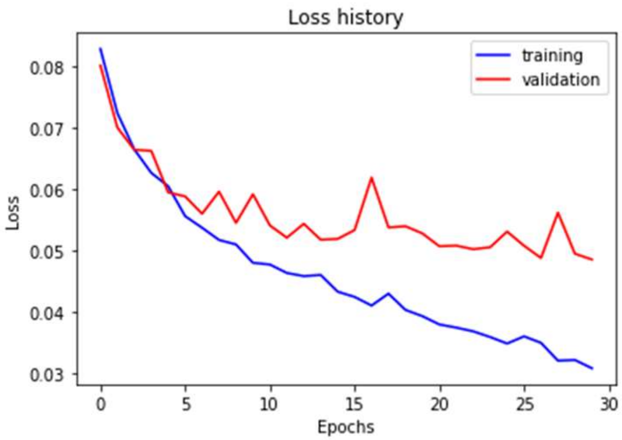
CS 4476/6476 Project 4

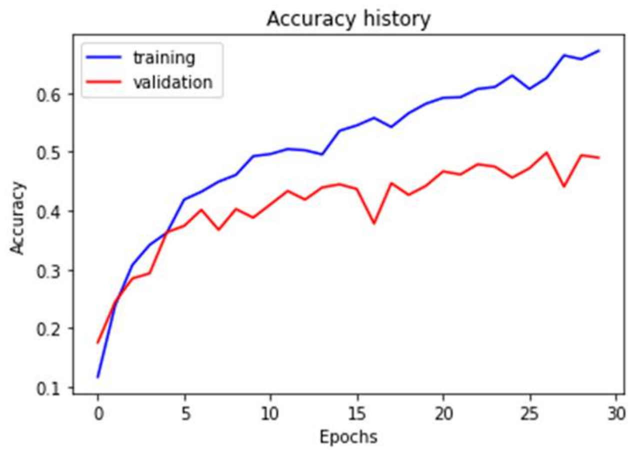
Cameron Potter
cpotter8@gatech.edu
cpotter8
903465425

Part 1: SimpleNet

[Insert loss plot for SimpleNet here]



[Insert accuracy plot for SimpleNet here]



Final training accuracy: 0.671356783919598

Final validation accuracy: 0.49

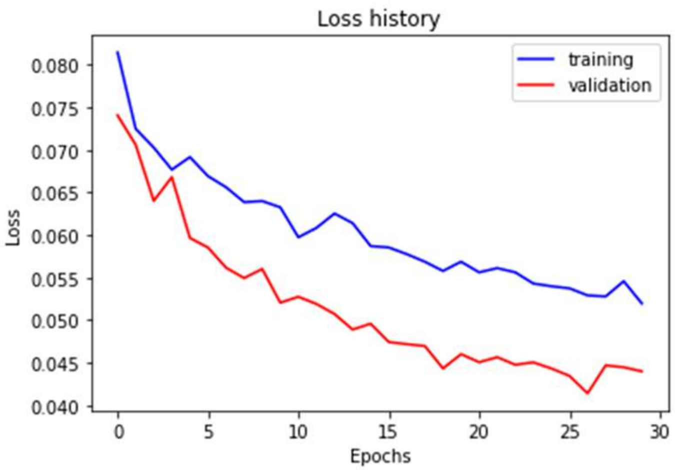
Part 2: SimpleNetFinal

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

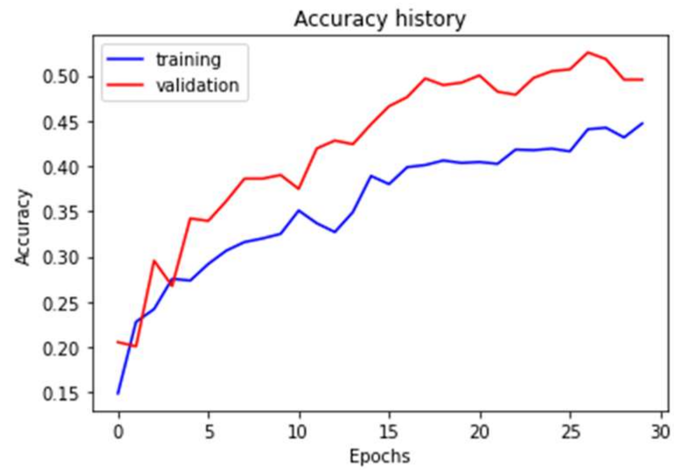
	Training accuracy	Validation accuracy
SimpleNet	0.6713567839	0.49
+ Jittering	0.5189	0.4573
+ Zero-centering & variance-normalization	0.8412060301	0.3893333333
+ Dropout regularization	0.5262981574	0.2373333333
+ Making network "deep"	0.4375209380	0.29
+ Batch normalization	0.5031825795	0.5806666666

