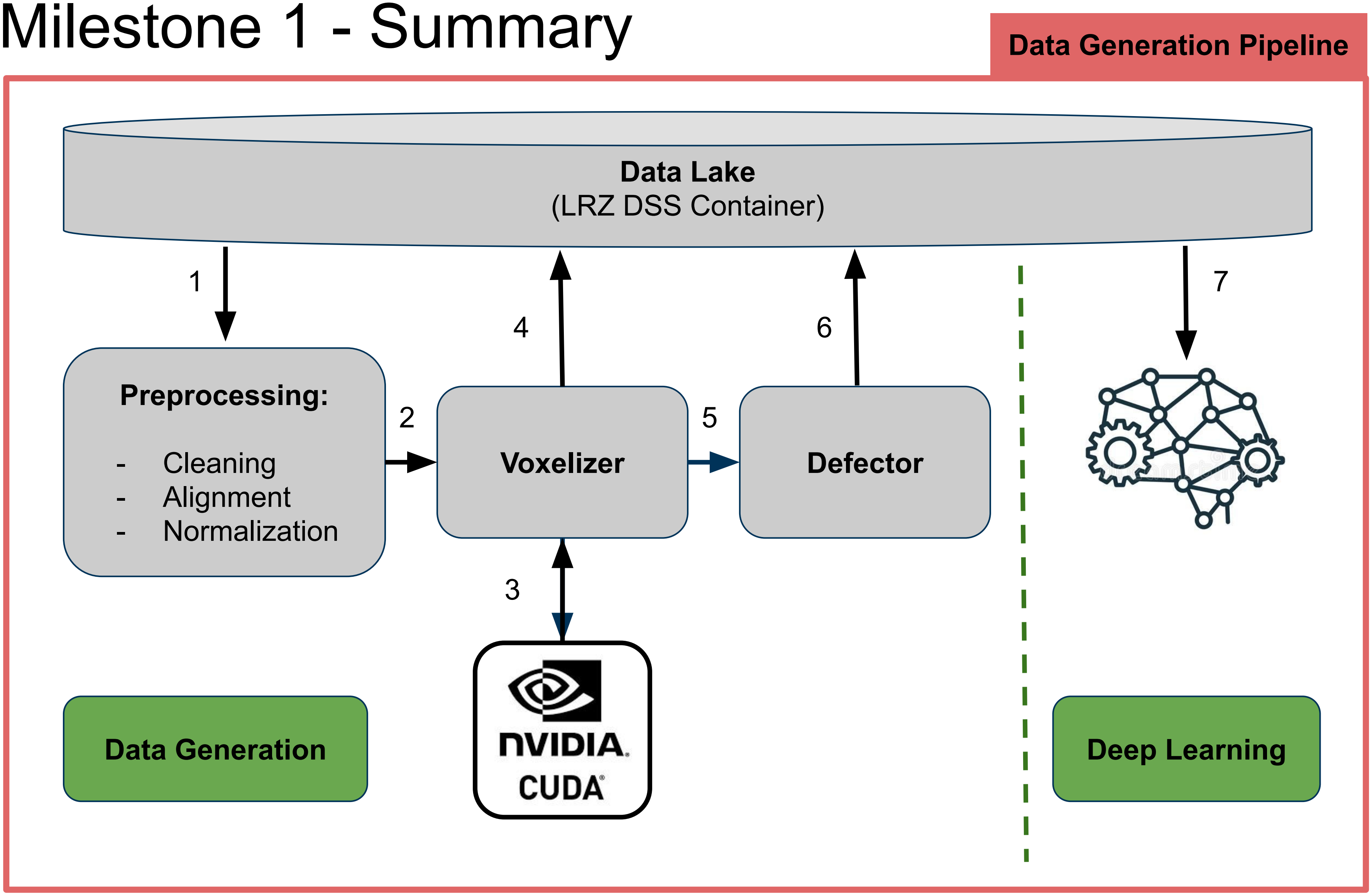
1 **Overview**

2 **Progress**

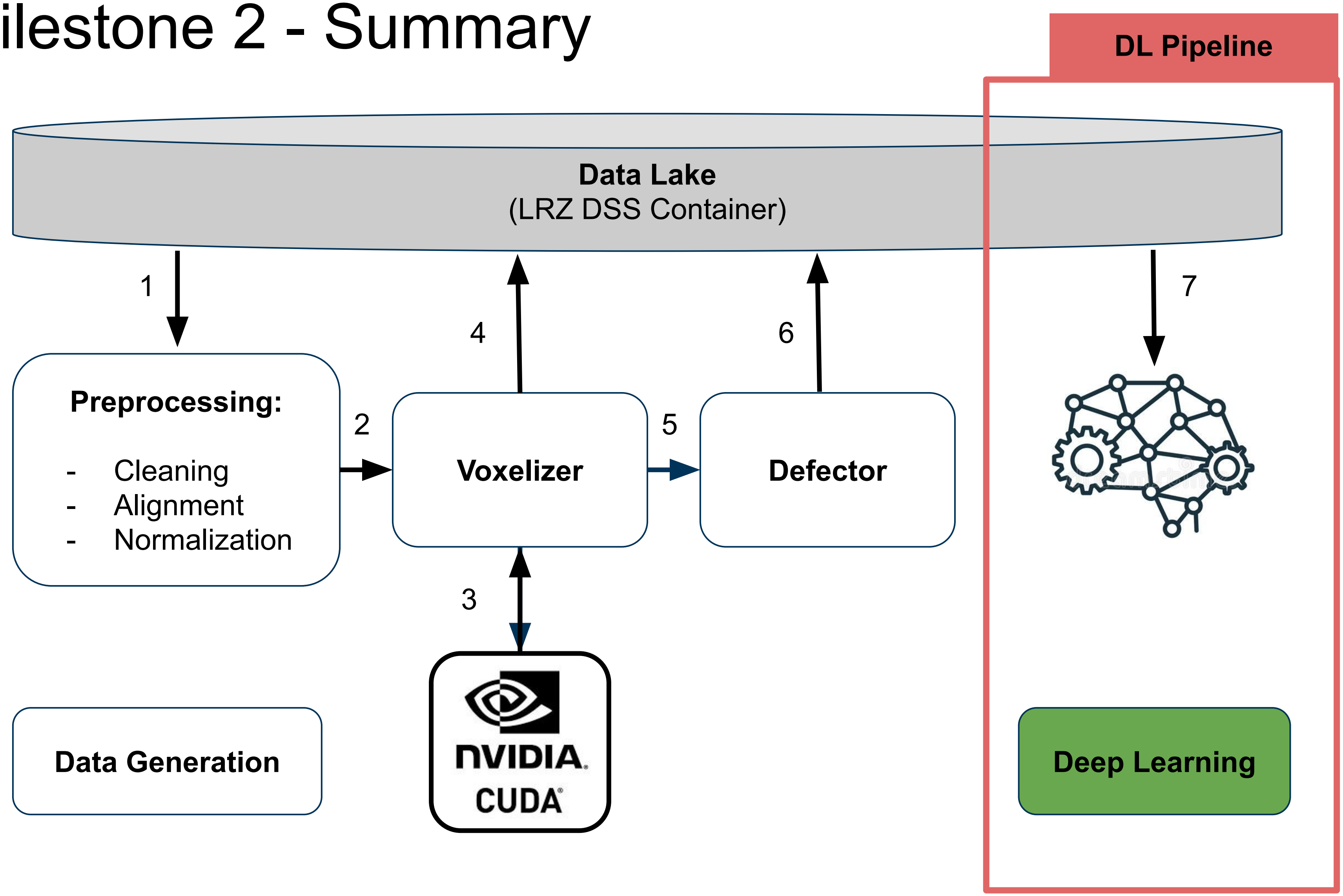
- 2.1 **Motivation - Felix**
- 2.2 **Horovod Distributed Training - Ahmed**
- 2.3 **Deep Learning Modelling: Rotated Data - Nouhayla**
- 2.4 **Deep Learning Modelling and Experimentation - Adi**
- 2.5 **Performance Analysis - Johannes, Felix**

