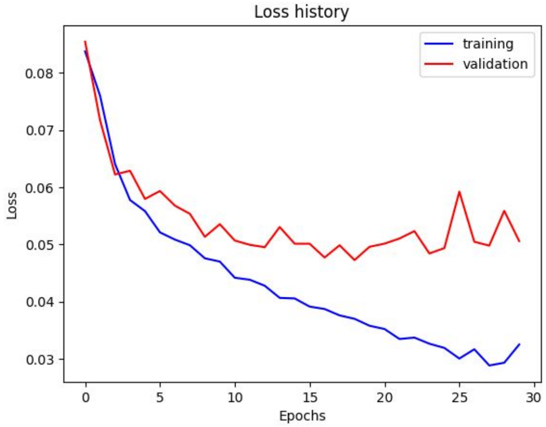
CS 6476 Project 4

Mengying Lin
mlin365@gatech.edu
mlin365
904016037

Part 1: SimpleNet

[Insert loss plot for SimpleNet here]

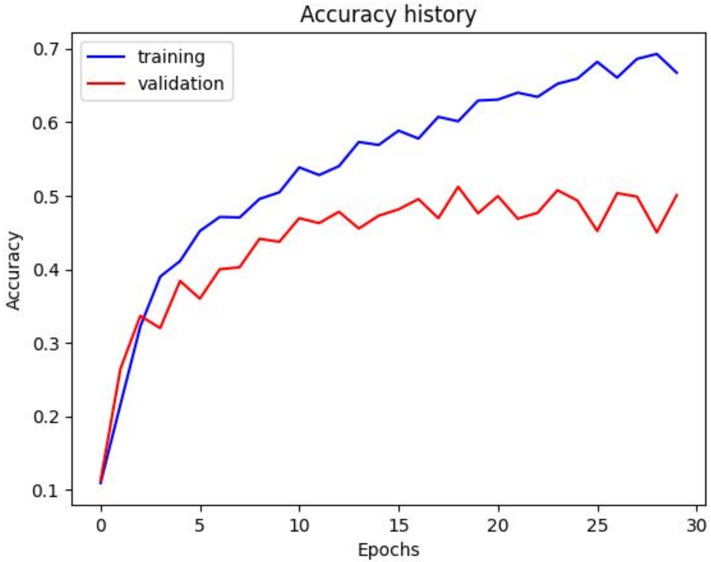


Final training accuracy:

Final validation accuracy:

```
Train Accuracy = 0.6670016750618442; Validation
Accuracy = 0.5006666668256123
```

[Insert accuracy plot for SimpleNet here]



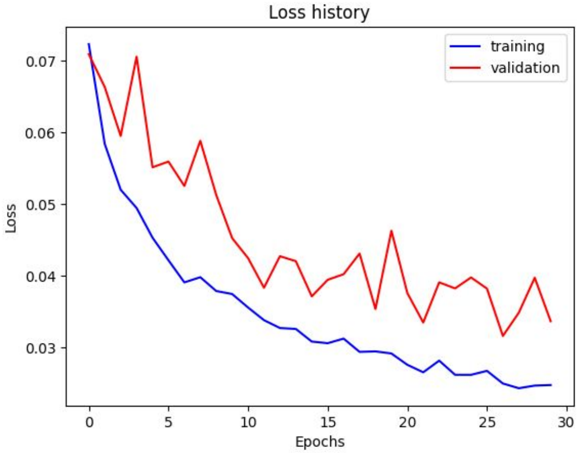
Part 2: SimpleNetFinal

Add each of the following (keeping the changes as you move to the next row):

	Training accuracy	Validation accuracy
SimpleNet	0.67	0.50
+ Jittering	0.65	0.55
+ Zero-centering & variance-normalization	0.72	0.65
+ Dropout regularization	0.70	0.72
+ Making network "deep"	0.83	0.81
+ Batch normalization	0.89	0.90