Part 2: SimpleNetFinal

[Insert loss plot for SimpleNetFinal here]



[Insert accuracy plot for SimpleNetFinal here]



Final training accuracy: 0.5031825795644891

Final validation accuracy: 0.5806666666666667

Part 2: SimpleNetFinal

[Name 10 different possible transformations for data augmentation.]

- Noise
- Cropping
- Flipping
- Scaling
- Brightness
- Rotation
- Resizing
- Jittering
- Normalization
- Translation

[What is the desired variance after each layer? Why would that be helpful?]

We would ideally want unit variance for each layer. This is because we want to transform each input to have unit variance, and so the layers should also have unit variance.

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?]

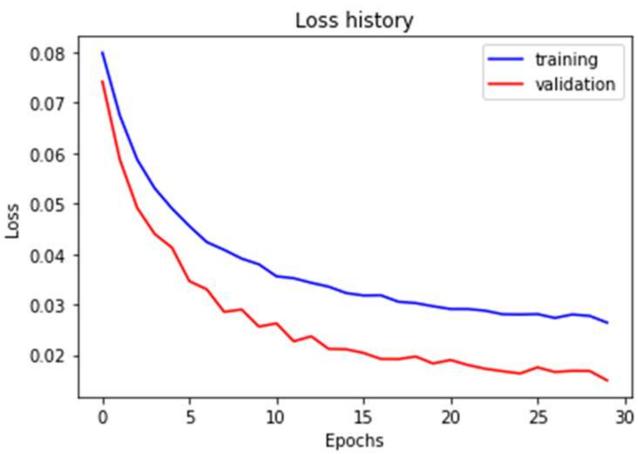
SimpleNet: 56895
SimpleNetFinal: 987698

[What is the effect of batch norm after a conv layer with a bias?]

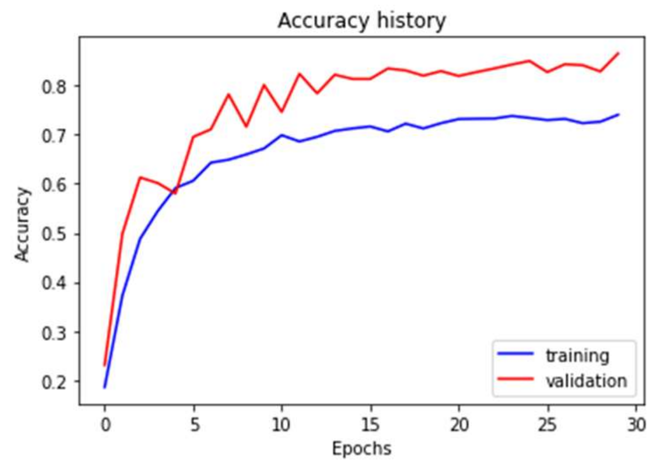
We shift the activation by the mean values, which has the effect of eliminating constants.

Part 3: ResNet

[Insert loss plot here]



[Insert accuracy plot here]

