RSNA Bone Age Challenge

Data and Preprocessing

- The provided data was split into twenty groups based on sex and bone age score. 400 samples from each group were taken, resulting in 8000 data, further divided into 6000 training data and 2000 validation data.

- Input pixel dimensions of 500 × 500 was selected as an optimal size for the problem

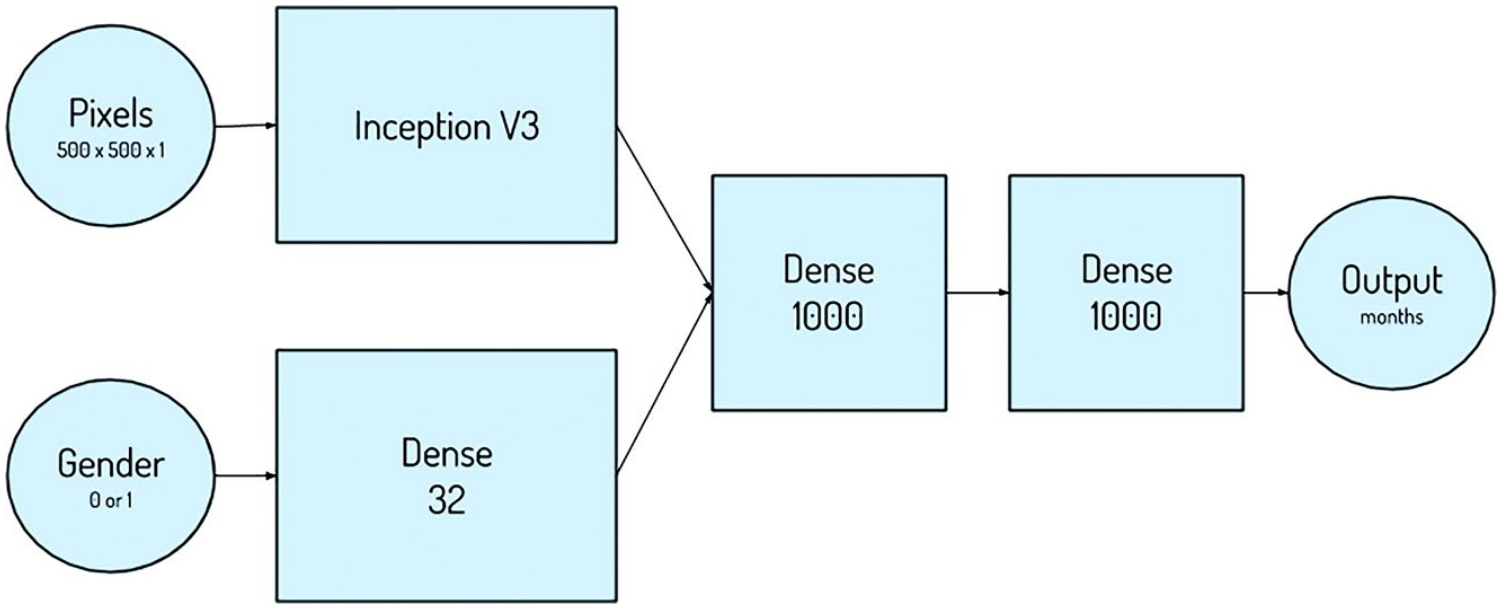
Transformer

```
from torchvision.transforms import v2
preprocess = v2.Compose([

    v2.RandomHorizontalFlip(0.5),
    v2.Resize(500),
    v2.CenterCrop(500),
    v2.Compose([v2.ToImage(), v2.ToDtype(torch.float32, scale=True)]),    #ToTensor()
    v2.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

Model Architecture

- My model Architecture was inspired by the RSNA Bone Challenge winner 2017.



Inception v3 model modification

```
├─AdaptiveAvgPool2d (avgpool)    [1, 2048, 14, 14]    [1, 2048, 1, 1]
├─Dropout (dropout)              [1, 2048, 1, 1]      [1, 2048, 1, 1]
├─Linear (fc)                   [1, 2048]            [1, 1000]
```

