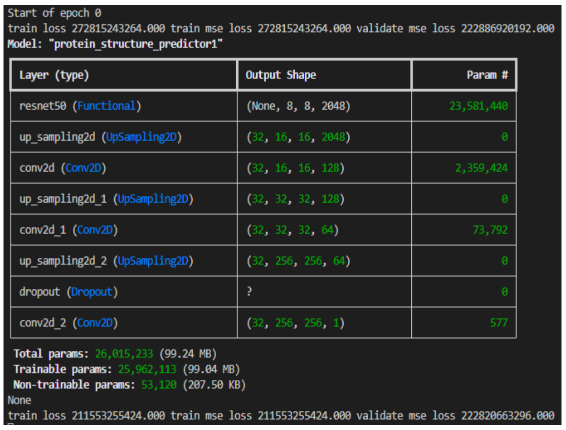
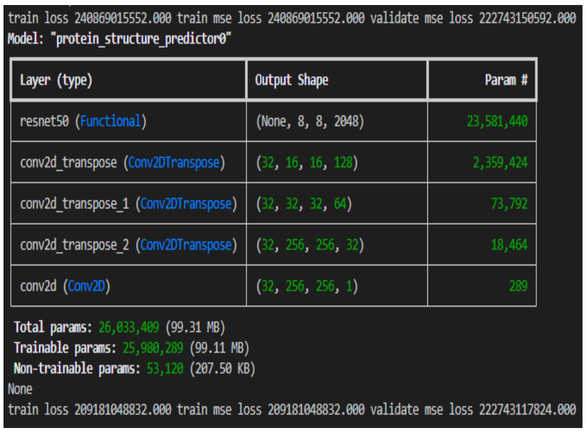
Experimental Design and Methods

The initial model, ProteinStructurePredictor0, uses a ResNet50 architecture as the backbone for feature extraction, with no pre-trained weights, and uses the distance matrix as input. Due to the output size of ResNet50 the model then applies three Conv2DTranspose layers to upsample the feature maps. These layers are designed to increase the spatial dimensions of the data while reducing the number of channels, producing an output that matches the shape of the distance matrix. The last layer is the initial Conv2D output layer with a linear activation, which produces the predicted distance matrix. This model summary can be seen in figure 1.

I hypothesized that replacing the Conv2DTranspose layers with UpSampling2D followed by Conv2D layers in the new model, ProteinStructurePredictor1, will improve the training. Transposed convolutions, while effective for upscaling, are known to introduce checkerboard artifacts. By using UpSampling2D for scaling, these artifacts can be avoided, providing smoother, more consistent upscaling. Due to the complexity of the ResNet50 model being the base for the output a more stable upscaling method should produce better results. The use of ResNet50, which is a relatively large model, may make the model prone to overfitting, especially on smaller datasets. Thus, the inclusion of a Dropout layer (with a dropout rate of 0.3) will help prevent overfitting by randomly dropping neurons during training, ensuring that the model generalizes better to unseen data. I expect this to have a significant effect the models' performance, especially on the test data. The model summary can be shown in figure 2 below.



```
Start of epoch 0
train loss 272815243264.000 train mse loss 272815243264.000 validate mse loss 222886920192.000
Model: "protein_structure_predictor1"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ resnet50 (Functional)           │ (None, 8, 8, 2048)     │    23,581,440 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ up_sampling2d (UpSampling2D)    │ (32, 16, 16, 2048)     │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d (Conv2D)                 │ (32, 16, 16, 128)      │     2,359,424 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ up_sampling2d_1 (UpSampling2D)  │ (32, 32, 32, 128)      │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_1 (Conv2D)               │ (32, 32, 32, 64)       │        73,792 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ up_sampling2d_2 (UpSampling2D)  │ (32, 256, 256, 64)     │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout (Dropout)               │ ?                      │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_2 (Conv2D)               │ (32, 256, 256, 1)      │           577 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 26,015,233 (99.24 MB)
 Trainable params: 25,962,113 (99.04 MB)
 Non-trainable params: 53,120 (207.50 KB)
None
train loss 211553255424.000 train mse loss 211553255424.000 validate mse loss 222820663296.000
```

