Topology Optimization for MBB beam using Neural Network

Hongye Gu & Justin Guilfoyle
Mechanical Engineering

04/27/2021

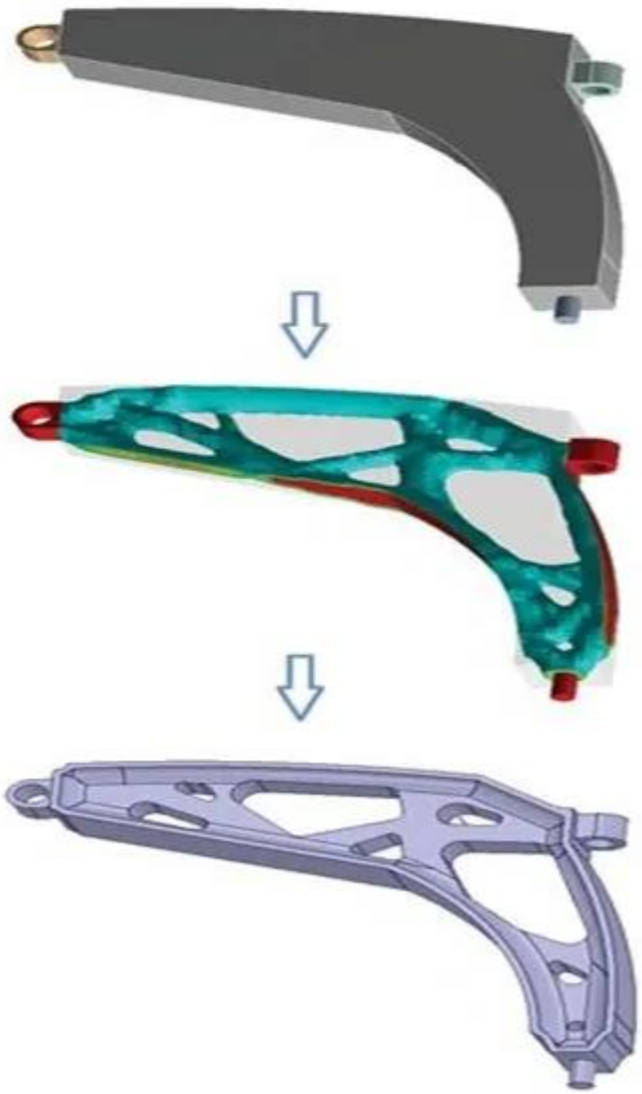
UCONN
SCHOOL OF ENGINEERING

UTC Institute
for Advanced
Systems Engineering
UNIVERSITY OF CONNECTICUT

Engineering Background

Data-driven Topology optimization:

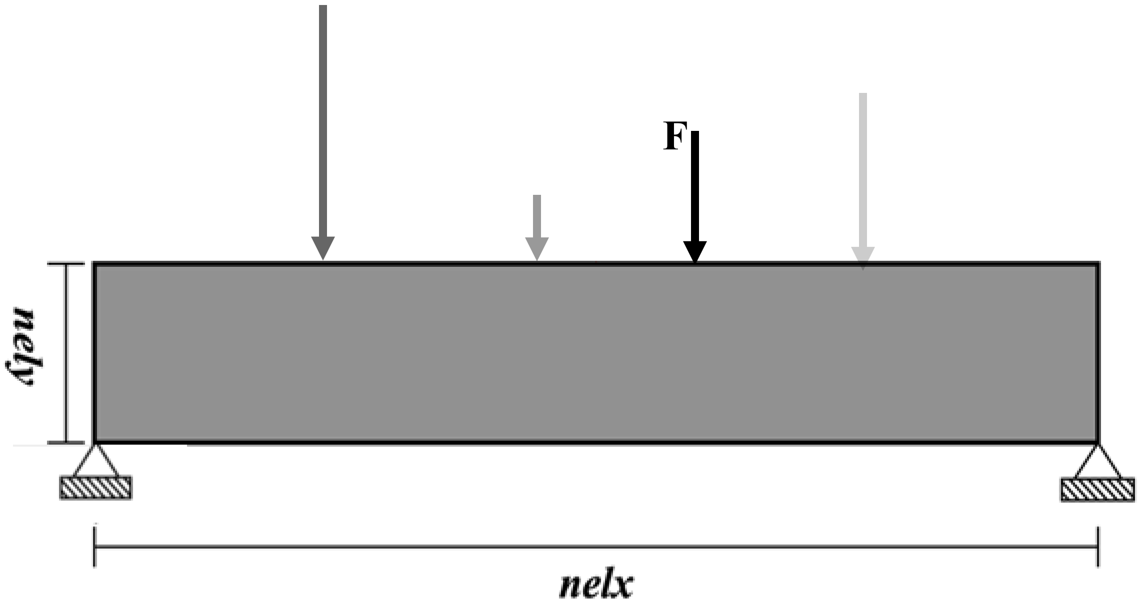
- Topology Optimization(TO) is the process of determining the optimum design for an engineering challenge where the design is defined as a set of point.
- All TO methods have a high computational cost
  - Finite Element Analysis
  - Gradient-Based Optimizer
  - Genetic Algorithms
- Machine learning can be used to supplement or replace the costly TO algorithm
- NN Designs may offer preliminary designs for TO which would reduce computational costs

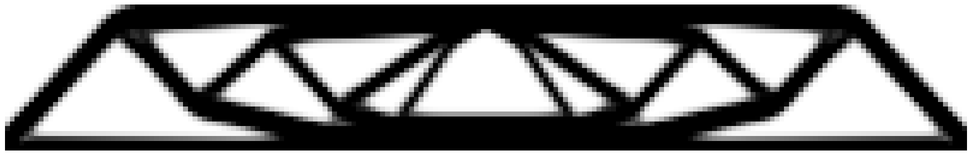


[1] Zhou, Ying, et al. "A new data-driven topology optimization framework for structural optimization." *Computers & Structures* 239 (2020): 106310.