```
├─MaxPool2d (avgpool)            [1, 2048, 14, 14]    [1, 2048, 7, 7]
├─Flatten (dropout)              [1, 2048, 7, 7]      [1, 100352]
├─Identity (fc)                  [1, 100352]          [1, 100352]
```

Model Architecture

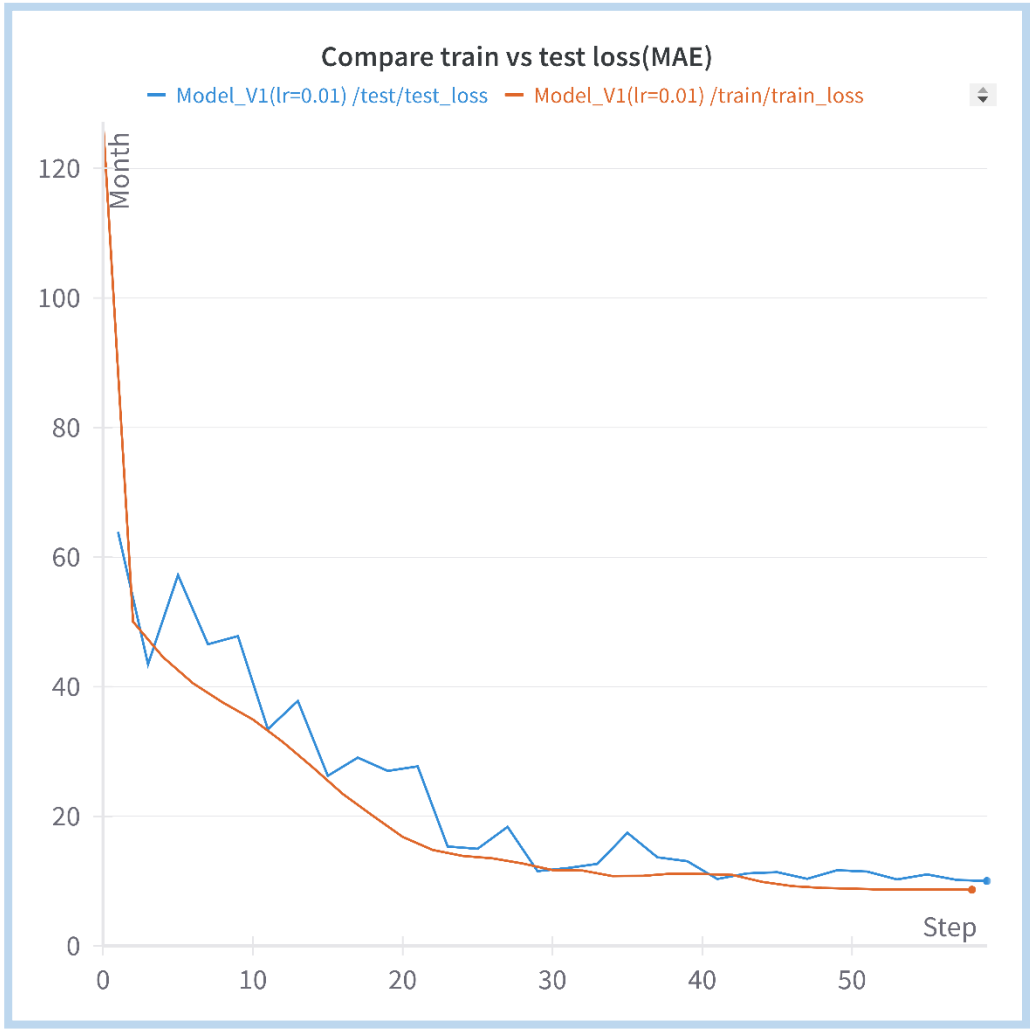
- Both gender and pixel information were incorporated into a single network using the inception v3 architecture for the pixel information. The layer after the final concatenation layer from the Inception V3 network was extracted, pushed through a two-dimensional polling layer(kernel=(2,2), stride =2) then flattened, and concatenated with the gender network which took as input binary gender information (0 for female or 1 for male). Before concatenation, the gender input was fed through a 32-neuron densely connected layer. Following concatenation, two additional 1000-neuron densely connected layers with ‘Relu’ activation were used before the final single-output layer.

Training

- The model was trained with a minibatch size of 32 with the ADAM optimizer attempting to minimize the mean absolute error of the output.
- The best model(model_v2) achieved a MAD of 5.5949 Months on the validation set

Model_V1(Lr =0.01)

Project Link