3 **Wrap up: Next Steps**

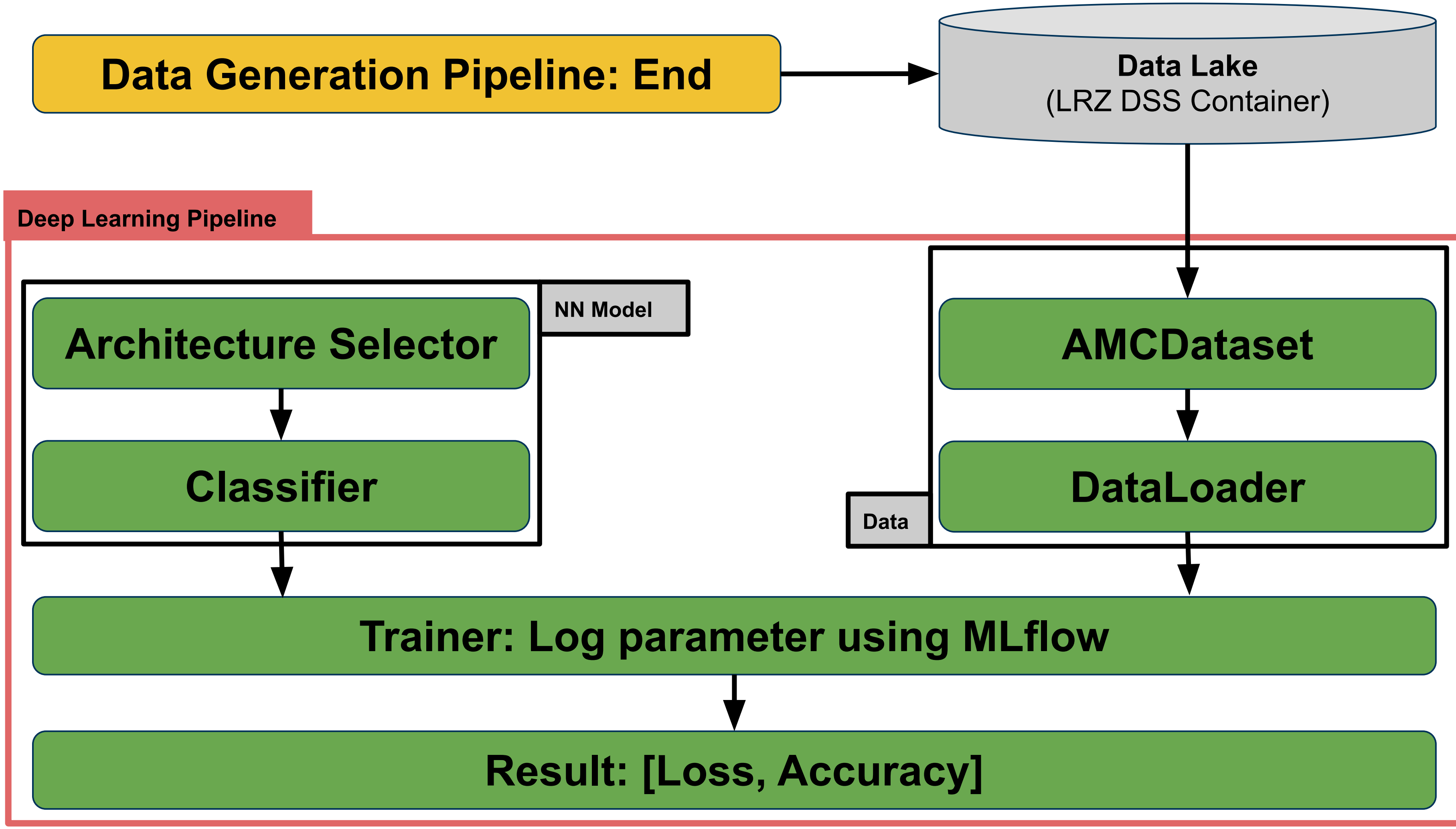
Milestone 1 - Summary



Milestone 2 - Summary

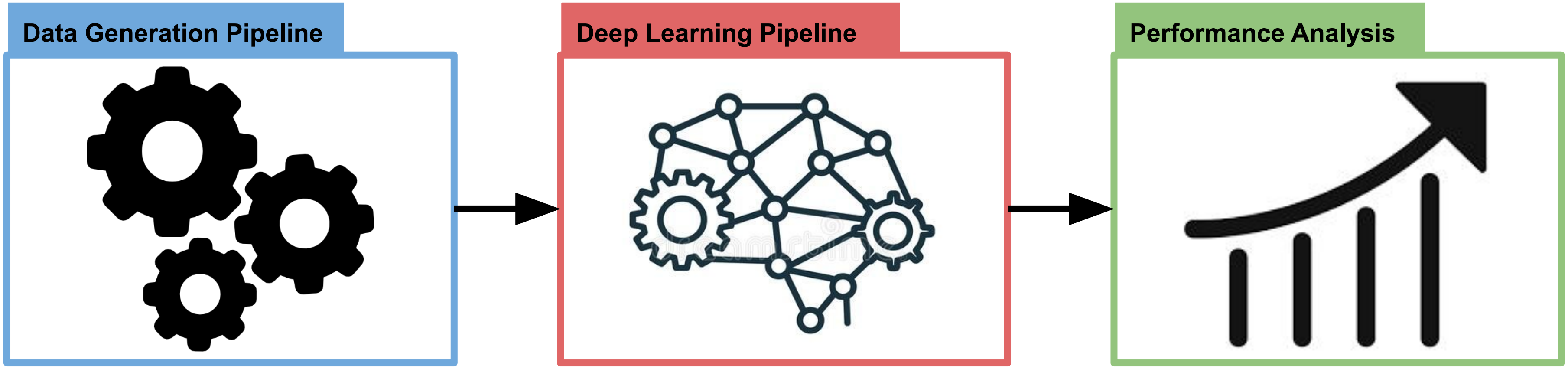


Milestone 2 - Summary



VOLKSWAGEN DATA:LAB

Today's Focus: Performance Analysis



- **F1 Score**
- **Confusion matrix**
- **ROC curve with AUC**
- **Analysis of false predictions**

Performance Metrics

F1-Score

- "Harmonic mean" of precision and recall

$$\text{Precision} = \frac{tp}{tp + fp}$$
$$\text{Recall} = \frac{tp}{tp + fn}$$

- Range: [0, 1]
"F1-Score of 1 is indicating a perfect precision and recall"

