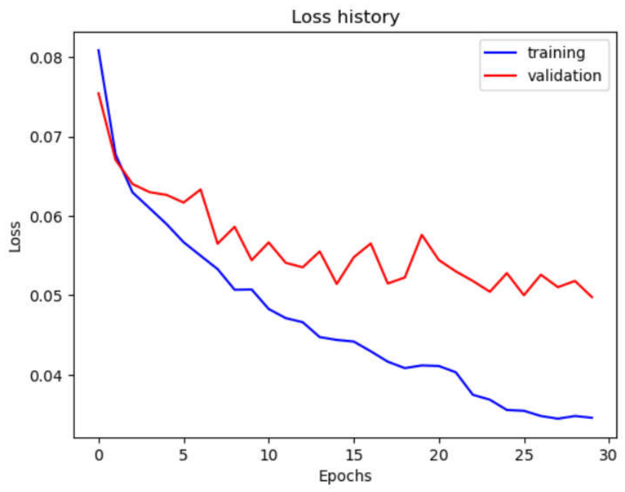
CS 4476/6476 Project 4

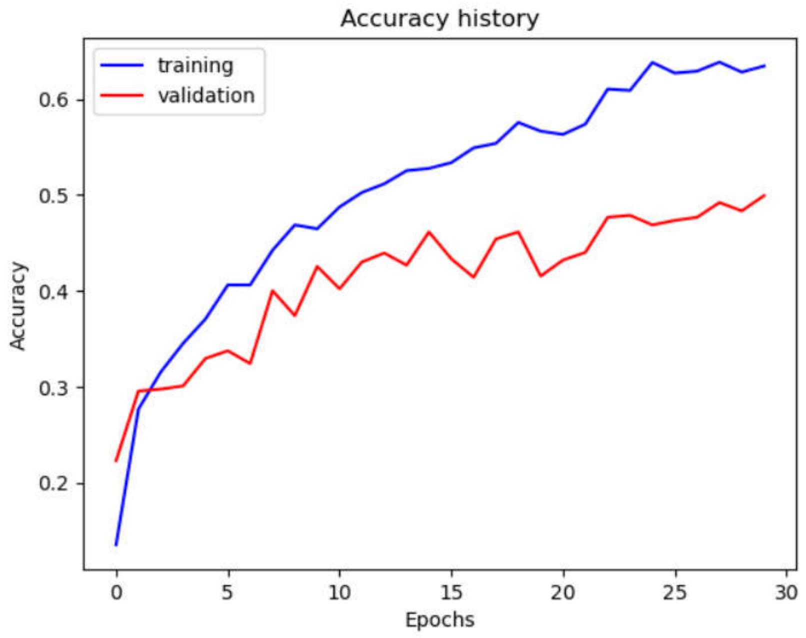
[Jiarui Xu]
[jxu605@gatech.edu]
[jxu605]
[903617486]

Part 1: SimpleNet

[Insert loss plot for SimpleNet here]



[Insert accuracy plot for SimpleNet here]