Problem Statement

- **Problem setup:**
  - nelx = 200
  - nely = 30
- **Boundary conditions:**
  - Fixed joints on both bottom left and right corners
- **Input variables:**
  - Force location (50-150)
  - Force magnitude (0-1)
- **Responses:**
  - 2D design for MBB beam
- **Performance Metric:**
  - Compliance





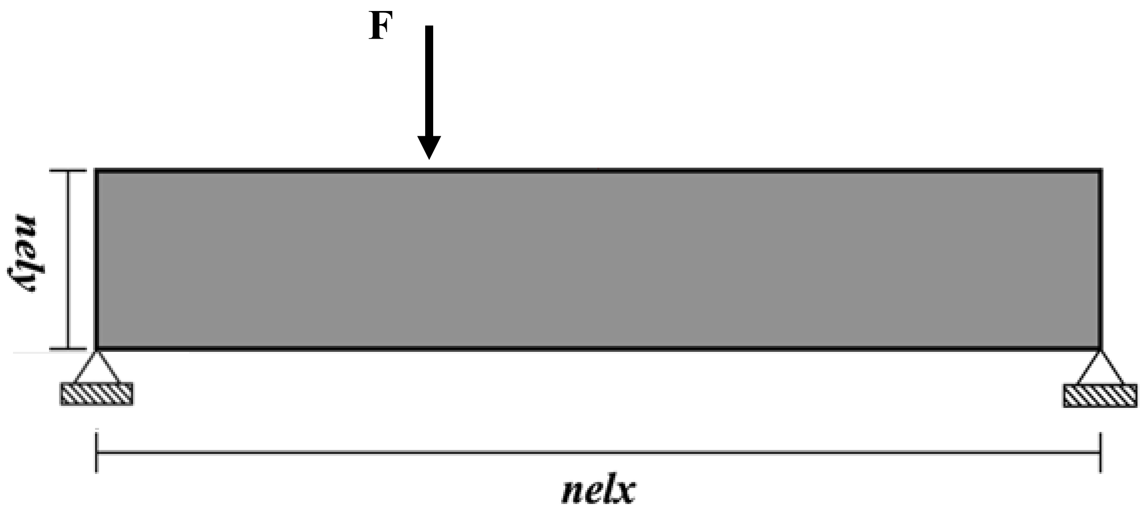
Data Source

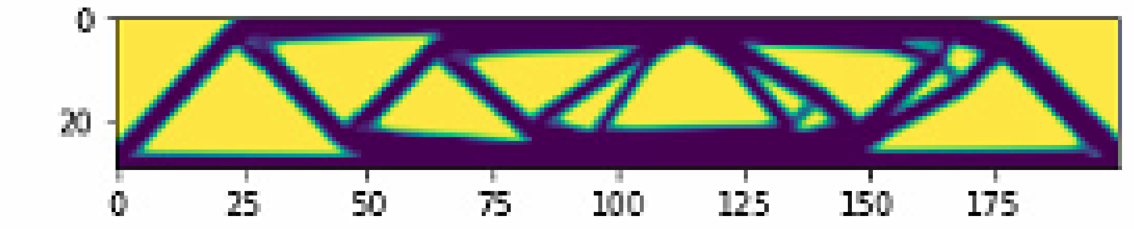
Data Generation:

- Simulation:
  - 99-line topology optimization Matlab code[2]
  - Robust - scalable with many features

- Design of Experiments(DOE):
  - Latin Hypercube Sampling

- Data types :
  - Training & Validation Data
    - X: Force Magnitude & Location (.csv)
    - Y: Design (.csv) & Compliance (.csv)
    - Y_{alt}: Design (.bmp)



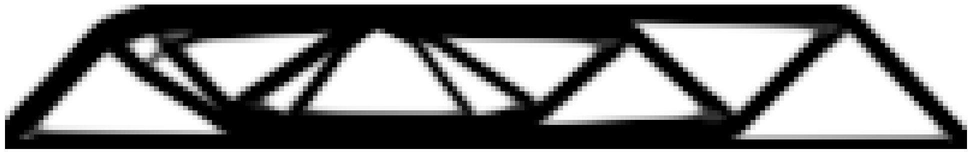


[2] Sigmund, Ole. "A 99 line topology optimization code written in Matlab." *Structural and multidisciplinary optimization* 21.2 (2001): 120-127.

Data Visualization

Dataset:

- 80 iterations of gradient-based TO
- Density ~ Element Shade
  - $\rho \in [0,1]$
- Black/White Design for TO



- 25/5 sample set for Single-load case
- 60/15 sample set for Single-load cases
- 80/20 sample set for Multi-load cases