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}$$

ROC / AUC Curve

- Top left corner: Ideal point
- Large "AUC" is desirable

Confusion Matrix

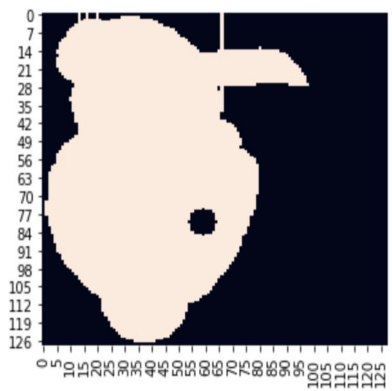
- Diagonal elements: correct prediction
- Off-diagonal elements: wrong predictions

VOLKSWAGEN DATA:LAB

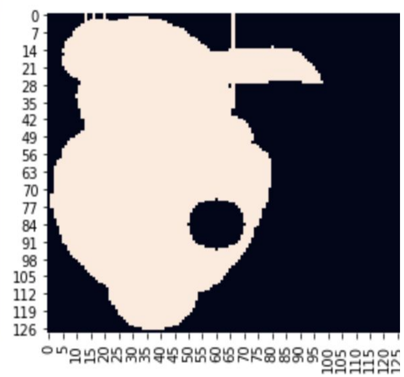
Training/Validation Properties

- **Types of defects:**

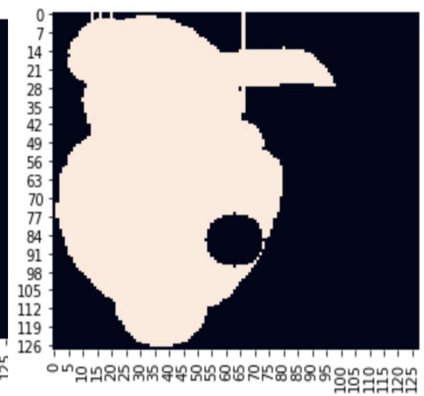
non-printable



printable



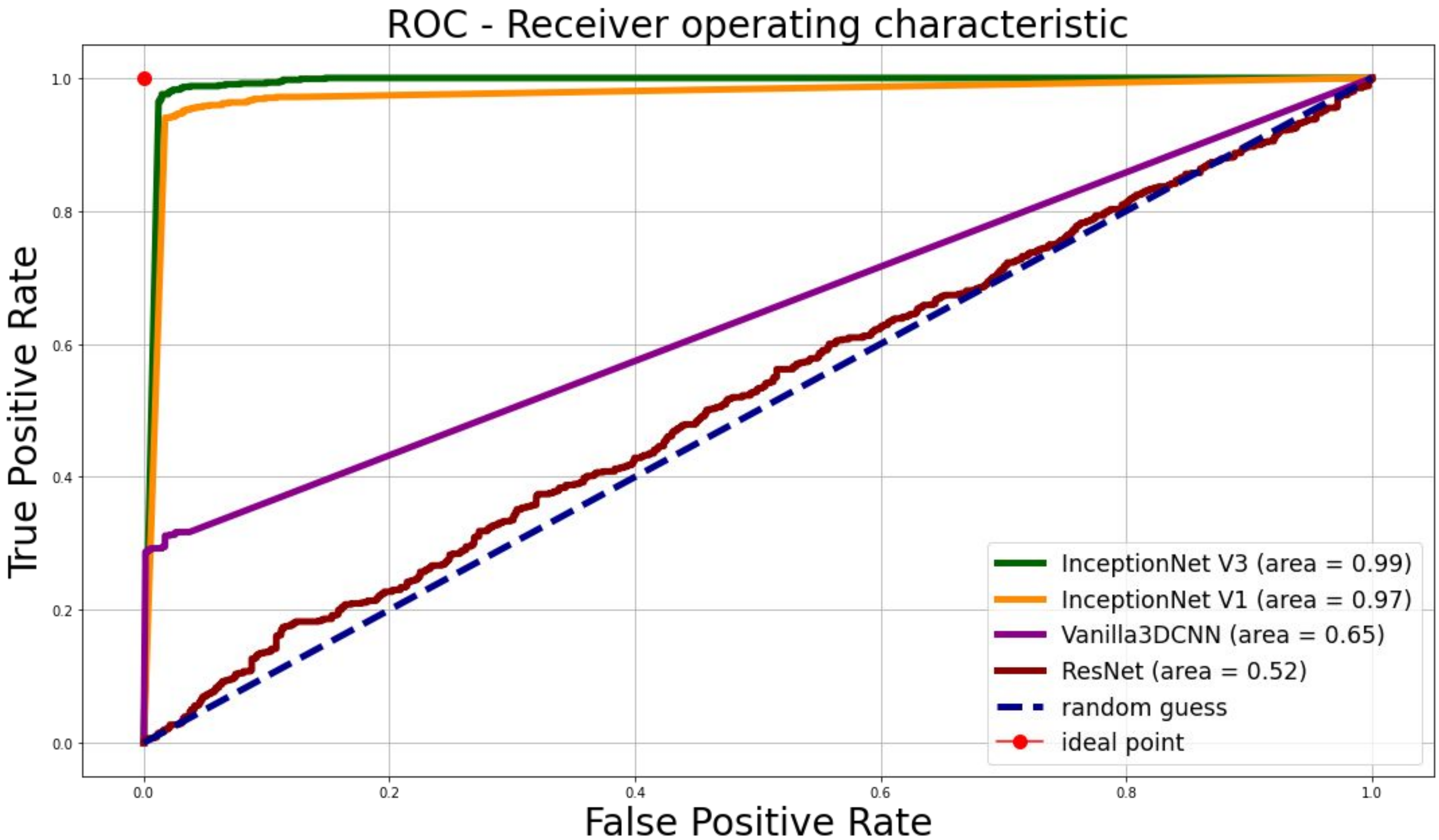