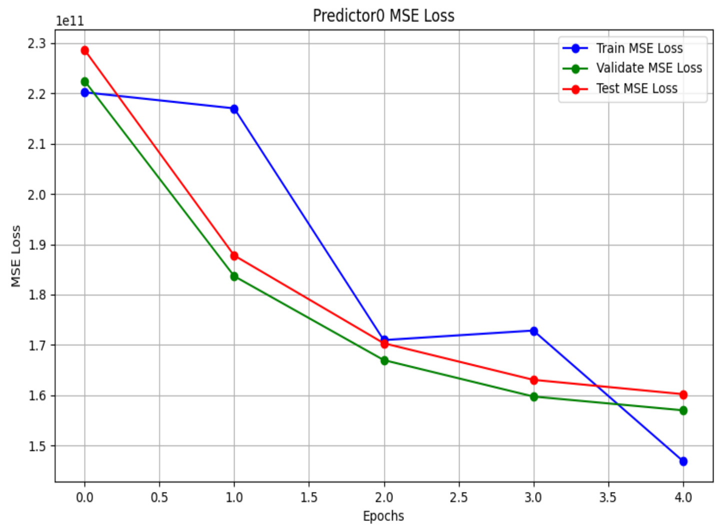


```
train loss 240869015552.000 train mse loss 240869015552.000 validate mse loss 222743150592.000
Model: "protein_structure_predictor0"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                        ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ resnet50 (Functional)               │ (None, 8, 8, 2048)     │    23,581,440 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_transpose (Conv2DTranspose)  │ (32, 16, 16, 128)      │     2,359,424 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_transpose_1 (Conv2DTranspose)│ (32, 32, 32, 64)       │        73,792 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_transpose_2 (Conv2DTranspose)│ (32, 256, 256, 32)     │        18,464 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d (Conv2D)                     │ (32, 256, 256, 1)      │           289 │
└─────────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 26,033,409 (99.31 MB)
 Trainable params: 25,980,289 (99.11 MB)
 Non-trainable params: 53,120 (207.50 KB)
None
train loss 209181048832.000 train mse loss 209181048832.000 validate mse loss 222743117824.000
```

Figure 1& 1: ProteinStructurePredictors model summary

Results

The results of the initial model, ProteinStructurePredictor0, is a final test mse loss of 157,040,312,320. This can be shown in figure 3. This loss is extremely high, however the improvements made to the model reduced the test mse loss down to 39,560,581,120 in ProteinStructurePredictor1. This is a significant difference in the results, the loss during training and initial predictions for the ProteinStructurePredictor0 model can be shown in figures 3 & 4 while the improved model's loss over training and final prediction is shown in figures 5 & 6.



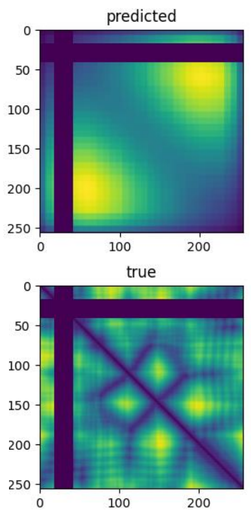
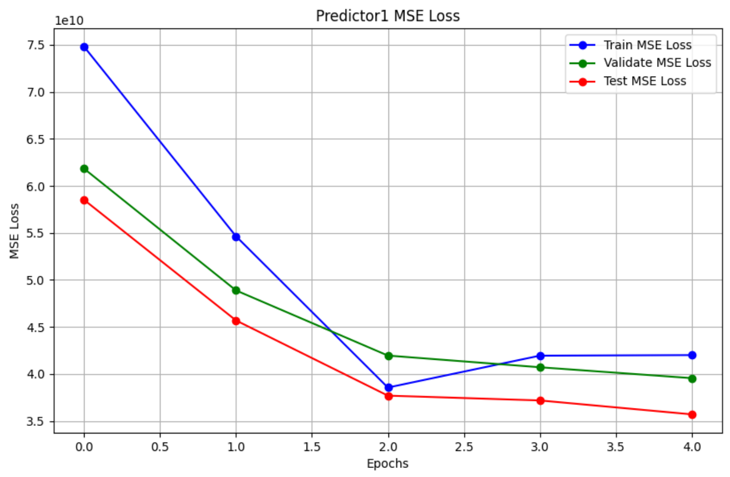


Figure 3 & 4: ProteinStructurePredictor0 Training and Prediction



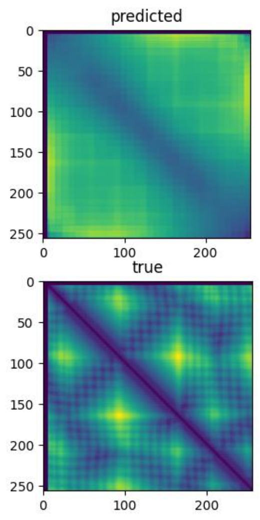


Figure 5 & 6: ProteinStructurePredictor1 Training and Prediction

Conclusion

In conclusion the results of training both models shows that my hypothesis was correct with the difference of final test MSE loss being a 75% decrease from 157 billion to 39 billion. For others working on similar tasks, these results show the importance of choosing the right up sampling technique depending on the tasks complexity. In the future, a comparison between the use of different pre-trained networks would be beneficial as well as including the use of Attention layers in the network to retain important relationships. An evaluation of different loss techniques could also be beneficial to the model's performance. Screenshots of the code running can be seen in figures 1 & 2.