Final training accuracy:0.6345058626465662

Final validation accuracy:0.49933333333333335

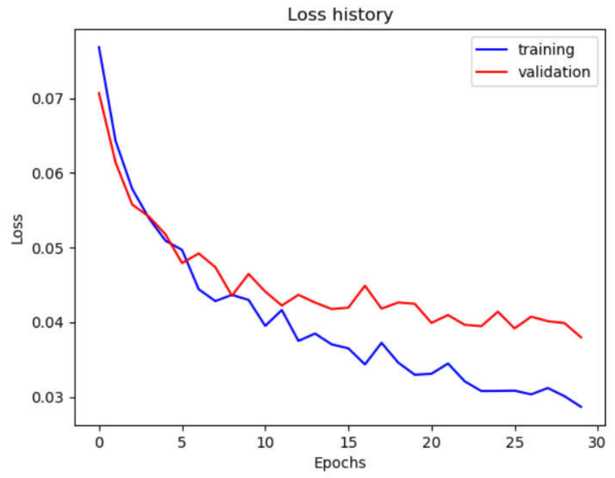
Part 2: SimpleNetFinal

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

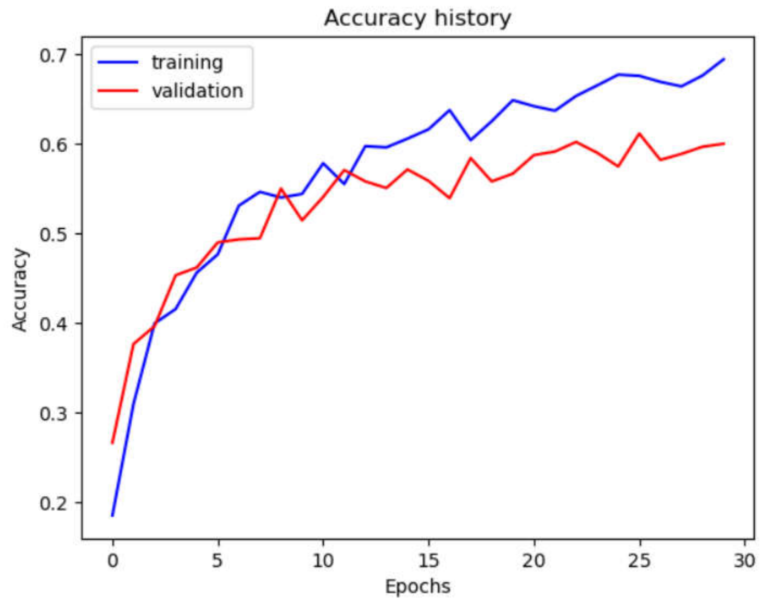
	Training accuracy	Validation accuracy
SimpleNet	0.6345058626465662	0.49933333333333335
+ Jittering	0.5820268006700167	0.4883
+ Zero-centering & variance-normalization	0.6666834170854271	0.52
+ Dropout regularization	0.626214405360134	0.5331
+ Making network "deep"	0.6435845896147404	0.5793
+ Batch normalization	0.6941373534338359	0.6

Part 2: SimpleNetFinal

[Insert loss plot for SimpleNetFinal here]



[Insert accuracy plot for SimpleNetFinal here]



Final training accuracy:0.6941373534338359

Final validation accuracy:0.6

Part 2: SimpleNetFinal

[Name 10 different possible transformations for data augmentation.]

1.Cropping
2.Flipping
3.Padding
4.Resizing
5.Change hue
6.Change saturation
7.Change contrast
8Change brightness
9.Rotation
10.Affine transformation

[What is the desired variance after each layer? Why would that be helpful?]
After normalized transformation and batch normalization, we will have the unit variance.
As for the normalized transform, it keeps the pixel value in a small range and improves the computational efficiency.
In the aspect of batch normalization, the training time and accuracy of neural network are improved, the influence of weight initialization is reduced, and the regularization effect of the network is increased.

Part 2: SimpleNetFinal

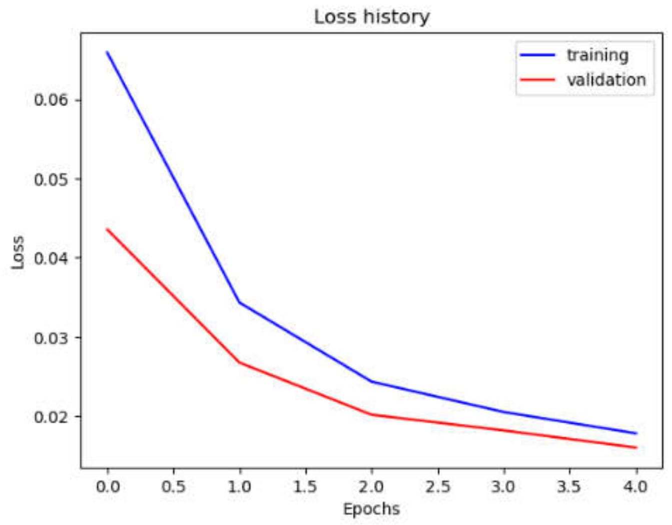
[What distribution is dropout usually sampled from?]
From a Bernoulli random variable, where to keep the variable with the probability of 1-p, and drop with the probability of p.