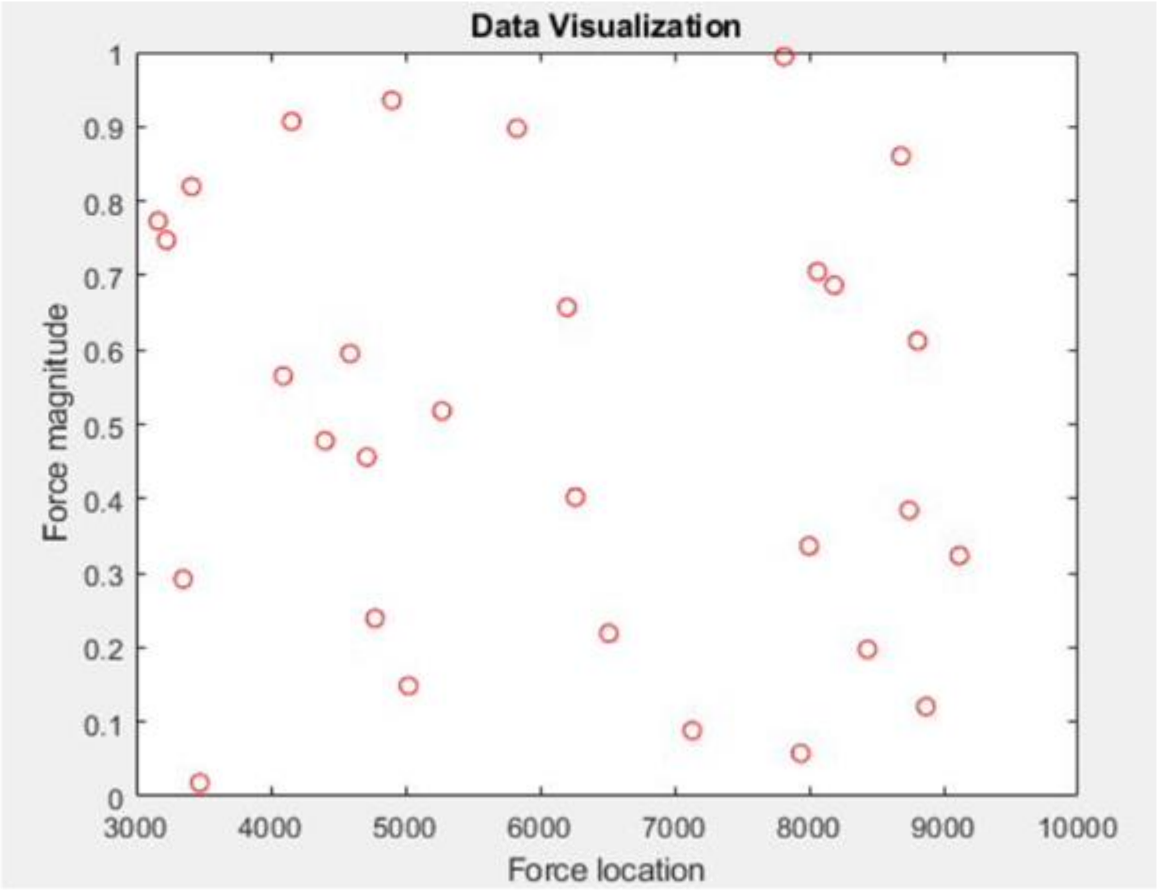


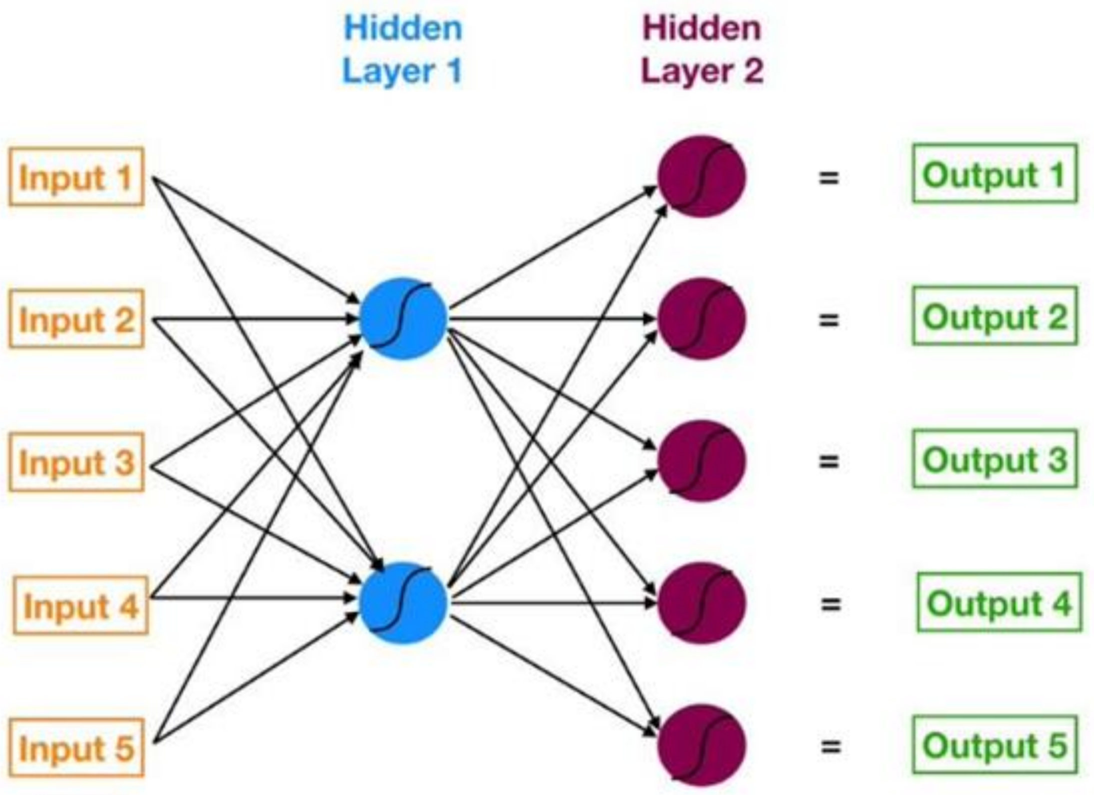
Fig. Data visualization for one load

UTC Institute for Advanced Systems Engineering
UNIVERSITY OF CONNECTICUT

Development of Machine Learning Models

Model:
- Neural Network

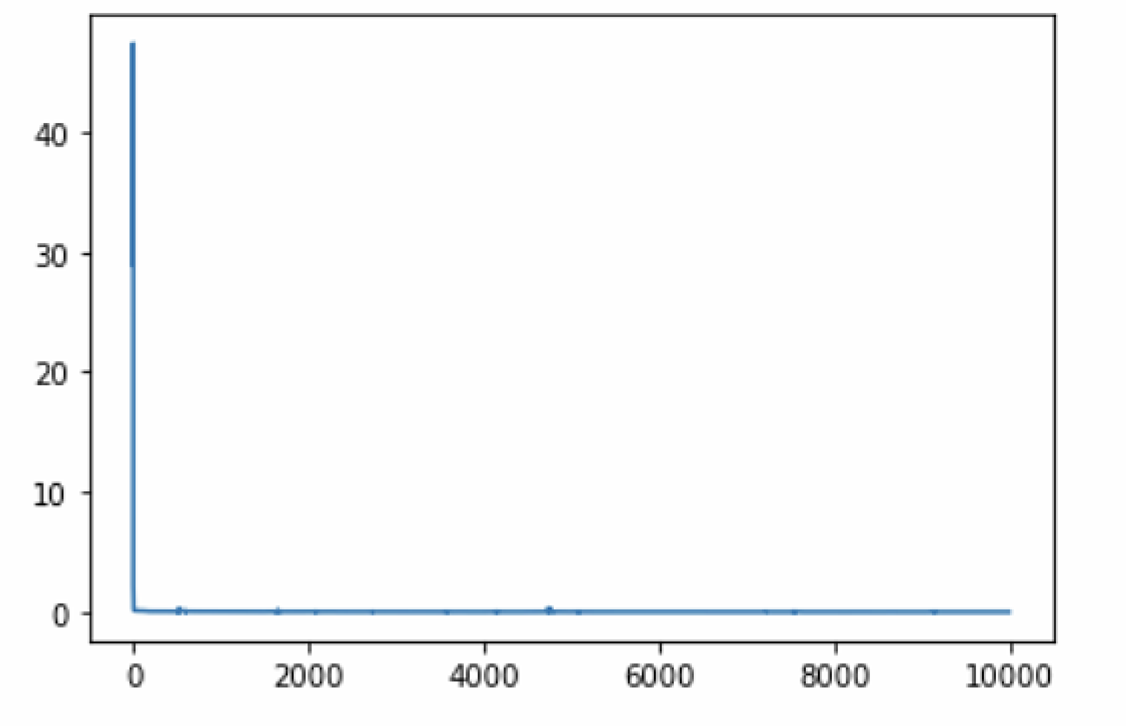
Reasons:
- Number of Inputs << Number of Outputs
  - Layers can produce more Output Values
- Material Configuration of Topology Optimization
  - Complex Layering of Numerical Subroutines
- Image Generation
  - Nonlinear Relationship between Pixels



