Project 2

CSCE 790 – Edge and Neuromorphic Computing

Mark Shperkin

Overview:

In this project, the goal was to optimize a neural network architecture for face bounding box prediction using a given facial image dataset. The primary objective was to design a lightweight and efficient model capable of performing inference on resource-constrained hardware, NVIDIA Jetson Nano. Additionally, a competition through a Kaggle evaluated the architecture based on their prediction accuracy and inference latency. Accuracy was quantified using Intersection over Union (IoU), a metric that assesses the degree of overlap between the model's predicted bounding boxes and the ground-truth coordinates. Latency evaluation was performed directly on the NVIDIA Jetson Nano to accurately reflect performance in a realistic deployment scenario.

Solution:

In order to identify the optimal neural network architecture and corresponding hyperparameters, I conducted a structured neural architecture and hyperparameter search.

- The neural architecture search space included 21 different base CNN feature extractors:

alexnet	resnet18	densenet121	mobilenet_v2	mnasnet0_5	regnet_y_400mf	regnet_x_800mf
vgg11	squeezenet1_0	shufflenet_v2_x0_5	mobilenet_v3_small	mnasnet1_0	regnet_y_800mf	convnext_tiny
vgg11_bn	squeezenet1_1	shufflenet_v2_x1_0	mobilenet_v3_large	mnasnet1_3	regnet_x_400mf	convnext_small

- The hyperparameter research considered learning rates ranging from 1e-2 to 1e-5.

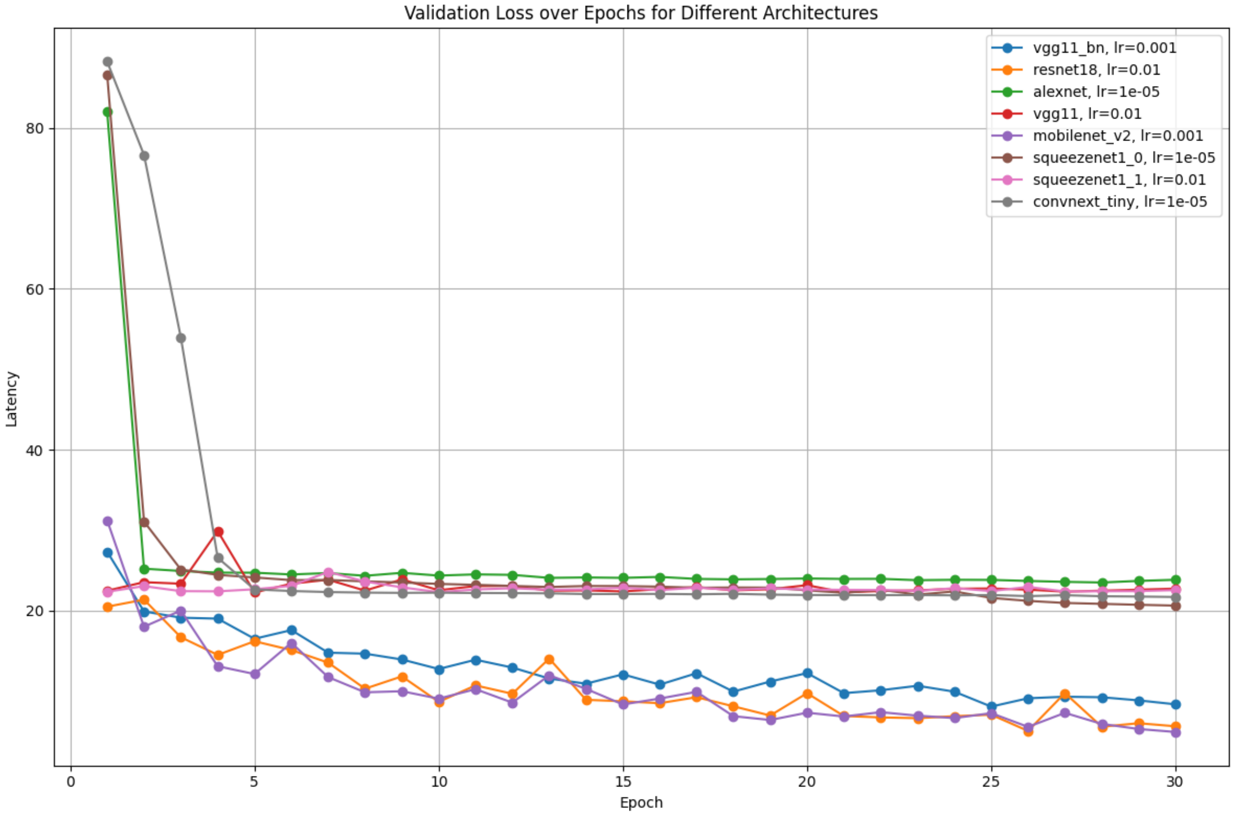
The network architecture employed a regression head consisting of two layers:

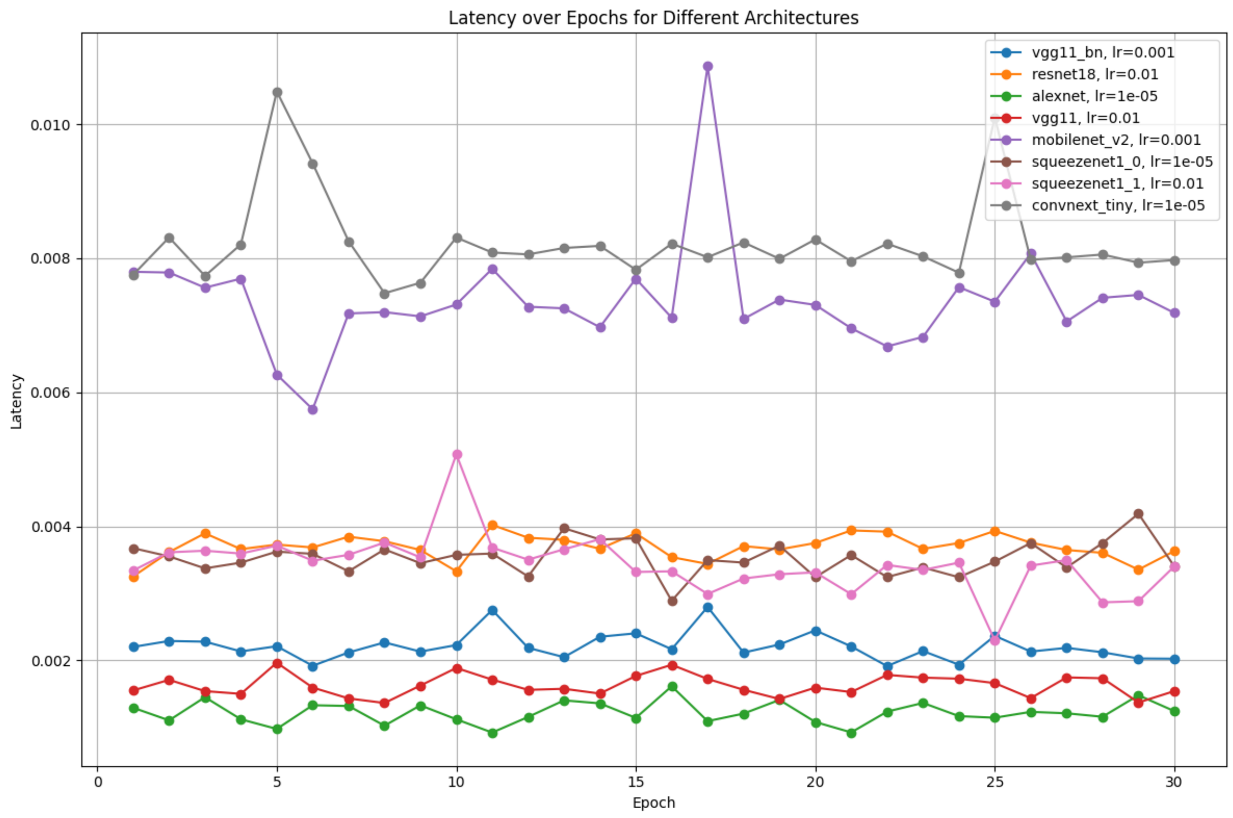
- The input layer received neurons from the selected base architecture and reduced the dimensionality by half
- The output layer produced four values representing bounding box coordinates.

The search procedure was structured in three distinct stages.