[How many parameters does your base SimpleNet model have? How many parameters does your SimpleNetFinal model have?]
SimpleNet model: 56895
SimpleNetFinal model: 112005

[What is the effect of batch norm after a conv layer with a bias?]
The bias will be canceled out because batch norm makes the activation change to the center by using their mean.

Part 3: ResNet

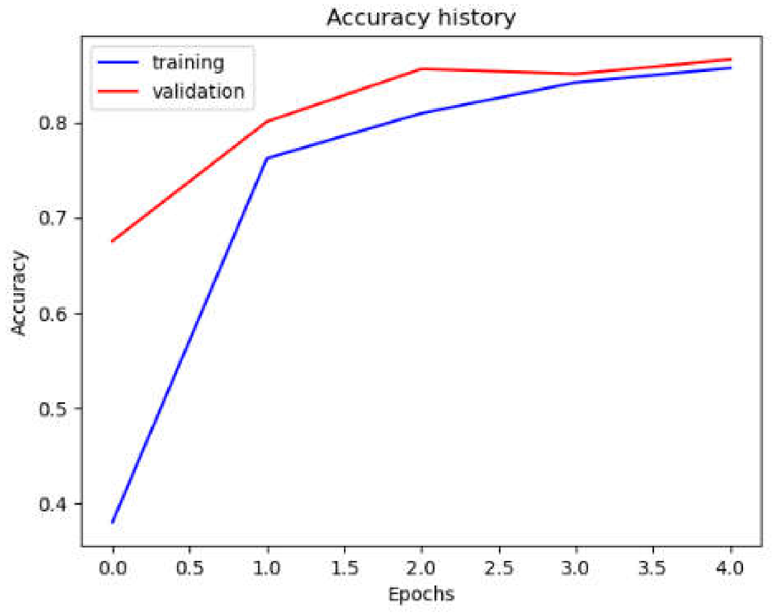
[Insert loss plot here]



