Support Vector Machine for CIFAR-10

Whitney Bennett

This project used the CIFAR-10 dataset which includes ten categories of images: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. Image classification has the potential to be useful in a variety of applications. For example, healthcare, security, manufacturing, defense, and more. Datasets such as CIFER-10 are great tools for practicing with image classification algorithms and sometimes, improving upon them.

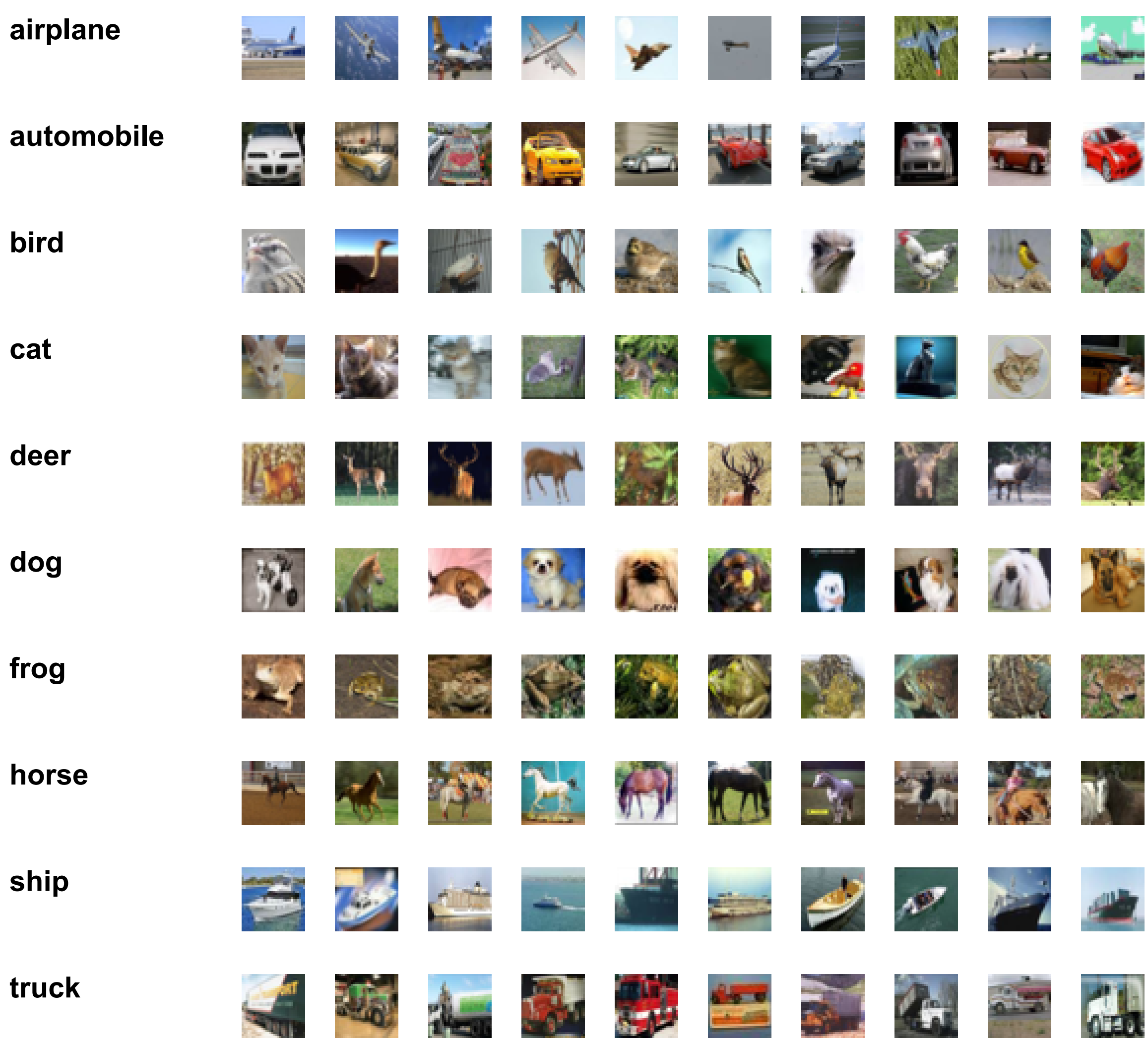


Table 1: Examples of images in the CIFAR-10 dataset. Image from https://www.cs.toronto.edu/~kriz/cifar.html

Methods:

A support vector machine (SVM) offers a memory efficient solution to some machine learning problems. In the simplest terms, suppose you have a two coordinate plane with two separate clusters of points. A SVM would determine the line that best separates these two classes. However, the data can have high dimensionality and the separation does not need to be linear. An SVM's goal is to maximize the distance between each category. One way of accomplishing this is to use soft margin SVM optimization and the kernel trick. The kernel, $K: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, is a map whose input is a point of vectors and output is a real number. Mercer's Theorem says, $K(x, y) = (\phi(x), \phi(y))$ for some ϕ, $\phi: \mathbb{R}^n \to V$, if and only if K is a positive definite kernel. Thus, only the inner product of the map is found, not the map itself. This results in the quadratic program shown in Equation 1 where C is the regularization parameter (in a way, this controls how much you want to sacrifice loss wise in order to gain in regularization), $x_n, x_m \in \mathbb{R}^n$ are training inputs, and y_n, y_m are training labels.