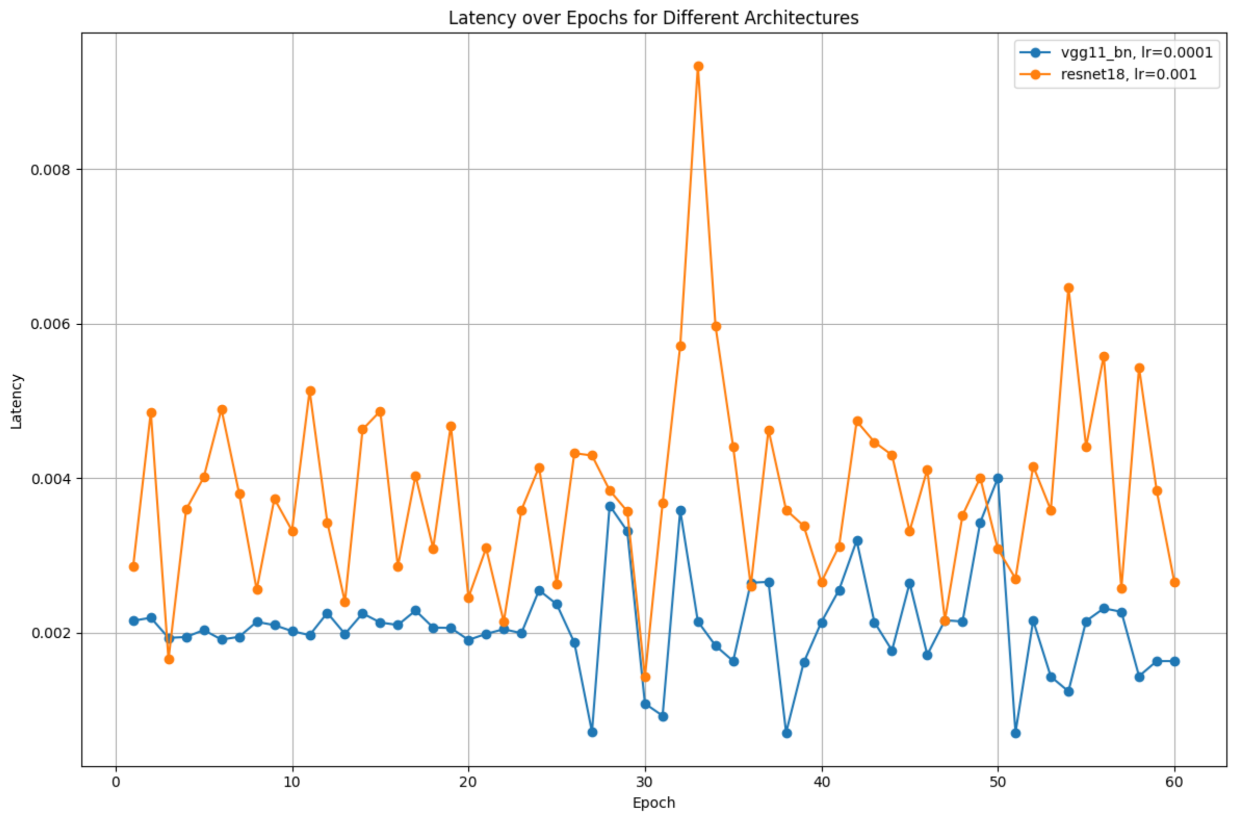
- In the first stage, each combination of architecture and learning rate was trained and evaluated for 10 epochs, after which I selected the top 8 architectures based on their validation loss and inference latency. (there where 84 different configurations and I was not able to plot the results effectively)