Final training accuracy:0.8569514237855946

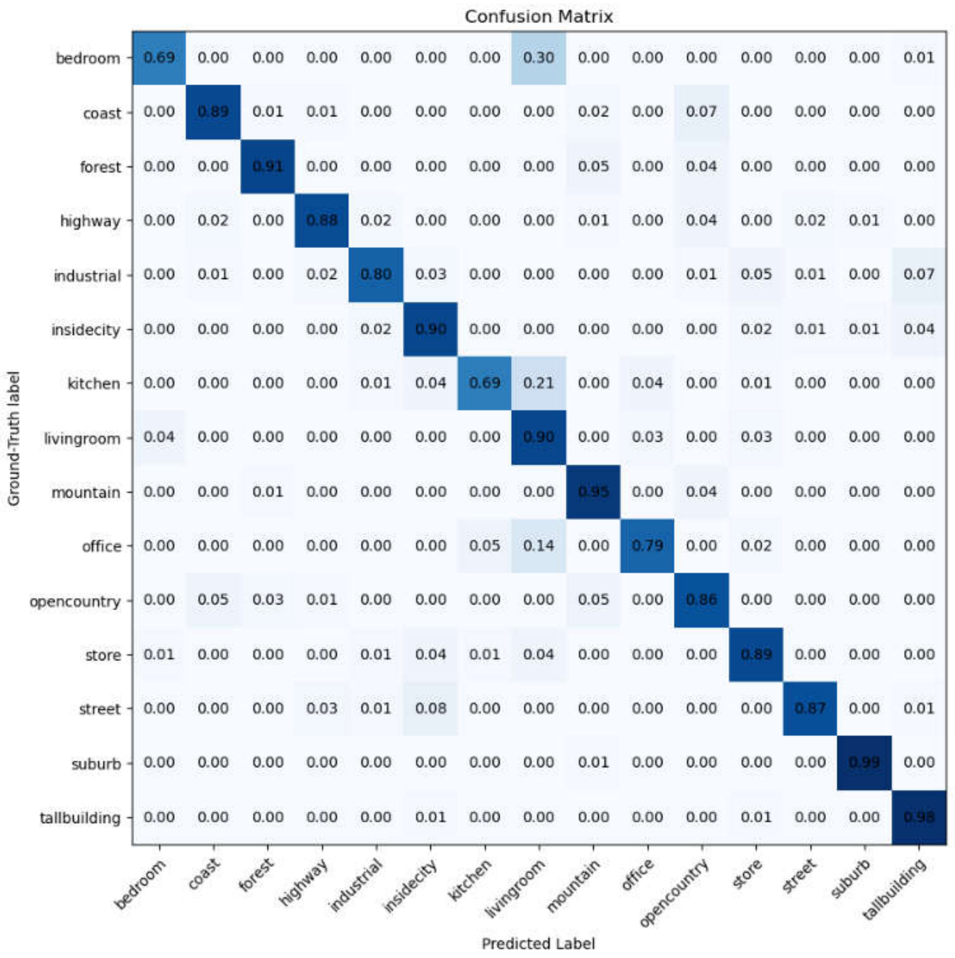
Final validation accuracy:0.866

[Insert accuracy plot here]



Part 3: ResNet

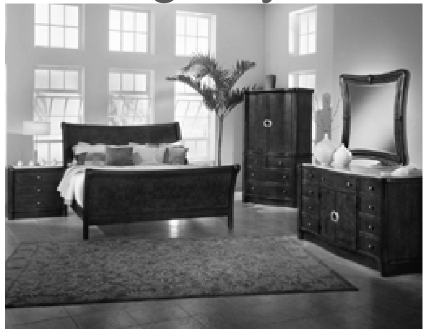
[Insert visualization of confusion matrix obtained from your final ResNet model.]

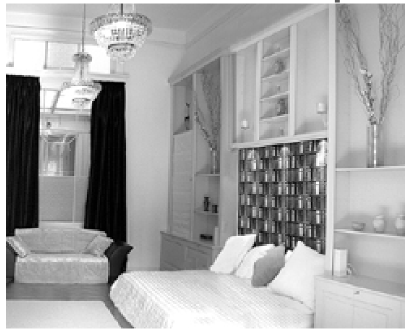


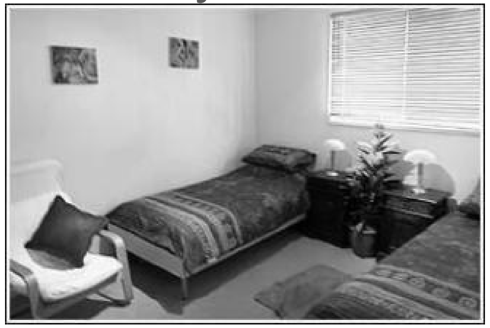
	bedroom	coast	forest	highway	industrial	insidecity	kitchen	livingroom	mountain	office	opencountry	store	street	suburb	tallbuilding
bedroom	0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.01
coast	0.00	0.89	0.01	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.07	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.04	0.00	0.00	0.00	0.00
highway	0.00	0.02	0.00	0.88	0.02	0.00	0.00	0.00	0.01	0.00	0.04	0.00	0.02	0.01	0.00
industrial	0.00	0.01	0.00	0.02	0.80	0.03	0.00	0.00	0.00	0.00	0.01	0.05	0.01	0.00	0.07
insidecity	0.00	0.00	0.00	0.00	0.02	0.90	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.01	0.04
kitchen	0.00	0.00	0.00	0.00	0.01	0.04	0.69	0.21	0.00	0.04	0.00	0.01	0.00	0.00	0.00
livingroom	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.90	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	0.95	0.00	0.04	0.00	0.00	0.00	0.00
office	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.14	0.00	0.79	0.00	0.02	0.00	0.00	0.00
opencountry	0.00	0.05	0.03	0.01	0.00	0.00	0.00	0.00	0.05	0.00	0.86	0.00	0.00	0.00	0.00
store	0.01	0.00	0.00	0.00	0.01	0.04	0.01	0.04	0.00	0.00	0.00	0.89	0.00	0.00	0.00
street	0.00	0.00	0.00	0.03	0.01	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.87	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	0.00	0.00	0.00	0.99	0.00
tallbuilding	0.00	0.00	0.00	0.00	0.00	0.01	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.]