```
  1 Epoch: 0
  2 -------
  3 Train loss: 127.16167 |
  4  Test loss: 63.89064
  5 Epoch: 1
  6 -------
  7 Train loss: 50.03253 |
  8  Test loss: 43.46796
  9 Epoch: 2
 10 -------
 11 Train loss: 44.57225 |
 12  Test loss: 57.23732
 13 Epoch: 3
 14 -------
 15 Train loss: 40.53738 |
 16  Test loss: 46.56078
```

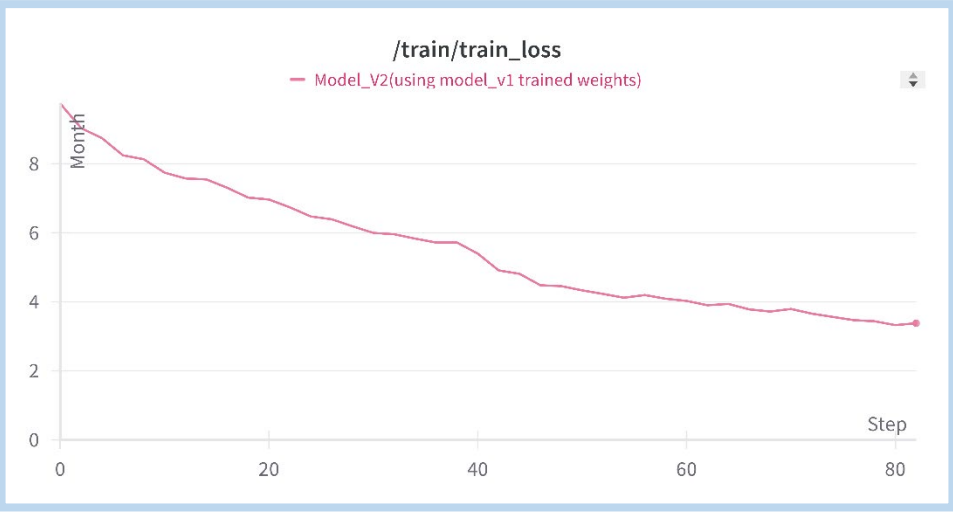
```
105 Epoch: 26
106 -------
107 Train loss: 8.69899 |
108  Test loss: 10.25974
109 Epoch: 27
110 -------
111 Train loss: 8.70995 |
112  Test loss: 11.03230
113 Epoch: 28
114 -------
115 Train loss: 8.73719 |
116  Test loss: 10.17792
117 Epoch: 29
118 -------
119 Train loss: 8.66754 |
120  Test loss: 10.02208
```

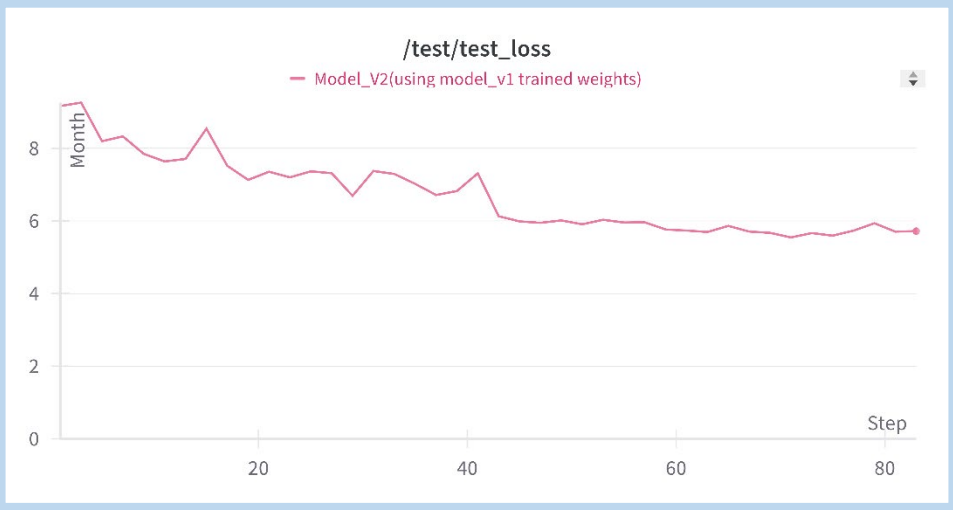
Model_V2(lower Model_V1 loss even more.)

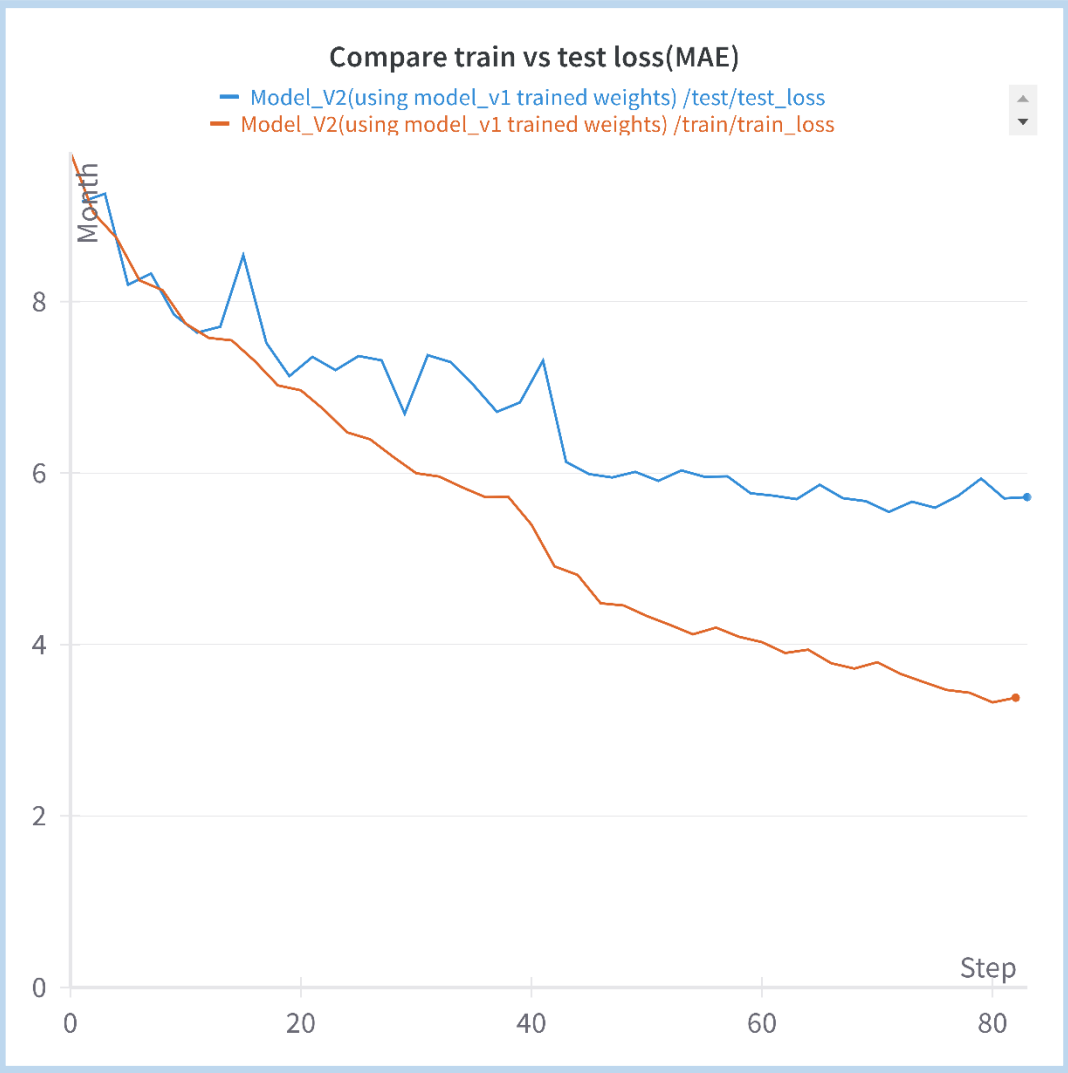
- To do so I create Model_V2.
- first, I feed Model_V1 weights into the model.
- then I create an Lr scheduler and train the Model_v2 for 40 more epochs.

Model_V2

Project Link