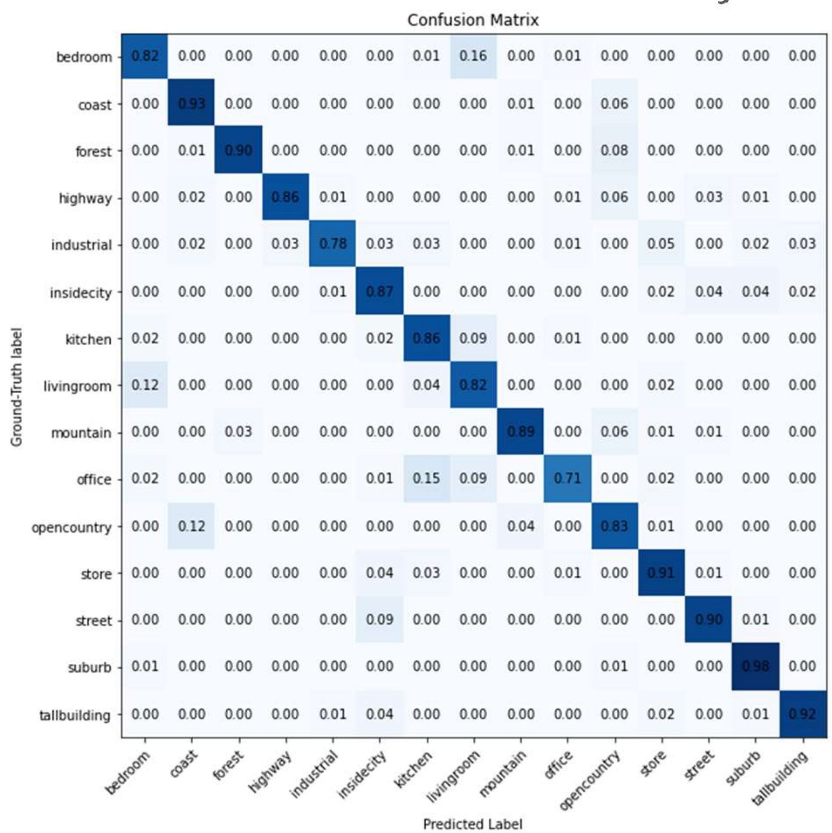


Final training accuracy: 0.7403685092127303

Final validation accuracy: 0.8653333333333333

Part 3: ResNet

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

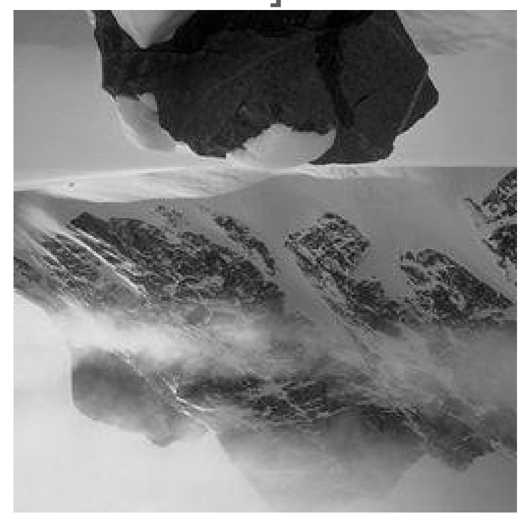


Part 3: ResNet

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

Prediction: “Mountain”, Target: “Bedroom. The mountain class most likely has lots of diagonal lines and high or low intensity regions, which this image has lots of compared to other bedroom images

Prediction: “Kitchen”, Target: “Office. The kitchen class most likely has countertops, and the copier has similar image features to a countertop



Prediction: “Bedroom”, Target: “Mountain”. The bedroom class most likely has lots of straight lines and various intensities, which this image has lots of compared to other mountain images

Part 3: ResNet

[What does fine-tuning a network mean?]

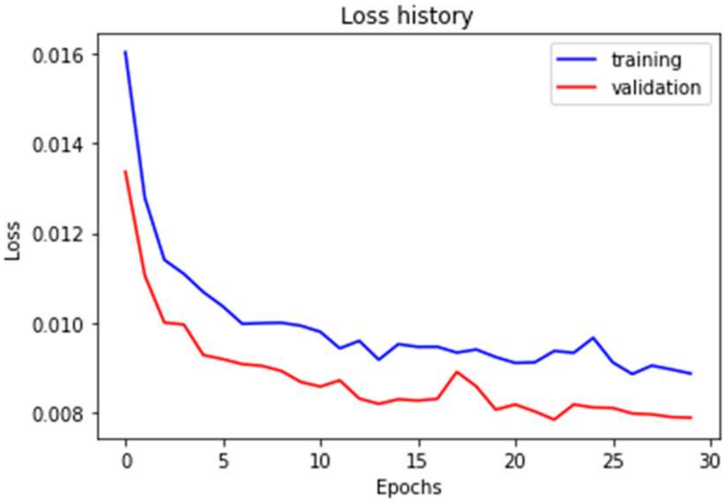
Fine-tuning a network is when we take a pre-trained model and use its weights as the initialization for a newly created model

[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?]