The most misclassified category is "bedroom." The above three is the images that are examples of misclassified images. They all had the label "bedroom" but were predicted to be "living room". The reason this happens is that there is usually sofa in living room which is very similar to the bed in bedroom. Both types of furniture are rectangular and look similar. The bed in the three picture is comparatively small which would be easier to be recognized as sofa.

Part 3: ResNet

[What does fine-tuning a network mean?]

Fine-tuning refers to using the weights of an already trained network as the starting values for training a new network model. The model is then tuned to perform a second, similar task. It consists of four steps :1. The neural network model is pre-trained on the source data set. 2. Create a new neural network model that copies all model designs and their parameters except the output layer to the source model. 3. Add a new output layer to the target model and randomly initialize the model parameters of this layer. 4. Train the target model on the target data set. The output layer is trained from scratch, while the parameters of all other layers are fine-tuned according to the parameters of the source model.

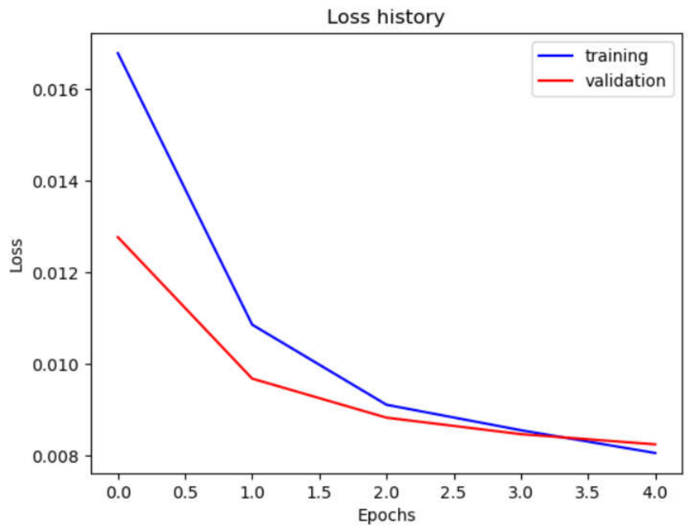
[Why do we want to "freeze" the conv layers and some of the linear layers from a pre-trained ResNet? Why can we do this?]

By freezing the conv layer and some linear layers in the pre-training model, we can perform future training rounds without destroying any of the information they contain. We can also partially train the model by freezing some layers so that we have more flexibility. Finally, we can make neural network training more efficient, a technique to accelerate neural network training.

We can use the frozen initial layer, relies on the fact: during the initial phase, the network is learning basic features. This is what we want to extract when implementing fine-tuning. We may need to change the output layer, for example, the old network was to distinguish between two classes, but in the current problem we have more classes. If the number of input features is different, so is the first layer.

Part 4: Multi-label Scene Attributes

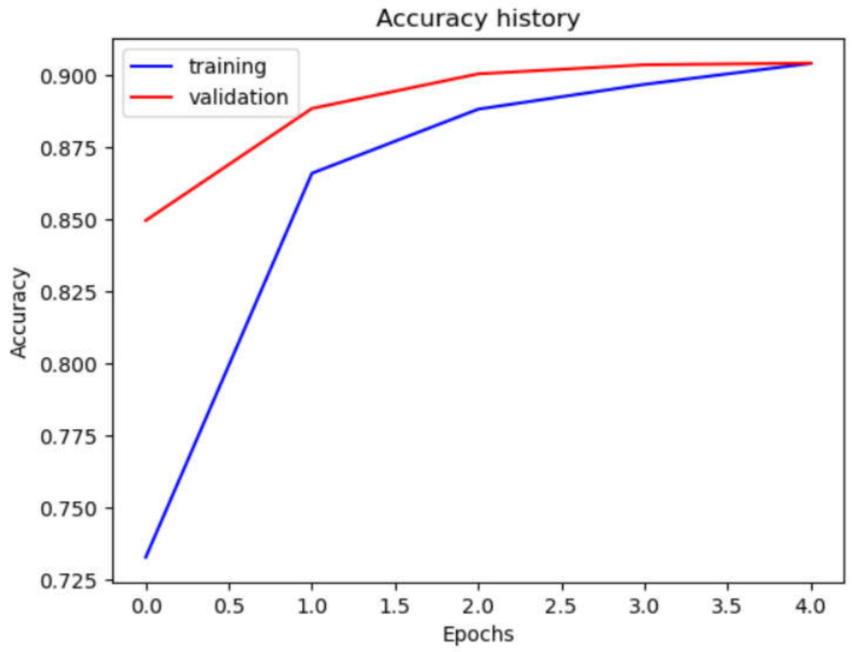
[Insert loss plot here]