Single Loading Cases

Development of Machine Learning Models

Training data set: 25
Validation data set: 5
Only dense layers

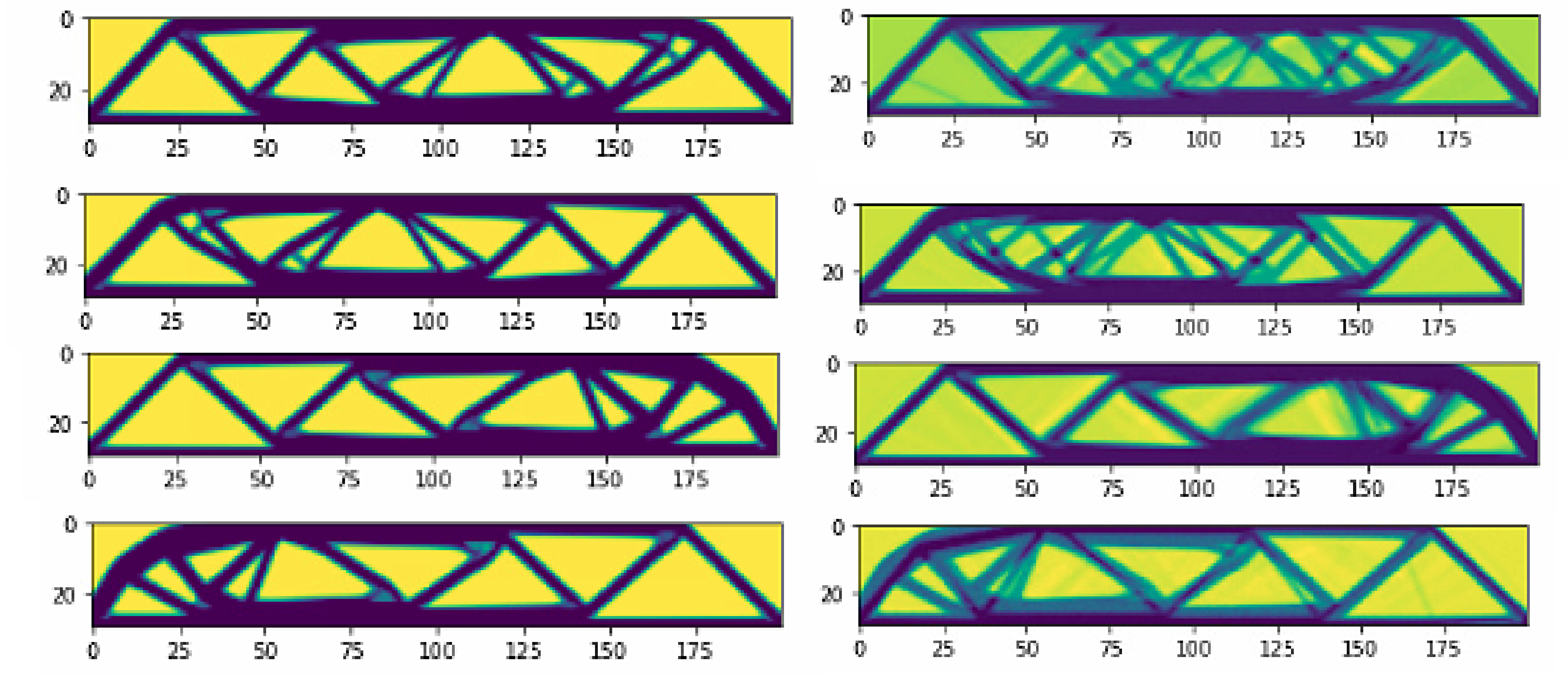


[inf, 0.0]

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 2)]	0
dense (Dense)	(None, 4)	12
dense_1 (Dense)	(None, 4)	20
dense_2 (Dense)	(None, 16)	80
dense_3 (Dense)	(None, 64)	1088
dense_4 (Dense)	(None, 256)	16640
dense_5 (Dense)	(None, 1024)	263168
dense_6 (Dense)	(None, 2048)	2099200
dense_7 (Dense)	(None, 4096)	8392704
dense_8 (Dense)	(None, 6000)	24582000
dense_9 (Dense)	(None, 6000)	36006000