Part 2: SimpleNetFinal

[Insert loss plot for SimpleNetFinal here]



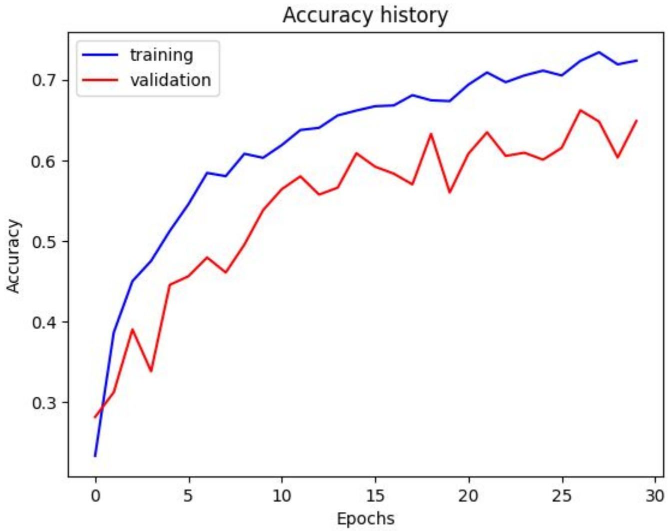
Final training accuracy:

Final validation accuracy:

Train Accuracy = 0.7236180904921974;

Validation Accuracy = 0.648666666507721

[Insert accuracy plot for SimpleNetFinal here]



Part 2: SimpleNetFinal

[Name 10 different possible transformations for data augmentation.]
Horizontal Flip, Vertical Flip, Rotation, Scaling, Translation, Random Crop, Color Jitter, Gaussian Noise, Brightness Adjustment, Shear Transformation

[What is the desired variance after each layer? Why would that be helpful?]
The desired variance after each layer in a neural network is 1. This is helpful because it ensures that the inputs to each layer have a consistent scale, which prevents exploding or vanishing gradients and stabilizes learning across layers.

Part 2: SimpleNetFinal

[What distribution is dropout usually sampled from?]

Bernoulli distribution

[How many parameters does your base SimpleNet model have? How many parameters does your SimpleNetFinal model have?]

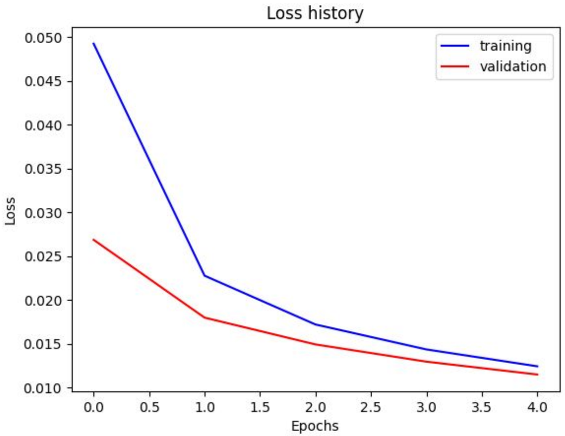
Simplenet: 56,895
SimpleNetFinal: 126,383

[What is the effect of batch norm after a conv layer with a bias compared to having no batch norm?]

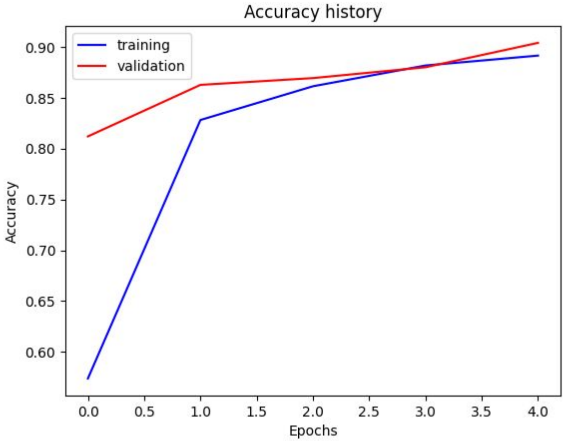
With Batch Norm: The effect of the bias in the convolutional layer is negated because Batch Norm subtracts the mean and normalizes the activations, making the bias unnecessary.
Without Batch Norm: The bias is used to shift the activations before they are passed to the next layer.