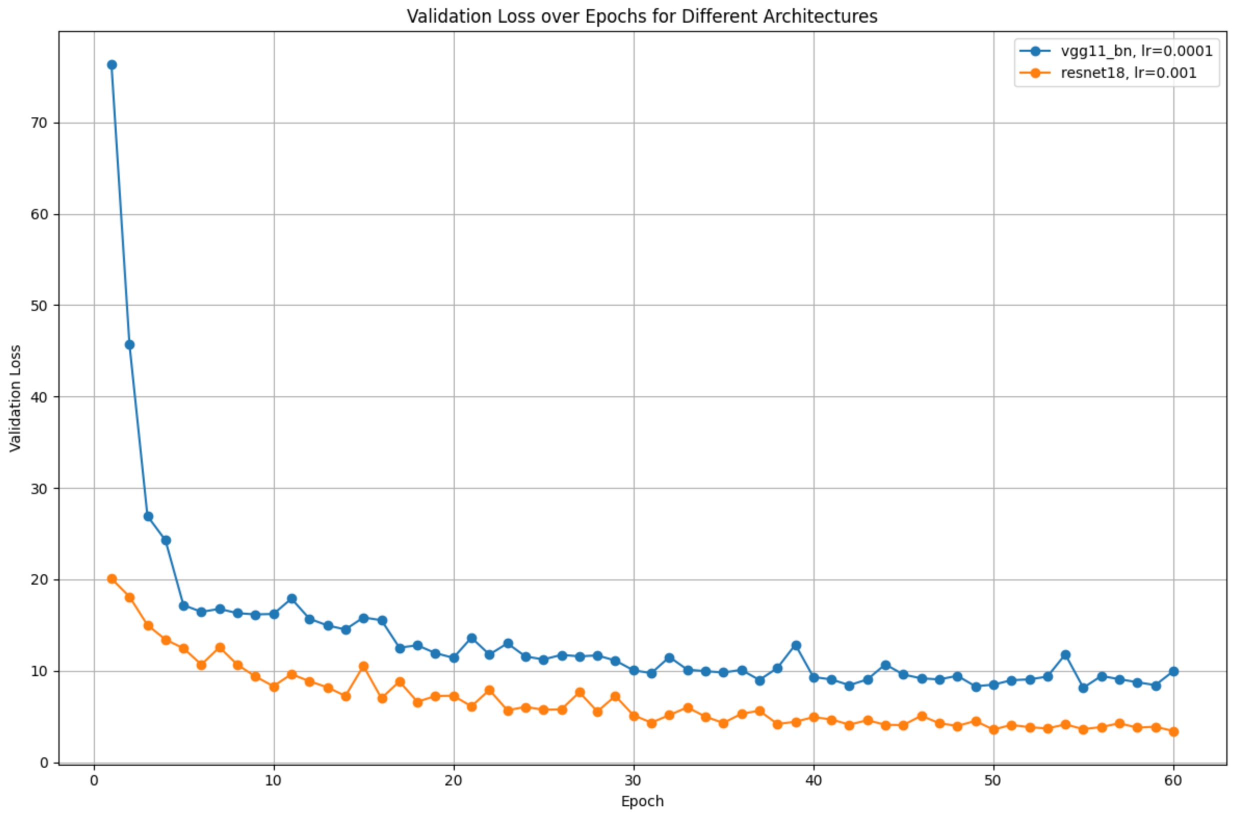
- In the second stage, these 8 selected architectures were further trained for 30 epochs using their optimal learning rates identified from the initial stage. This stage narrowed the selection down to the two best-performing architectures.





Finally, the third stage involved extensive training of these two architectures over 60 epochs. This allowed identification of the optimal architecture and learning rate for the bounding box prediction task.


Latency over Epochs for Different Architectures
vgg11_bn, lr=0.0001
resnet18, lr=0.001
Latency
0.008
0.006
0.004
0.002
0
10
20
30
40
50
60
Epoch


Validation Loss over Epochs for Different Architectures
vgg11_bn, lr=0.0001
resnet18, lr=0.001
Validation Loss
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
Epoch

The search concluded that resnet18 architecture with a learning rate of 0.01 was the optimal architecture for this problem, achieving an inference latency of 0.013 seconds and an IoU accuracy of 40%.

Challenges & Future Work

I faced a few challenges during this project, especially when setting up the architecture search. Figuring out how many output neurons each network had was a bit tricky and took more effort than expected. Also, the first search I ran took about five and a half hours for 30 epochs, but afterward, I realized I had made some calculation mistakes. Unfortunately, this meant I had to redo everything, which was quite frustrating.

Conclusion

Despite these issues, this project provided me with invaluable experience, not only in understanding neural network architectures and hyperparameter search but also in effectively deploying models on small, resource-limited devices like the NVIDIA Jetson Nano. In future work, I plan to explore additional optimizations, possibly using techniques such as pruning and quantization, to further improve the accuracy and latency of models designed specifically for edge devices.