AI for Computational Mechanics
Centre for Building Materials
Technical University of Munich

Assignment 1

Development of a Neural Network for Microstructure-Property Predictions

Submitted by:

Isheanesu Roy Mugwagwa
Matriculation Number: 03771519

Master's Program in Computational Mechanics (COME)

Summer Semester 2025

Munich, Germany
June 20, 2025

Step 1: Data Preprocessing

The dataset consists of 645 grayscale microstructure images stored in `.npy` format and corresponding Young's modulus values (E_{eff}) stored in a CSV file. The preprocessing involved the following steps:

1. **Data Splitting:** The images were divided into training (70%), validation (15%), and test (15%) sets. Each subset was placed in separate folders named `Train`, `Validation`, and `Test` respectively.
2. **Drive Integration:** All image folders and the CSV metadata file were uploaded to Google Drive for seamless access and usage in Google Colab.
3. **Mapping Images to Targets:** A custom data loading function was implemented to associate each image with its corresponding E_{eff} value using the CSV file. File name matching was performed to ensure accurate linking.
4. **Verification:** A visual inspection was conducted to confirm that each image correctly corresponded to its E_{eff} value.

Step 2: Model Architecture and Hyperparameters

A custom Convolutional Neural Network (CNN) named **UltraWideCNN** was developed to predict the effective Young's modulus (E_{eff}) from 65×65 grayscale microstructure images. The architecture is deep and wide, designed to capture hierarchical spatial features through four convolutional layers, each followed by ReLU activation and max pooling to progressively reduce the spatial dimensions. To improve generalization and prevent overfitting, dropout with a probability of 0.5 is applied after the convolutional blocks and the first fully connected layer.

The model was trained using the Adam optimizer with a learning rate of 1×10^{-4}, and the loss function used was Mean Squared Error (MSELoss). A dropout rate of 0.5 and a batch size of 32 were used to further support generalization and training stability.

Step 3: Training, Validation, and Testing Strategy

The model was trained for 30 epochs using the Adam optimizer with a learning rate of 1×10^{-4}. After each epoch, performance was evaluated on the validation set to monitor overfitting and guide model checkpointing. The best-performing model, determined by the lowest validation loss, was then evaluated on the test set. The model demonstrated strong generalization capabilities, with 77.27% of test predictions falling within a 30% error margin. A random sample of 10 test predictions was also examined, confirming the model's reliability and superior performance compared to other architectures previously explored.

All implementation was done in Google Colab, and the dataset was loaded from Google Drive.

Step 4: Loss Curve

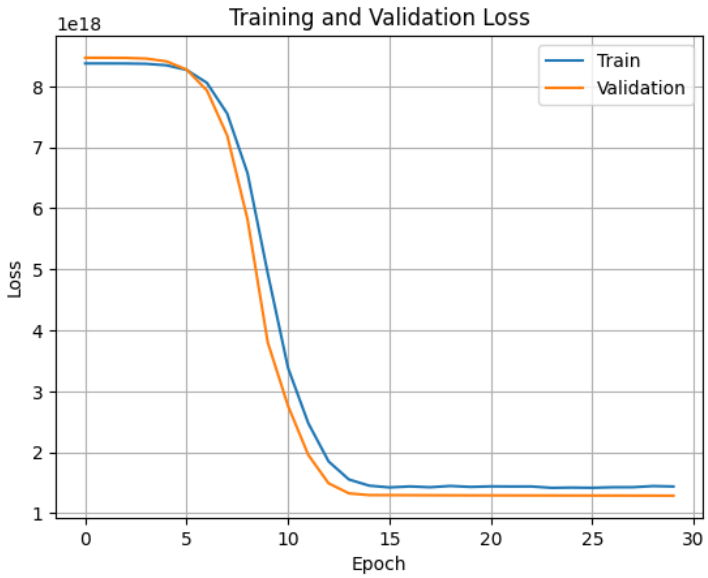

Figure 1: Loss curve of the final trained model

Step 5: Model Predictions Compared to True Values

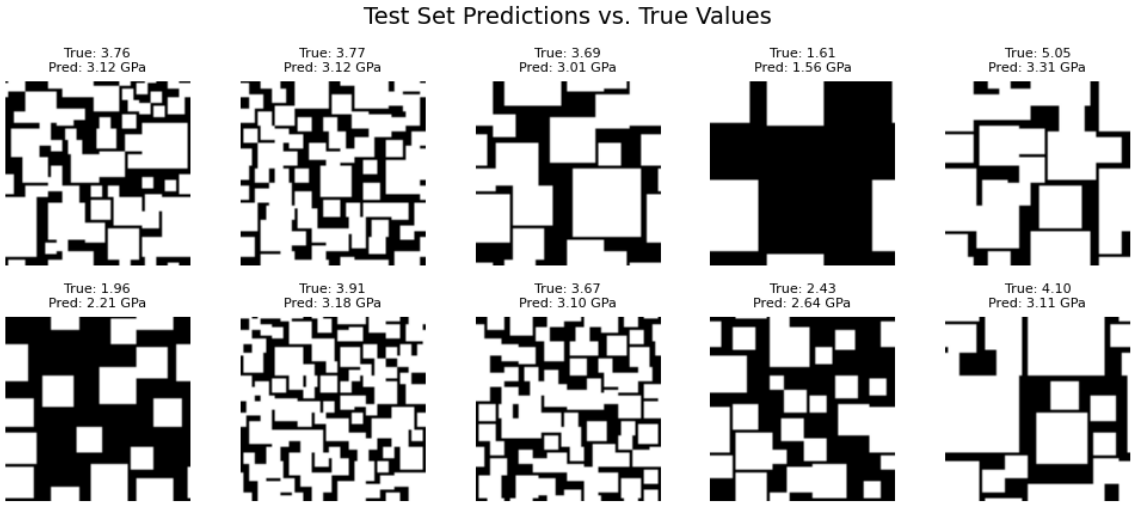

Figure 2: Comparison of model predictions and true E_{eff} values on 10 randomly selected test samples

Step 6: Bonus Step — Additional Model Experiments

As part of model exploration, several alternative architectures were tested, including EfficientNet-B0, MobileNetV2, and VGG. However, these pre-trained models failed to achieve satisfactory performance, with test prediction accuracy within the 30% error margin falling significantly below expectations.

In addition, a second custom CNN was developed, incorporating Batch Normalization layers after each convolutional layer to improve training stability and convergence.

While this model did not outperform the main model in terms of training and validation loss, it demonstrated notably better generalization on the test set. Specifically, this secondary custom model achieved 95.45% of test predictions within a 30% error margin, outperforming all other models in terms of test accuracy.

The model architecture and its performance results are included in this assignment submission for reference.