Part 3: ResNet

[Insert loss plot here]



[Insert accuracy plot here]



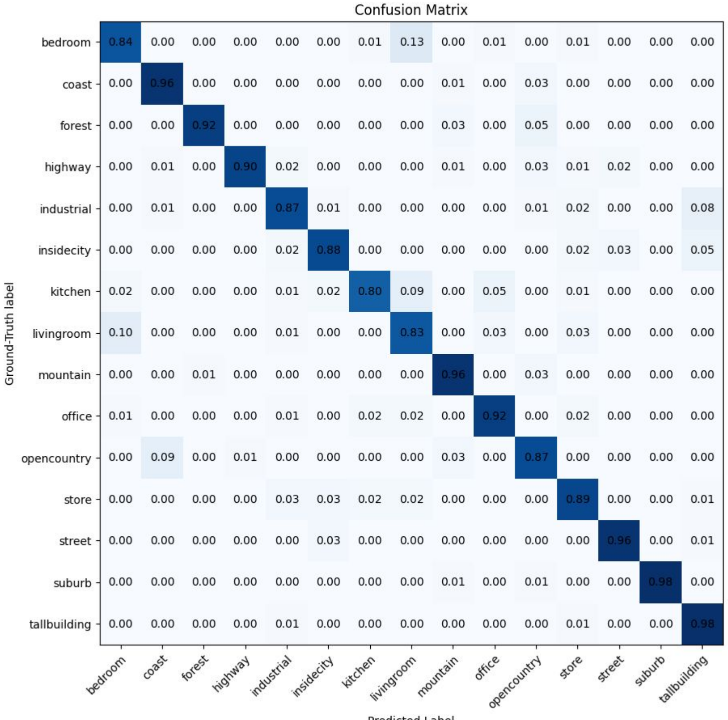
Final training accuracy:

Final validation accuracy:

Train Accuracy = 0.8914572864321608; Validation Accuracy = 0.9040000003178914

Part 3: ResNet

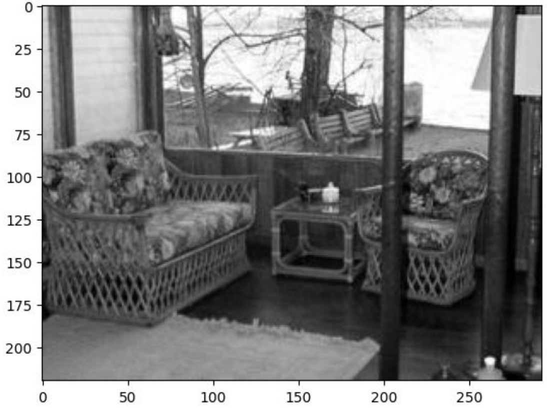
[Insert visualization of confusion matrix obtained from your final ResNet model.]