Total params: 71,360,912
Trainable params: 71,360,912
Non-trainable params: 0

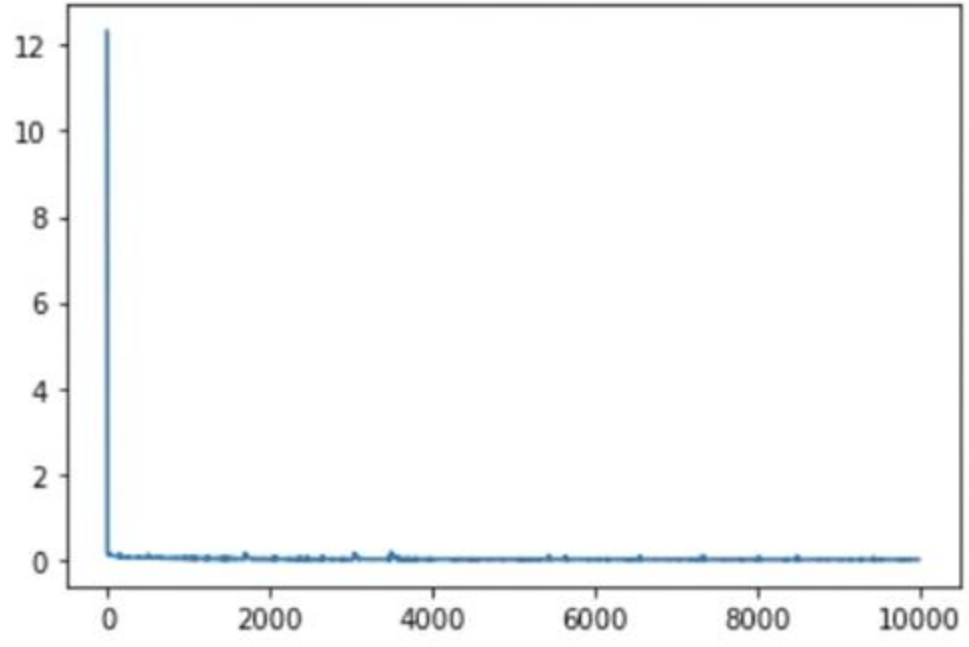
Results



Development of Machine Learning Models

Increasing the number of data and layers

Training data set: 60
Validation dataset: 15

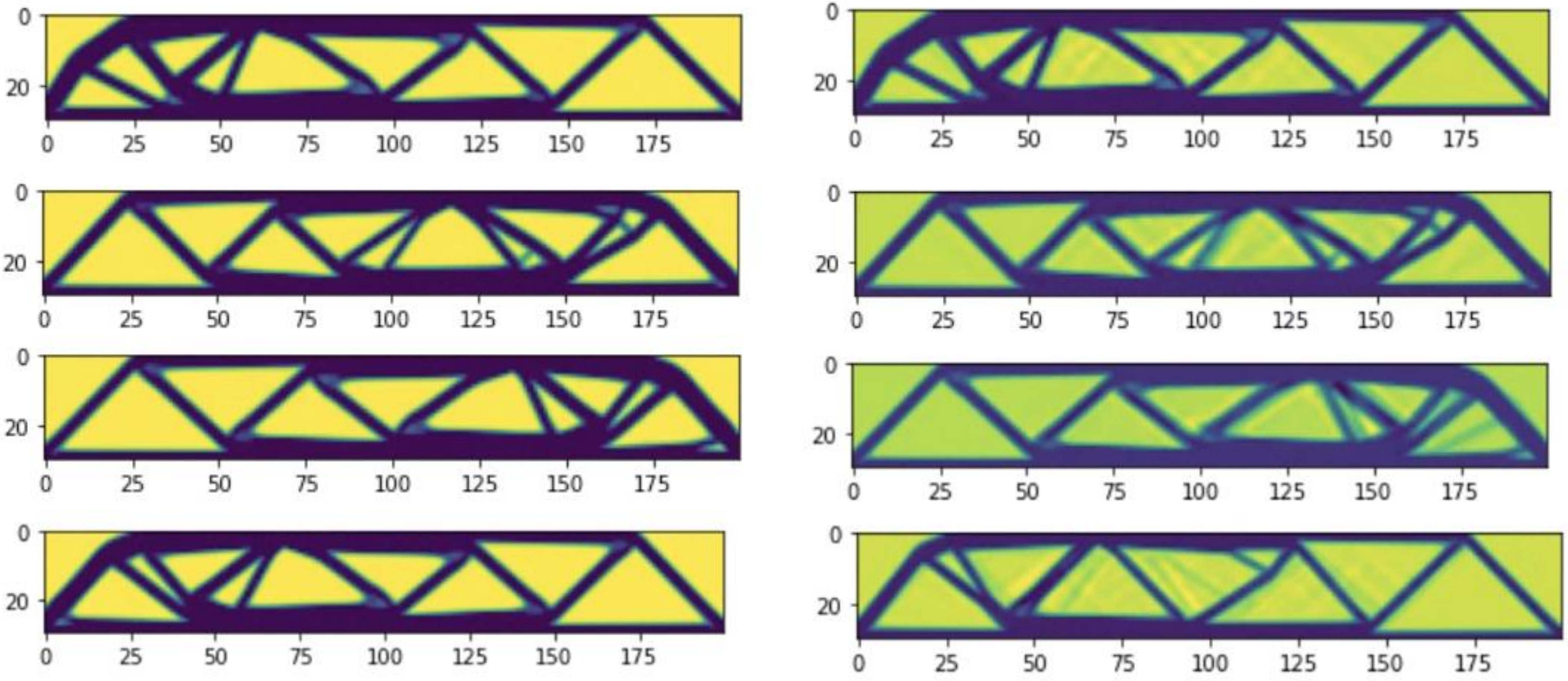


[0.04168110340833664, 0.0]

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 2)]	0
dense (Dense)	(None, 4)	12
dense_1 (Dense)	(None, 4)	20
dense_2 (Dense)	(None, 16)	80
dense_3 (Dense)	(None, 64)	1088
dense_4 (Dense)	(None, 256)	16640
dense_5 (Dense)	(None, 256)	65792
dense_6 (Dense)	(None, 1024)	263168
dense_7 (Dense)	(None, 1024)	1049600
dense_8 (Dense)	(None, 2048)	2099200
dense_9 (Dense)	(None, 4096)	8392704
dense_10 (Dense)	(None, 6000)	24582000
dense_11 (Dense)	(None, 6000)	36006000

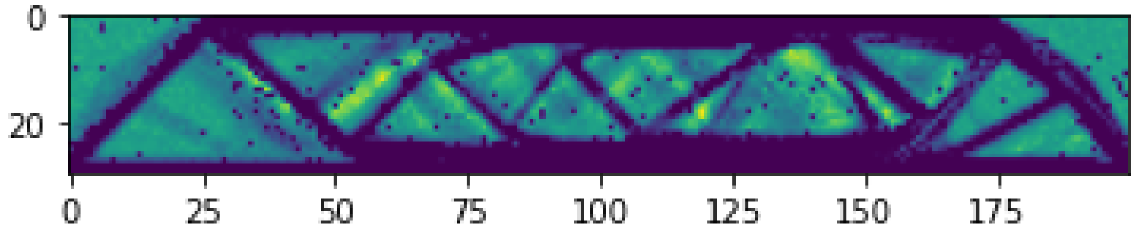
Total params: 72,476,304
Trainable params: 72,476,304
Non-trainable params: 0