Equation 1 (Dual Soft Margin SVM):

$$\min_{\alpha} \left[\frac{1}{2} \left[\sum_n \sum_m y_n y_m \alpha_n \alpha_m K(x_n, x_m) \right] - \sum_n \alpha_n \right]$$

subject to

$$0 \le \alpha_n \le C, \quad \sum_n \alpha_n y_n$$

Equation 2 (Decision Function):

$$f(x) = \sum (\alpha_n y_n K(x, x_n)) + b$$

Where

x_n = training input
x = test input

The model is fitted when the optimal α_n is found. The decision function is then used to predict $f(x)$ (see Equation 2). The kernels used in this project are the linear kernel and the radial basis function (see Equation 3 and 4). Due to multiple classes, a "one-vs-one" approach is taken. This means that for all 10 classes, a 10 choose 2 binary classifier is trained. For example, if there were only three classes: airplane, car, and truck there would be three binary classification datasets: airplane vs. car, airplane vs. truck, and car vs. truck. For this project, five SVM models were trained and were compared to one neural network and one convolution neural network (see Table 2).

Equation 3 (Linear Kernel):

$$K(x, y) = x \bullet y$$

Equation 4 (RBF Kernel)
$$K(x, y) = Exp\left[\frac{-\|x-y\|^2}{2\sigma^2}\right]$$

Name	Type	Characteristics	Inputs
Model 1	SVM	C=1, linear kernel	HOG features
Model 2	SVM	C=1, RBF kernel	HOG features
Model 3	SVM	C=50, RBF kernel	HOG features
Model 4	Neural Network	SGD optimizer, cross entropy, one layer with 20 neurons, one layer with 10 neurons, sigmoid activation	HOG features
Model 5	SVM	C=0.5, RBF kernel	HOG features
Model 6	SVM	C=10, RBF kernel	HOG features
Model 7	CNN	1 Conv1D layer with 15 filters, kernel of size 5, and sigmoid activation. 1 pooling layer of size 2. 1 Conv1D layer with 10 filters, kernel of size 3, and sigmoid activation. 1 pooling layer of size 2. 1 dropout layer with 0.3 dropout. 1 dense layer with 10 sigmoid neurons. Adam optimization with a learning rate of 0.0005 and cross entropy loss. Stopped at 100 epochs.	32x32x3 image arrays

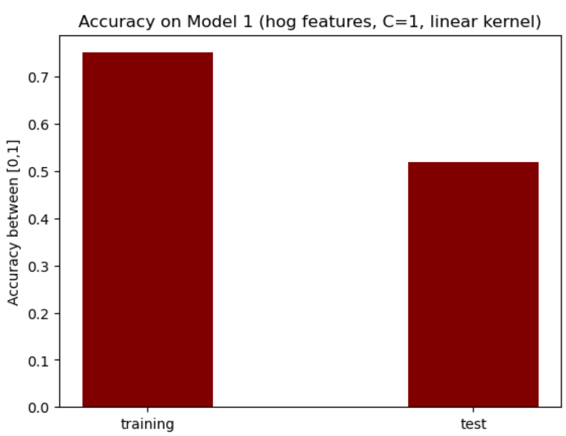
Table 2: Architectures of SVM models and the neural network model.

Data:

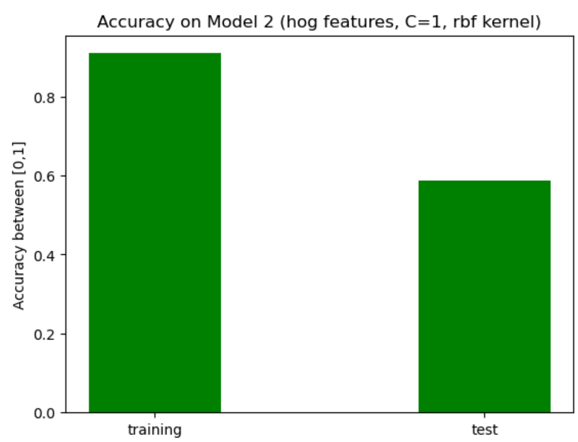
CIFAR-10 includes 60,000 32x32 color images with 6,000 images per category (see Table 1). This project used 10,000 training images and 2,000 test images from the dataset. The models are trained using HOG (Histogram of Oriented Gradients) features. This improves the efficiency of the learning process. “The technique counts occurrences of gradient orientation in the localized portion of an image” (Tyagi, 2021). This process involves calculating the gradient of each image. The gradient is a combination of the magnitude and angle from each pixel which are functions of G_x and G_y (see Equation 5 and 6). This results in a matrix of magnitudes and angles corresponding to each pixel which is then divided into square blocks of pixels. For each block, a histogram is calculated by placing pixels into bins based on the intensity of the pixel’s gradient. For example, 9 bins going from 0 to 180 degrees would result in each bin having an angle range of 20 degrees (Tyagi, 2021). Once the histograms are calculated, the process of turning them into a vector of features gets a bit complicated for the purposes of this paper, but if interested see Tyagi, 2021.