non-printable



- **Dataset:** This means a total of **7430** samples
- **Split ratio (random):**
    - Training: 80 % This means a total of **5944** samples
    - Validation: 20 % This means a total of **1486** samples

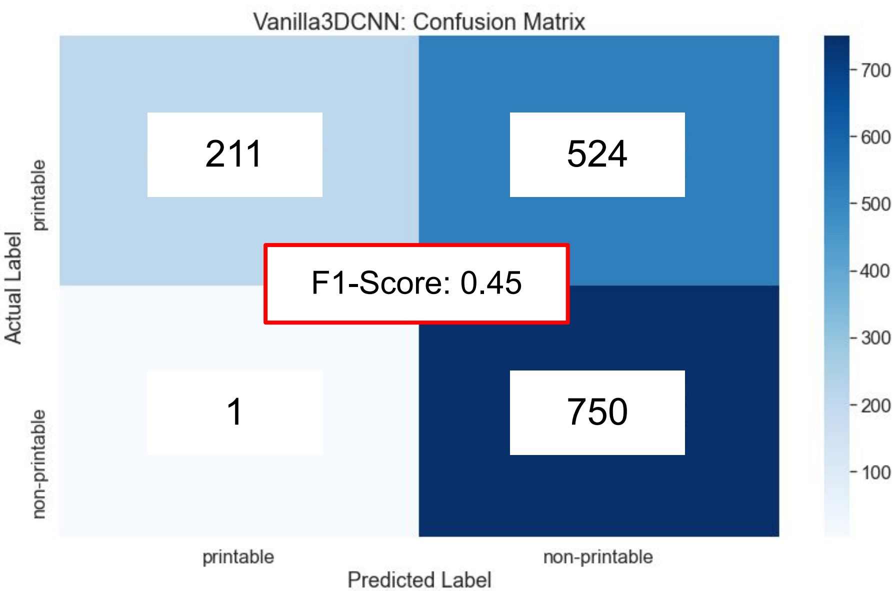
- **Assumption:** Training and validation set are **<u>balanced</u>**

Result Evaluation - ROC/AUC

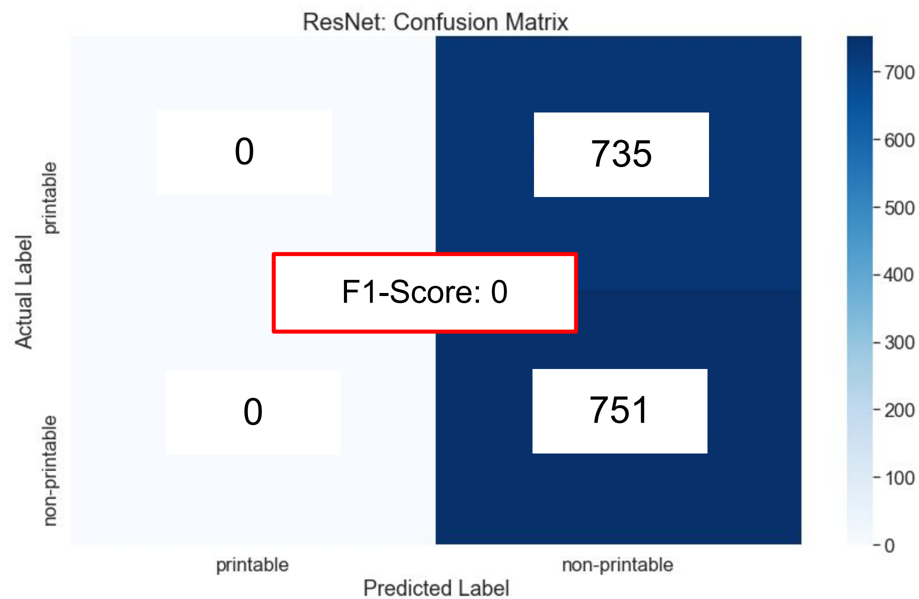


Result Evaluation - Confusion Matrix & F1-Score

Vanilla3DCNN:



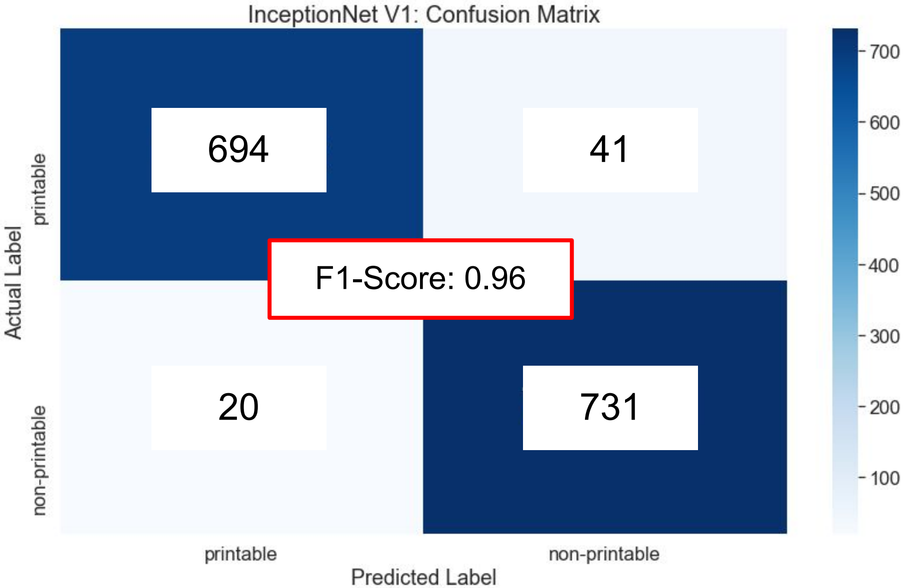
ResNet:



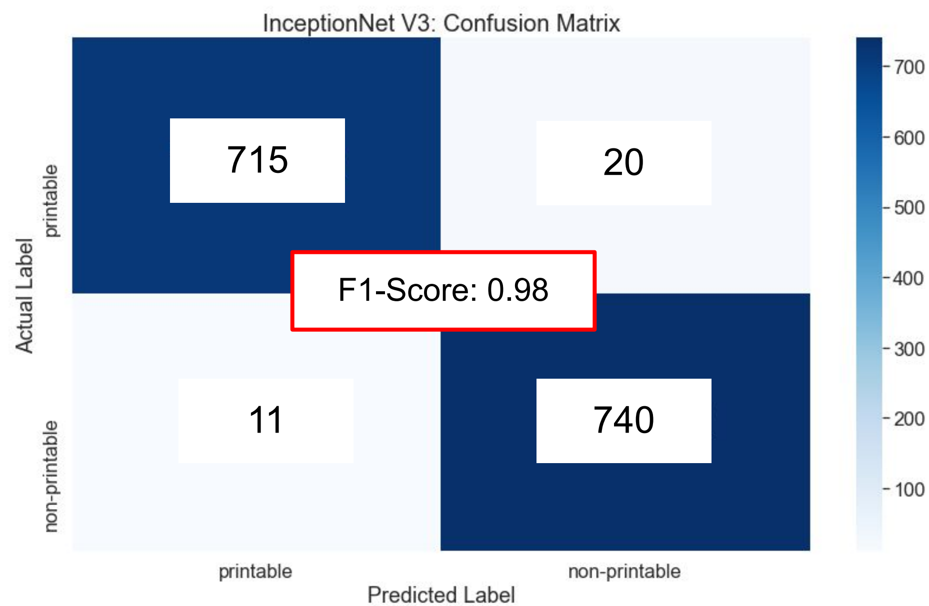
- **Vanilla3DCNN: Seems to have problems with “hard defects”**
- **ResNet: Does not learn anything, only “non-printable predictions”**

Result Evaluation - Confusion Matrix & F1-Score

InceptionNet V1:



InceptionNet V3:

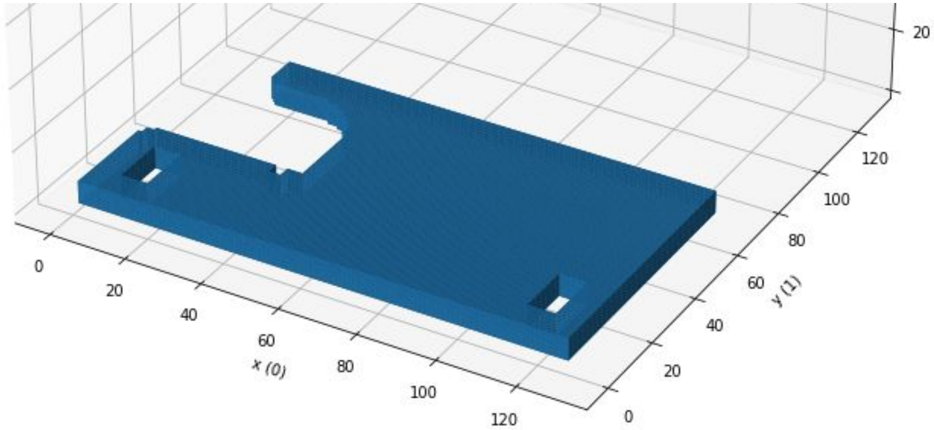


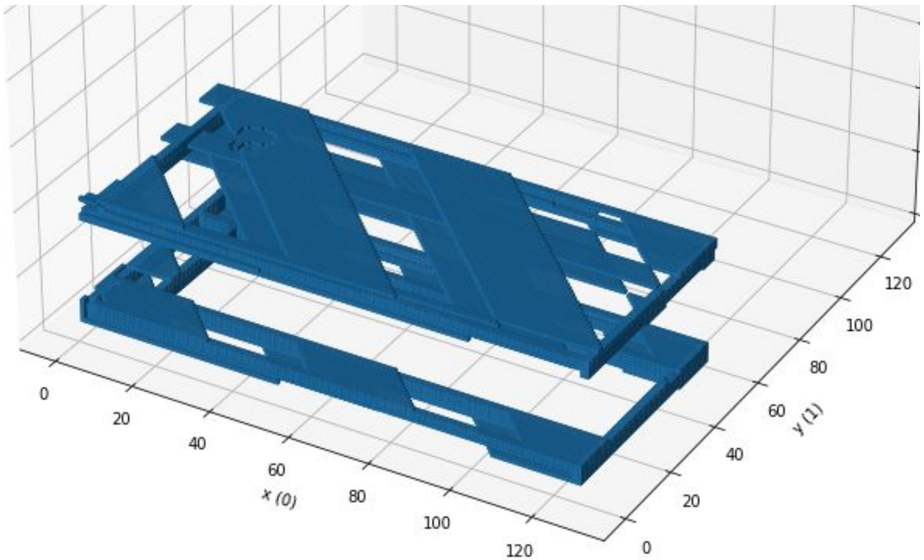
- **Somewhat same prediction "ability"**
- **Parameter setting of V3 performs better than V1 (9.3 M | 17.9 M)**

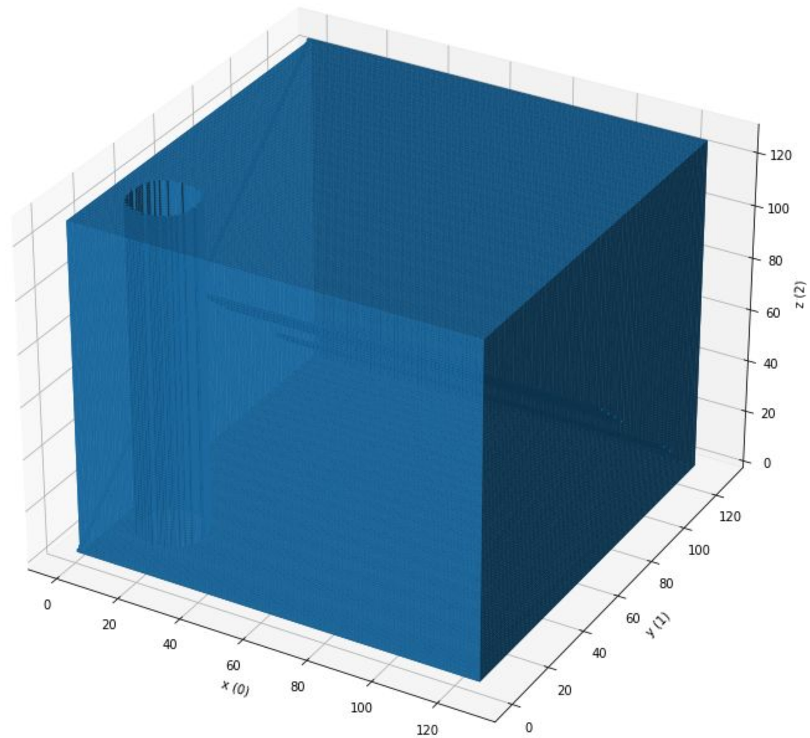
Evaluation of failed models (InceptionNet V1)

Defect type	Counter	
nonprintable_defect_border3	10	
printable_defect_middle10	12	
nonprintable_defect_middle5	24	← 1.
No Defect	16	← 2.