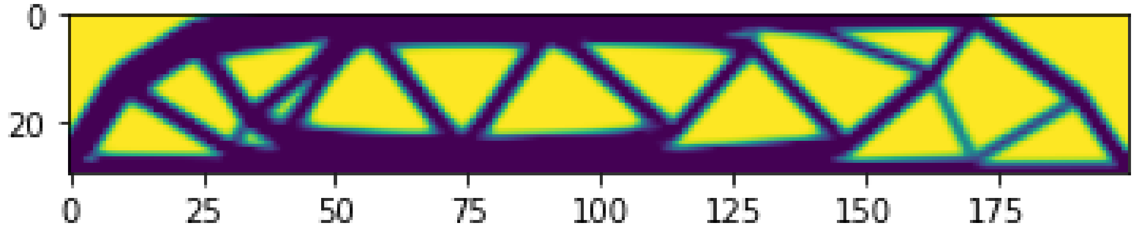
Development of Machine Learning Models



Alternative Layering Scheme

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 6)]               0
_________________________________________________________________
reshape (Reshape)            (None, 3, 2, 1)           0
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 3, 2, 4)           68
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 3, 2, 16)          6416
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 3, 2, 512)         131584
_________________________________________________________________
flatten (Flatten)            (None, 3072)              0
_________________________________________________________________
dense (Dense)                (None, 6000)              18438000
=================================================================
Total params: 18,576,068
Trainable params: 18,576,068
Non-trainable params: 0
_________________________________________________________________
```



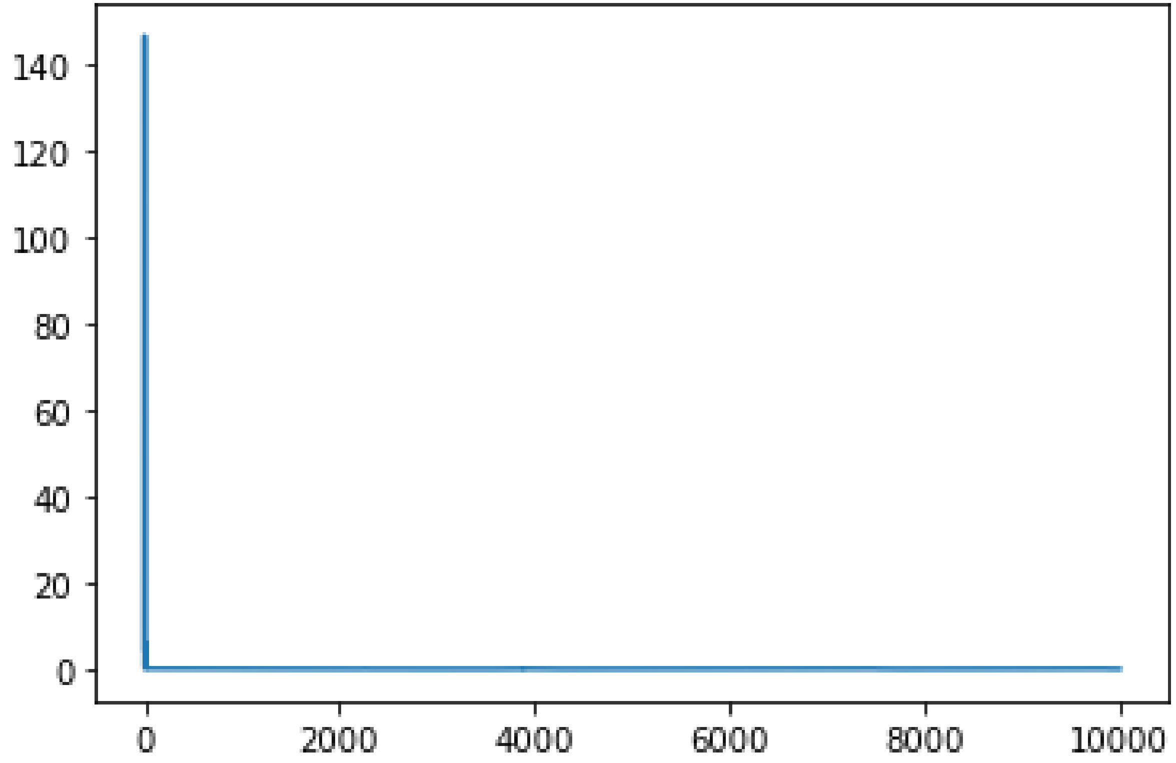


Multiple Loading Cases

Development of Machine Learning Models