Final training accuracy: 0.9042044911610129

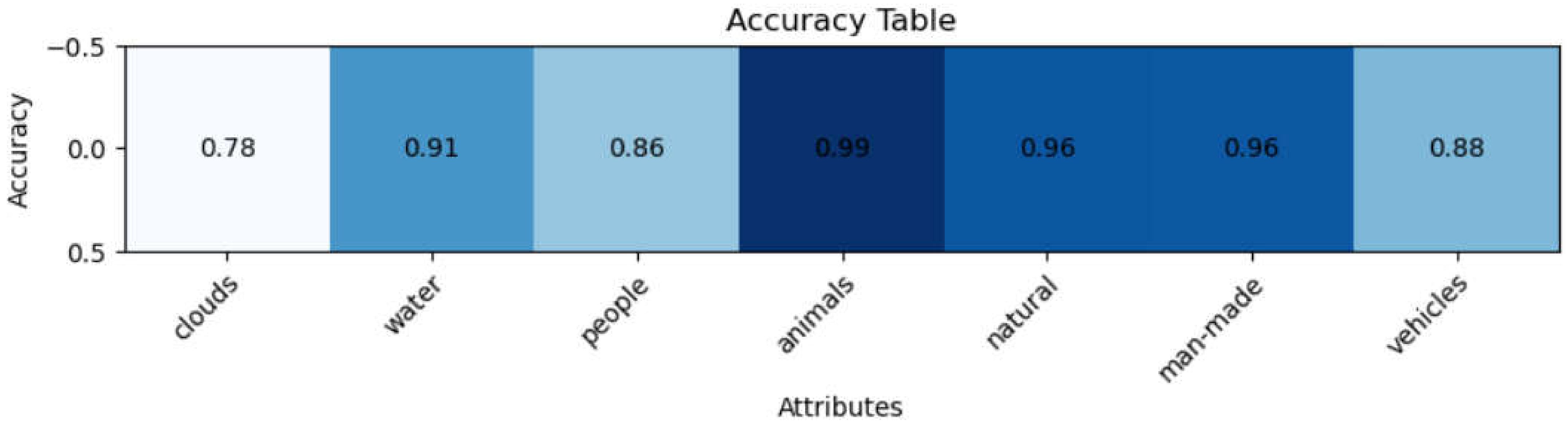
Final validation accuracy: 0.9042857142857142

[Insert accuracy plot here]



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.]

1.Changed the output linear layer from (512,15) to (512,7)

2.Changed the loss function from CrossEntropy to Binary cross entropy loss.

3.Added a Sigmoid layer at the end of the output part of the network.

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

No.

When CrossEntropy is taken as loss, the effective part for model learning is only the pred value of the position corresponding to label=1. Conversely, whether the pred value of the position corresponding to 0 is large or small, it has no influence on the calculation of loss.

When BCE is taken as loss, both the position of label=0 and the position of label=1 have influence on model learning. This means that the training will produce an effect: if a certain position of the label is 0, the output value of the model will be "close" to 0 at that position.

So BCE is more suitable for multi-label problems and CrossEntropy is not suitable.

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.]
One problem would be wary of is that the labels we choose for classification should be mutually excluded. If we choose two labels with strong correlation carelessly, the accuracy would be very low. As we can see from the accuracy table, the cloud's accuracy is the lowest. The reason for that is the cloud's shape and color is very ambiguous and can be seem as water, nature or something else. On the other hand, animals are distinct and with little correlatin with other labels, so animals have the highest accuracy.