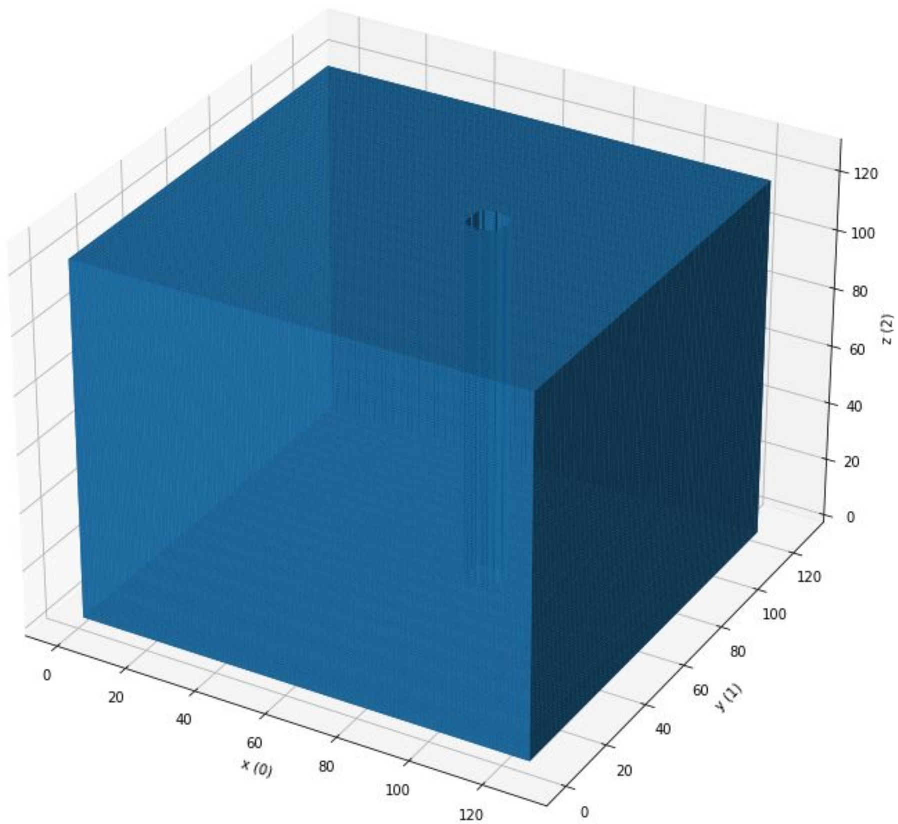
Comparison of models in detail (InceptionNet V1)

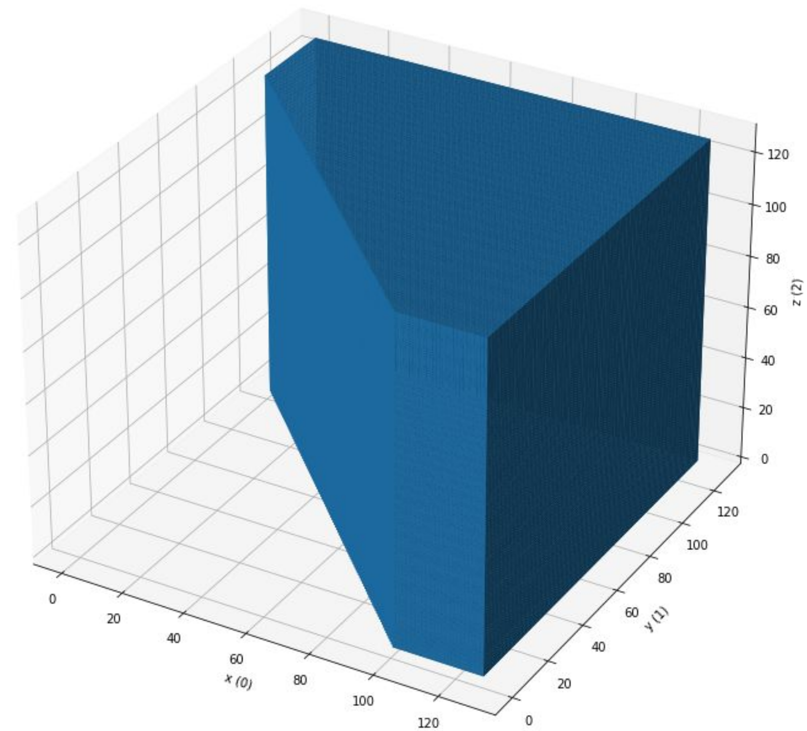






Comparison of models in detail (InceptionNet V1)





Identified Limitations (InceptionNet V1)

- 3D models are **too complex**
- 3D model already contains an **edge like hole**
- **Defects from Voxelization**
- But the deep learning model also fails at a few 3D models that look **very clean and easy to detect**.

Wrap up & Next steps

- Experiments done
  - logical results, reasoning out of common sense
  - overall result: satisfying

- Complete focus on the final report final presentation:
  - Final evaluations
  - Plots
  - Writing

Sources

[1] Ian Gibson et. al: Additive Manufacturing Technologies

[2] Mary Kathryn Thompson et. al: Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints

[3] Mehrpouya, Mehrshad et. al: The Potential of Additive Manufacturing in the Smart Factory Industrial 4.0: A Review

[4] Wohlers, T.T.: Wohlers Report 2021: 3D Printing and Additive Manufacturing : Global State of the Industry

[5] Jörg Bromberger and Richard Kelly: Additive manufacturing: A long-term game changer for manufacturers

[6] https://www.volkswagenag.com/de/news/stories/2018/12/brake-calipers-and-wheels-now-from-a-3d-printer.html#

[7] https://www.bmwgroup.com/de/news/2020/additive-manufacturing.html

[8] https://www.airbus.com/public-affairs/brussels/our-topics/innovation/3d-printing.html

[9]https://medium.com/swlh/pytorch-mlflow-optuna-experiment-tracking-and-hyperparameter-optimization-132778d6defc