Multiple loading case (3 loading)

Training data set: 80
Validation dataset: 20

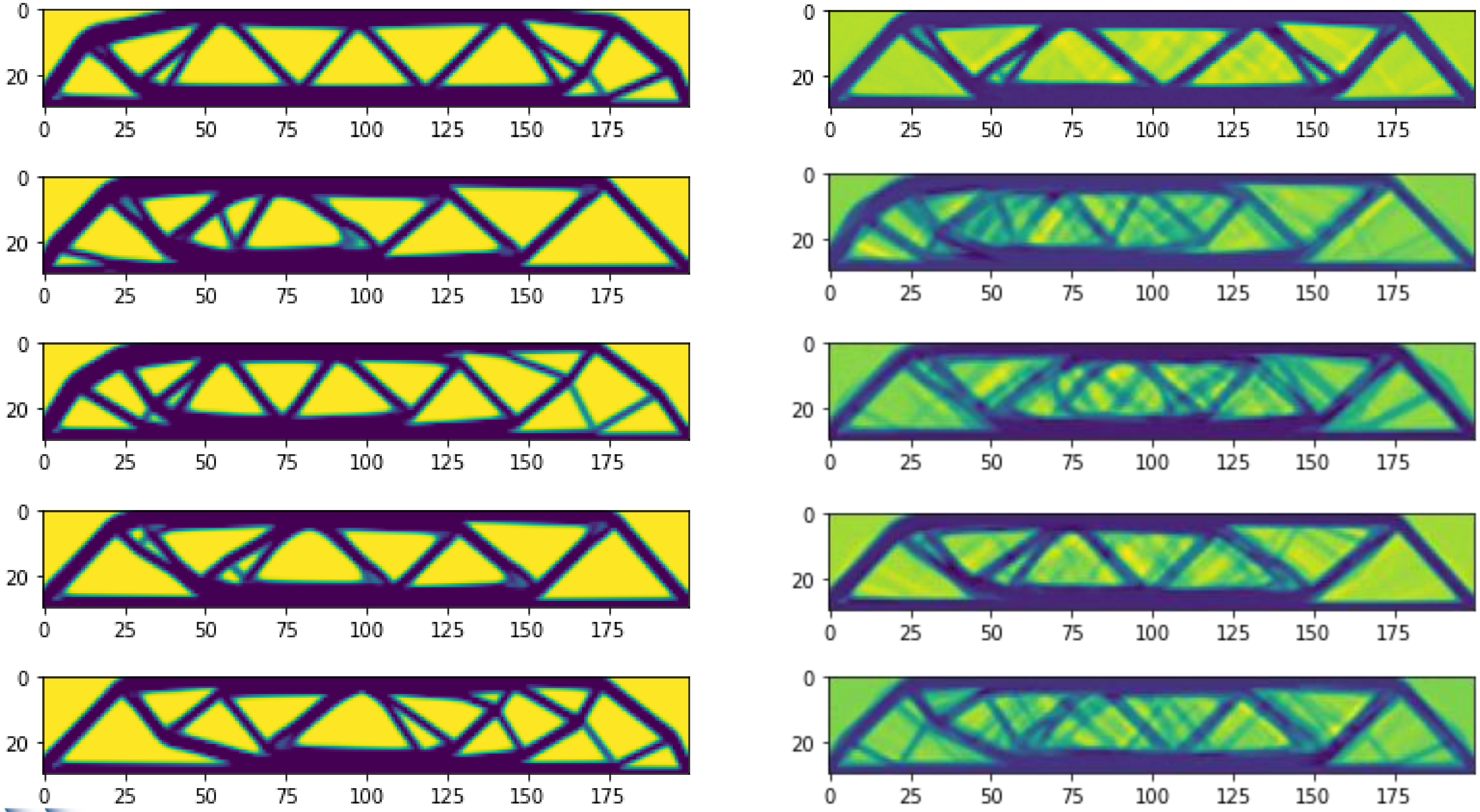


[0.10625715553760529, 0.0]

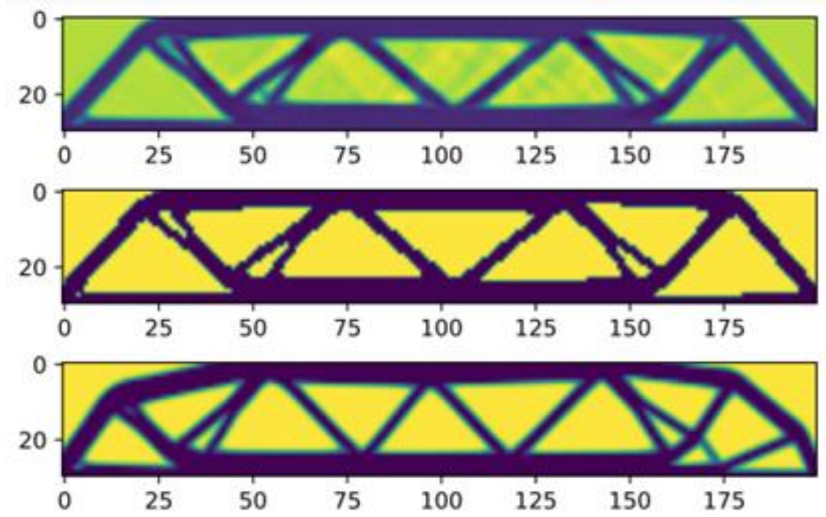
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 6)]	0
dense (Dense)	(None, 4)	28
dense_1 (Dense)	(None, 4)	20
dense_2 (Dense)	(None, 16)	80
dense_3 (Dense)	(None, 16)	272
dense_4 (Dense)	(None, 64)	1088
dense_5 (Dense)	(None, 64)	4160
dense_6 (Dense)	(None, 256)	16640
dense_7 (Dense)	(None, 256)	65792
dense_8 (Dense)	(None, 1025)	263425
dense_9 (Dense)	(None, 1025)	1051650
dense_10 (Dense)	(None, 4096)	4202496
dense_11 (Dense)	(None, 4096)	16781312
dense_12 (Dense)	(None, 4096)	16781312
dense_13 (Dense)	(None, 4096)	16781312
dense_14 (Dense)	(None, 6000)	24582000
dense_15 (Dense)	(None, 6000)	36006000
dense_16 (Dense)	(None, 6000)	36006000

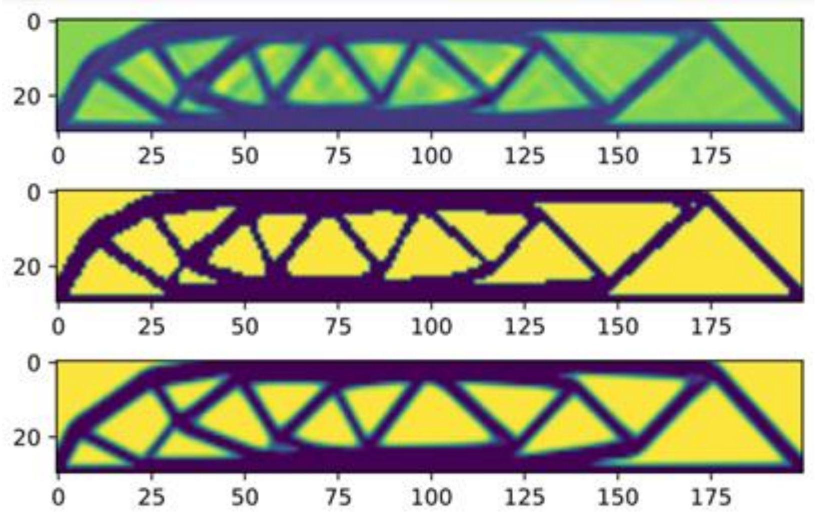
Total params: 152,543,587
Trainable params: 152,543,587
Non-trainable params: 0

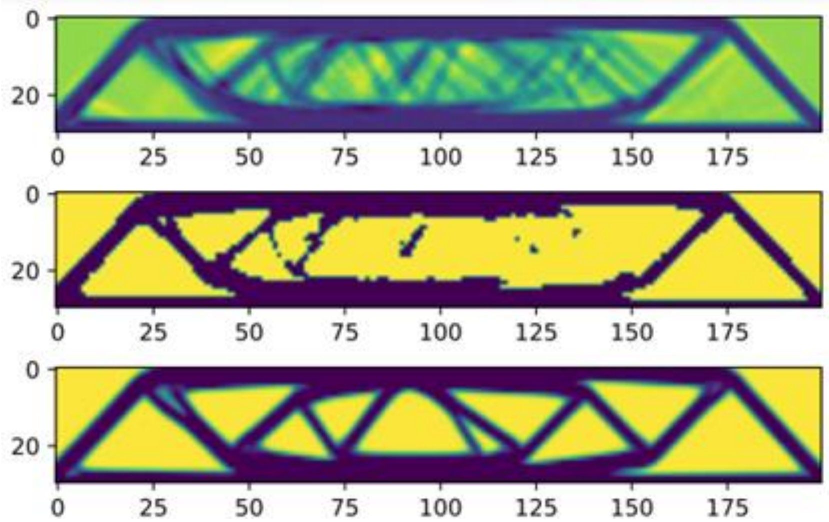
Results

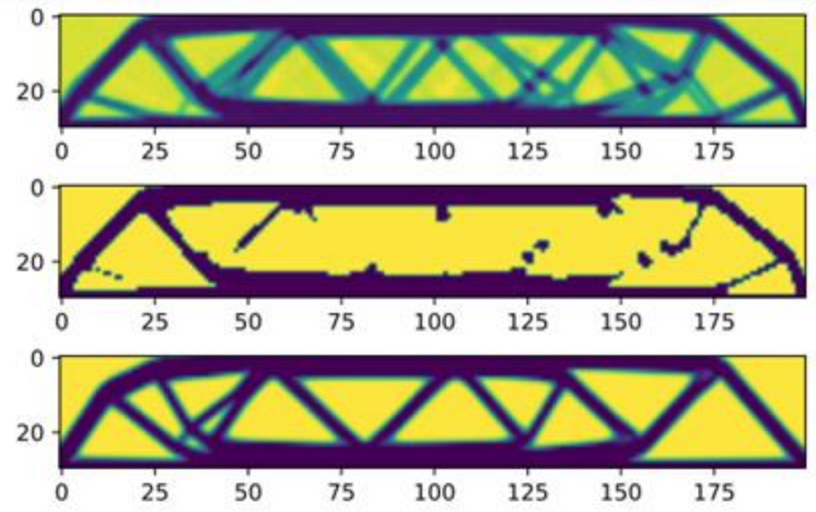

0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175
0 20