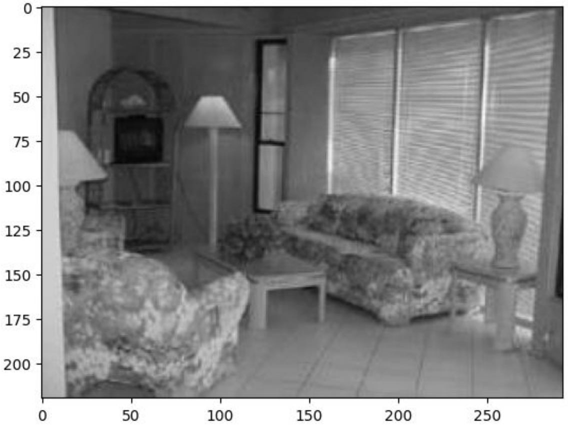


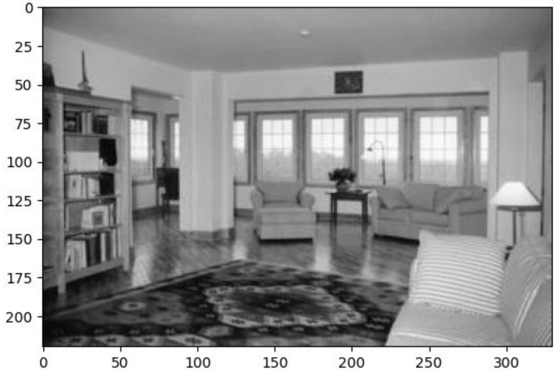
	bedroom	coast	forest	highway	industrial	insidecity	kitchen	livingroom	mountain	office	opencountry	store	street	suburb	tallbuilding
bedroom	0.84	0.00	0.00	0.00	0.00	0.00	0.01	0.13	0.00	0.01	0.00	0.01	0.00	0.00	0.00
coast	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.03	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.92	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.05	0.00	0.00	0.00	0.00
highway	0.00	0.01	0.00	0.90	0.02	0.00	0.00	0.00	0.01	0.00	0.03	0.01	0.02	0.00	0.00
industrial	0.00	0.01	0.00	0.00	0.87	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.08
insidecity	0.00	0.00	0.00	0.00	0.02	0.88	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.05
kitchen	0.02	0.00	0.00	0.00	0.01	0.02	0.80	0.09	0.00	0.05	0.00	0.01	0.00	0.00	0.00
livingroom	0.10	0.00	0.00	0.00	0.01	0.00	0.00	0.83	0.00	0.03	0.00	0.03	0.00	0.00	0.00
mountain	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.03	0.00	0.00	0.00	0.00
office	0.01	0.00	0.00	0.00	0.01	0.00	0.02	0.02	0.00	0.92	0.00	0.02	0.00	0.00	0.00
opencountry	0.00	0.09	0.00	0.01	0.00	0.00	0.00	0.00	0.03	0.00	0.87	0.00	0.00	0.00	0.00
store	0.00	0.00	0.00	0.00	0.03	0.03	0.02	0.02	0.00	0.00	0.00	0.89	0.00	0.00	0.01
street	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.01
suburb	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.98	0.00
tallbuilding	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.98

Part 3: ResNet

[Insert visualizations of 3 misclassified images from the most misclassified class according to your confusion matrix. Explain why this may have occurred.]







All of the above are labeled as ‘bedroom,’ while their ground truth (GT) labels are ‘living room.’ This might be because they all have a sofa in the scene, which looks similar to a bed, leading to confusion.

Part 3: ResNet

[What does fine-tuning a network mean?]

It means adjust the model architecture (e.g. adding more layers, adding normalizations, adding dropout, adjusting the number of neurons within layer) to make the performance better.

[Why do we want to "freeze" the conv layers and some of the linear layers from a pre-trained ResNet? Why can we do this?]

The pretrained model serves as a feature extractor, and we are typically unwilling to let the initially “bad” gradients to affect the well-established feature extractor. We can do this because the gradients won’t flow back to those layers if their parameters are detached from the computational graph.

Part 4: Logging to Weights and Biases

[Add the link to the weights and Biases project below. Make sure your project is **public.**]

View run **resnet** at: https://wandb.ai/cassielin0910/cv-assn4/runs/ltqvob9k
View project at: https://wandb.ai/cassielin0910/cv-assn4

Extra credit

[Add the link to the Weights & Biases run for the model used in the “huggingface.py” script below]

View run of Mobilevit-small (worse than ResNet) at https://wandb.ai/cassielin0910/cv-assn4/runs/8q3tuasx
View run of ConvNext (better than ResNet) at https://wandb.ai/cassielin0910/cv-assn4/runs/66jkfvbi