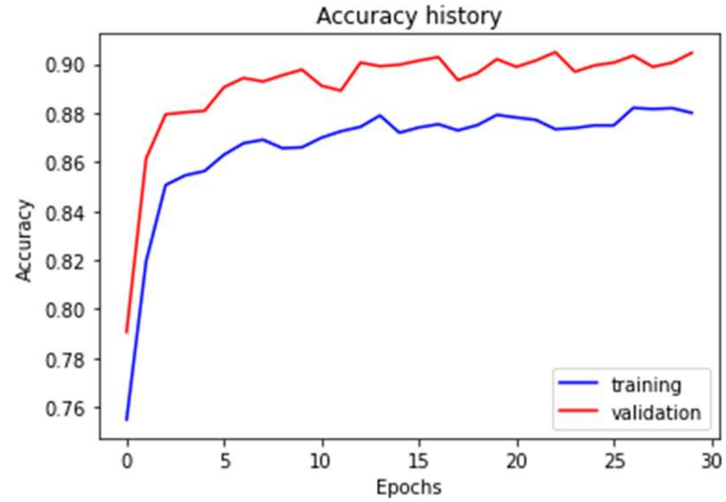
Freezing layers is making sure the gradients are not updated during training. It can help prevent overfitting and speed up computation, especially when using a pretrained model. We can do this because the weights of earlier layers we do not want to modify, as they are already trained. We just want to update how we actually classify the information.

Part 4: Multi-label Scene Attributes

[Insert loss plot here]



[Insert accuracy plot here]

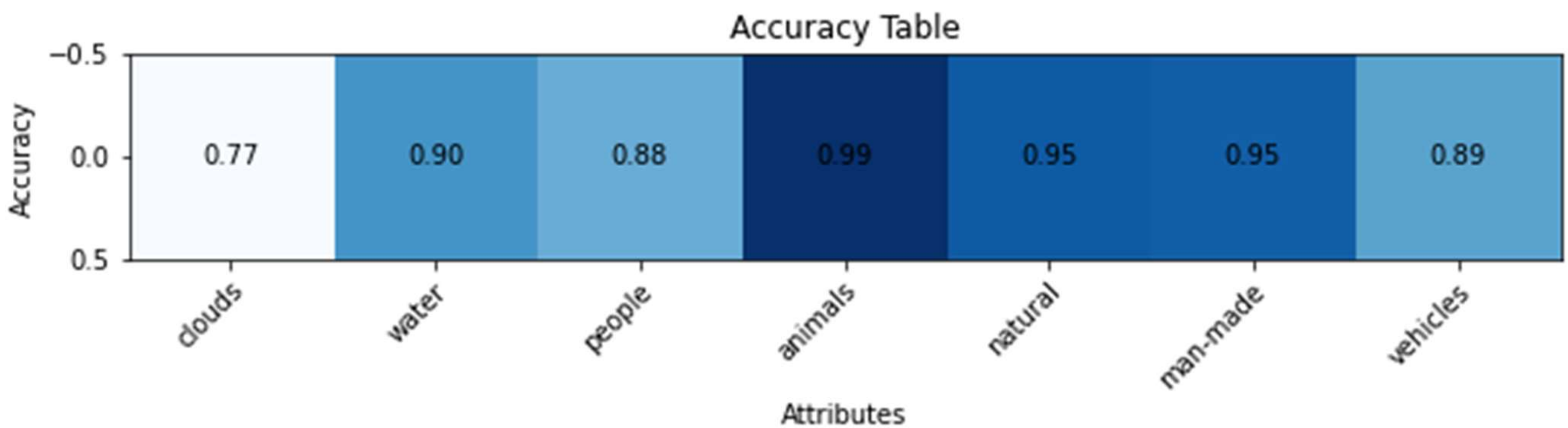


Final training accuracy: 0.8800764452938364

Final validation accuracy: 0.9045714285714286

Part 4: Multi-label Scene Attributes

[Insert visualization of accuracy table obtained from your final MultilabelResNet model.]



Part 4: Multi-label Scene Attributes

[List 3 changes that you made in the network compared to the one in part 3.]

- Changed Cross Entropy Loss to Binary Cross Entropy Loss
- Added sigmoid activation layer
- Changed output from 15 labels to 7 labels

[Is the loss function of the ResNet model from part 3 appropriate for this problem? Why or why not?]

It is not appropriate, as our output is not a probability vector. We treat each label as a binary classification problem, and thus we need to use binary cross entropy loss.

Part 4: Multi-label Scene Attributes

[Explain a problem that one needs to be wary of with multilabel classification. HINT: consider the purpose of visualizing your results with the accuracy table. You might want to do some data exploration here.]

Any problem where labels can be mutually exclusive would cause some issues with multilabel classification. For example, our labels include “man-made” and “natural”. In some cases, both these labels make sense together, such as a cabin in the woods; however, one should be wary when these labels are both true, as most of the time an image cannot be both labels.