0 25 50 75 100 125 150 175

Feasibility of NN Designs







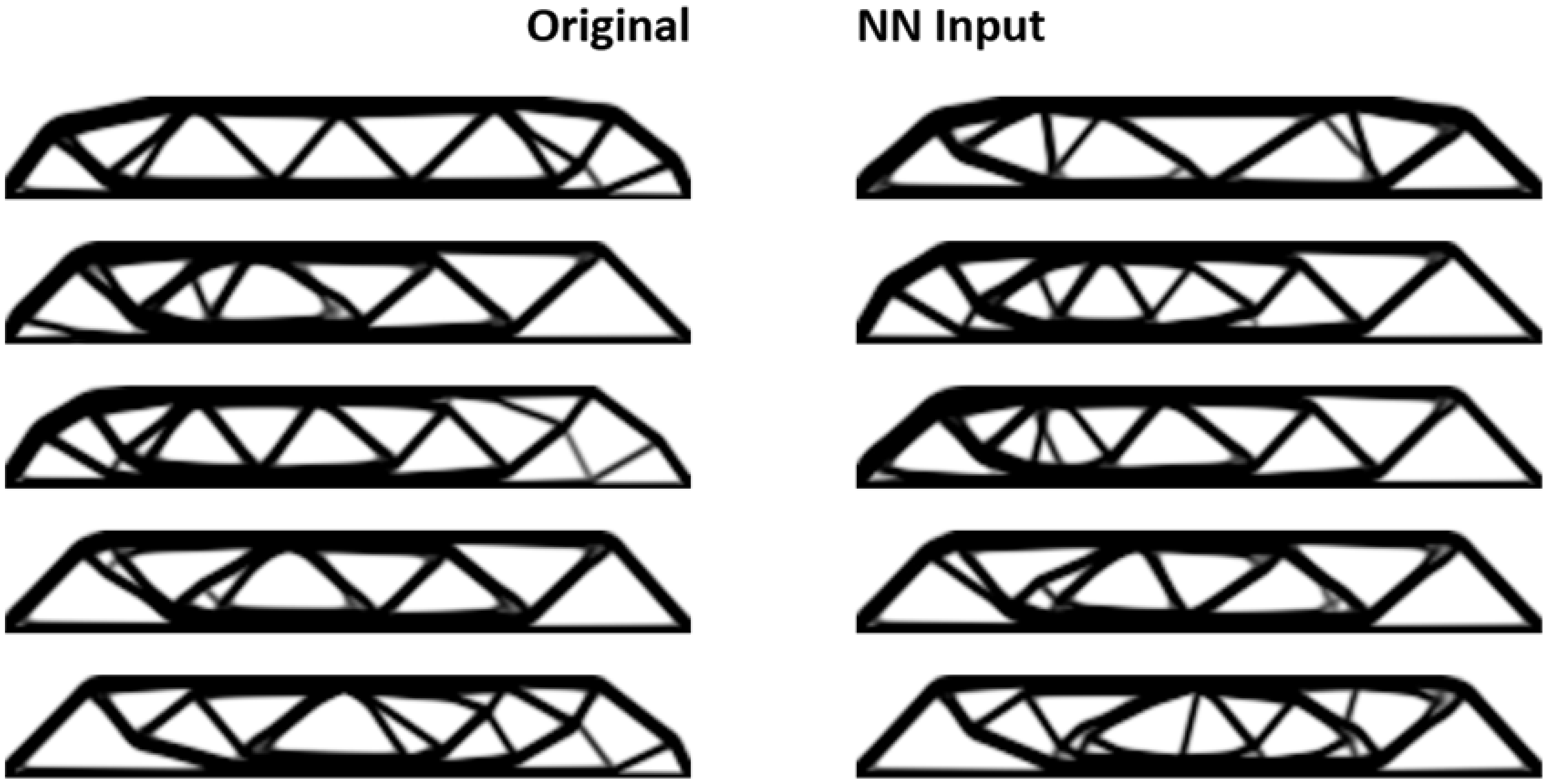


Elements with density less than 35% are set to zero

NN Designs as TO Input

Results (40 Iterations)

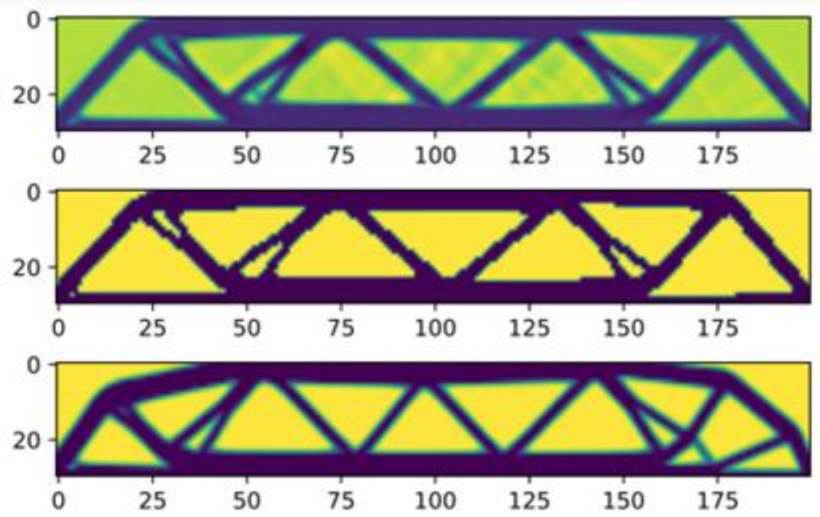
- Compliance improved for 35%
  - 0.065% Relative Improvement
  - 0.946% Relative Error
- Time Savings (2.5 s/iter)
  - 80 iterations → 200s
  - 40 iterations → 100s
- <u>**50% Time Reduction**</u>



Density: $\rho < .35 \rightarrow 0$ && $\rho > .65 \rightarrow 1$

Summary

Summary
- Identified a Problem:
  - Computational cost of Topology Optimization
- Performed Data Generation
  - TOP99[2]
- Applied a Machine Learning Technique:
  - Neural Network



Results
- Unfiltered NN designs are not feasible due to intermediate densities
- Filtered NN Designs can be used to support feasible designs

Application
- Initial density field for topology optimization algorithm
  - Reduces overall computational cost

[2] Sigmund, Ole. "A 99 line topology optimization code written in Matlab." *Structural and multidisciplinary optimization* 21.2 (2001): 120-127.

Q&A

Thank you!