```
5 Epoch: 0

6 -------

7 Train loss(MAE): 9.03659 |

8  Test loss(MAE): 9.25735

9 Epoch: 1

10 -------

11 Train loss(MAE): 8.74311 |

12  Test loss(MAE): 8.19616

13 Epoch: 2

14 -------

15 Train loss(MAE): 8.24719 |

16  Test loss(MAE): 8.32757

17 Epoch: 3

18 -------

19 Train loss(MAE): 8.13266 |

20  Test loss(MAE): 7.84683
```

```
77 Epoch: 18

78 -------

79 Train loss(MAE): 5.72028 |

80  Test loss(MAE): 6.82344

81 Epoch: 19

82 -------

83 Epoch 00021: reducing learning
rate of group 0 to 5.0000e-04.

84 Train loss(MAE): 5.39745 |

85  Test loss(MAE): 7.31157

86 Epoch: 20

87 -------

88 Train loss(MAE): 4.91108 |

89  Test loss(MAE): 6.12954

90 Epoch: 21

91 -------

92 Train loss(MAE): 4.81145 |

93  Test loss(MAE): 5.98737
```

```
150 Epoch: 36

151 -------

152 Train loss(MAE): 3.56267 |

153  Test loss(MAE): 5.59526

154 Epoch: 37

155 -------

156 Train loss(MAE): 3.47020 |

157  Test loss(MAE): 5.73341

158 Epoch: 38

159 -------

160 Train loss(MAE): 3.43710 |

161  Test loss(MAE): 5.93597

162 Epoch: 39

163 -------

164 Train loss(MAE): 3.32468 |

165  Test loss(MAE): 5.70421
```

Challenges and Failures

- Throughout my journey, I've experimented with various models and architectures, encountering numerous challenges and failures along the way. I'd like to share a minor setback that I inadvertently recorded, which I find quite interesting. When I first initialized my model_v1, I used a learning rate of 0.1, causing my model to fail to learn due to its high value (figures on the next slide). While it may not be incredibly intriguing, I still wanted to share it since I happened to record it ☺.

Model_V1(Lr=0.1)

Project Link