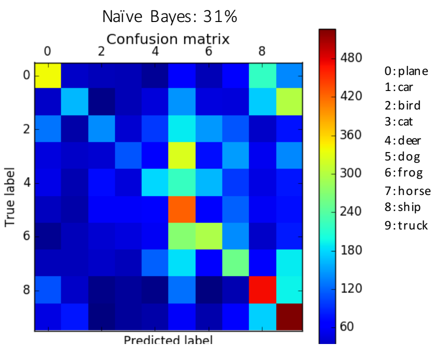
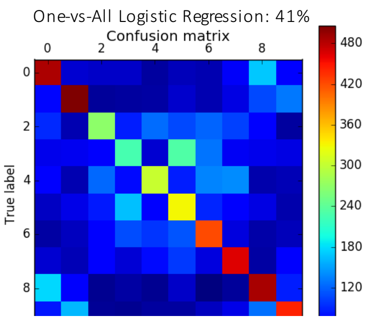
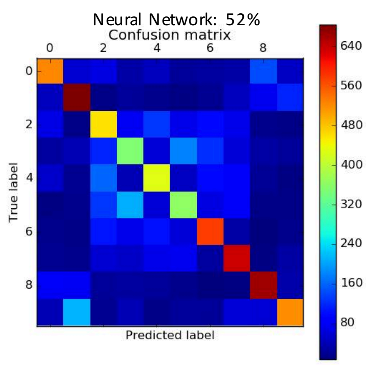
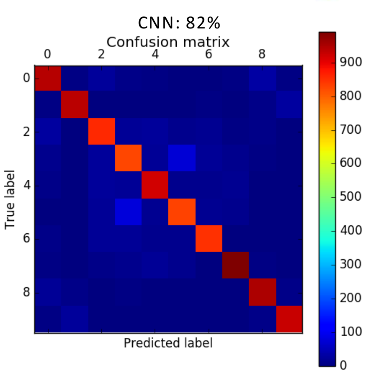
CIFAR 10 Image Recognition:
Ensembling with Support Vector Machines







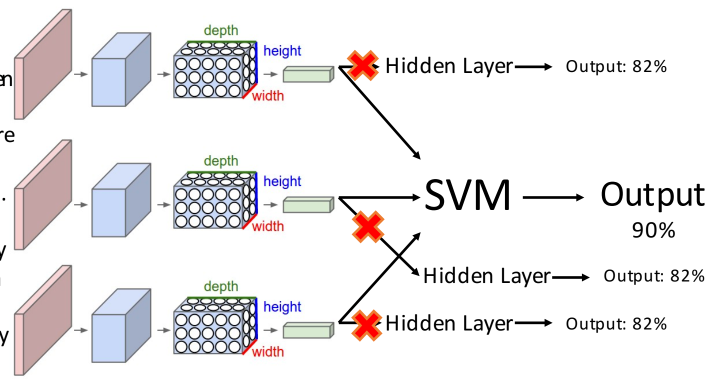


The Problem

The CIFAR-10 Object Recognition problem is a famous benchmark Machine Learning problem, which entails best predicting the class - airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck - of an image of one of those classes. Each image is given in a 32x32 png image format, and researchers are given 50,000 images to train on and need to predict on 10,000 held out test images. It is a challenge which thousands of researchers have taken up, vying for the highest accuracy possible, and it is an important one, as many technologies such as self-driving cars or facial recognition hinge upon it. Our goal was to investigate various methods for tackling the CIFAR-10 and see how well they perform, as well as combine these models in interesting ways to achieve higher accuracies. We find that running a linear Support Vector Machine on top of the features extracted using several distinct Convolutional Neural Networks (CNN's) produces a model of 90% accuracy, which is very fantastic, considering that human accuracy on the same problem is 94%.

Our Method

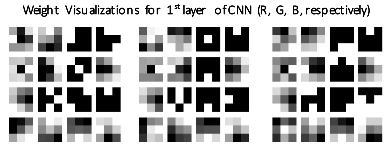
Instead of having each CNN output to a hidden layer which then does more processing before outputting, we hand that job off to an SVM. Each CNN summarizes each image into 64 key values, and the SVM, a powerful classifier, processes all of the key values of each image from each CNN into a solid classification



Other Methods on CIFAR-10

To the left, you will see various confusion matrices of different methods we tried on the CIFAR 10 Problem. We found that Naïve Bayes performed the worst (31%), likely since it does not account for the interdependence of the image pixels. Next was Logistic Regression (41%) which observes inter-pixel relationships, but only to a degree. Then was the Neural Network (52%) which heavily learns inter-pixel relationships. Finally was the Convolutional Neural Network (82%) which heavily learns inter-pixel relationships with a specific emphasis on 2D locality, a crucial aspect of images.

Weight Visualizations for 1st layer of CNN (R, G, B, respectively)



Why Our Method?

Where our method is stronger than many others in the field is its ability to produce good models very quickly or on lower compute power. Many can achieve higher accuracies, but take hours or days to do so, while our model can achieve ~86% accuracy within minutes. In addition, even with more time or compute power available, ensembling with an SVM still has a solid shot of outperforming a single CNN or another CNN ensemble.

CNN Architecture

#	name	size
0	Input	3x32x32
1	Convolution	16x32x32
2	BatchNormal	16x32x32
3	Convolution	16x32x32
4	BatchNormal	16x32x32
5	Convolution	16x32x32
6	BatchNormal	16x32x32
7	Convolution	32x16x16
8	BatchNormal	32x16x16
9	Convolution	32x16x16
10	BatchNormal	32x16x16
11	Convolution	64x8x8
12	BatchNormal	64x8x8
13	Convolution	64x8x8
14	BatchNormal	64x8x8
15	globalpool	64
16	output	10

To the left, you will see the CNN structure we used to train the various models that were underlying our SVM. This is a powerful ResNet architecture given by Microsoft Research for the CIFAR-10 Problem. For the conv layers, only 3x3 filters are used.