HOG Features Equations:	
Equation 5:	$G_x(r, c) = I(r, c + 1) - I(r, c - l)$
Equation 6:	$G_y(r, c) = I(r - 1, c) - I(r + 1, c)$
Equation 7:	$\mu = \sqrt{{G_x}^2 + {G_y}^2}$
Equation 8:	$\theta = \lvert tan^{-1}(G_y/G_x)\rvert$

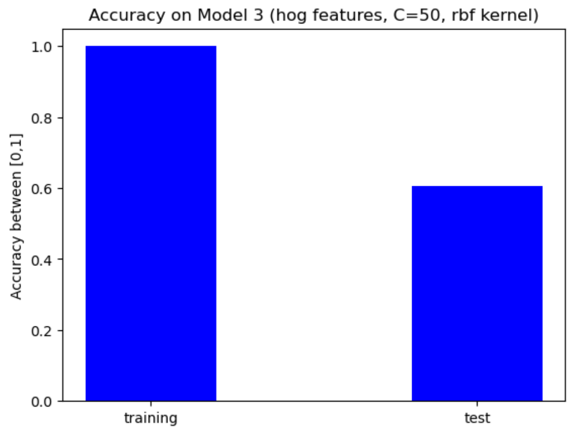
Results:



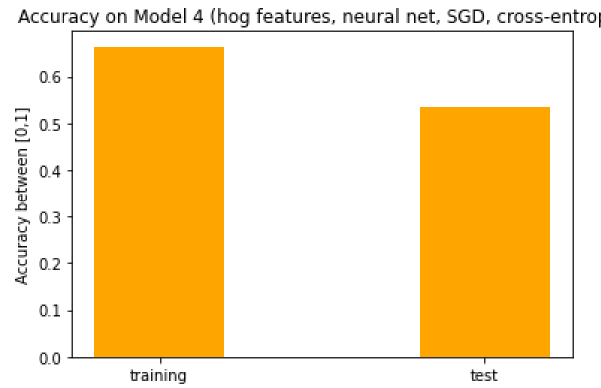
Model 1: Training and test accuracies of SVM model trained on HOG features with a linear kernel and $C = 1$.



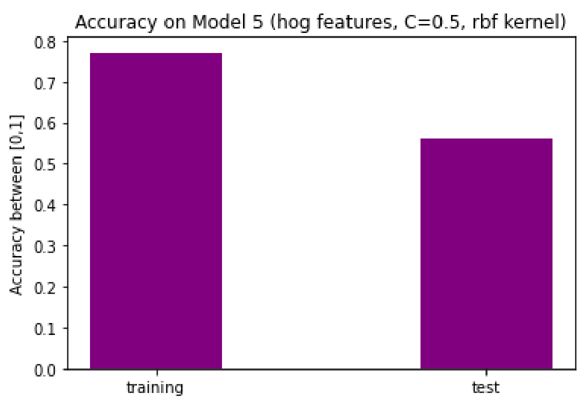
Model 2: Training and test accuracies of SVM model trained on HOG features with a RBF kernel and $C = 1$.



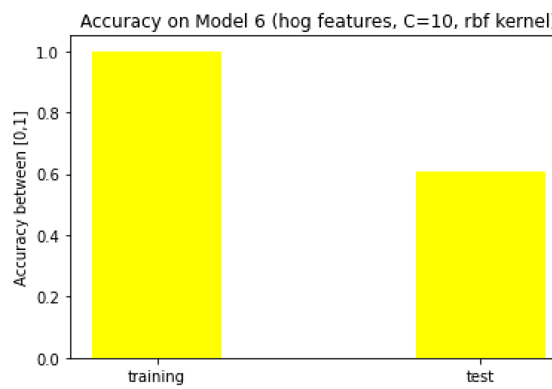
Model 3: Training and test accuracies of SVM model trained on HOG features with a RBF kernel and $C = 50$.



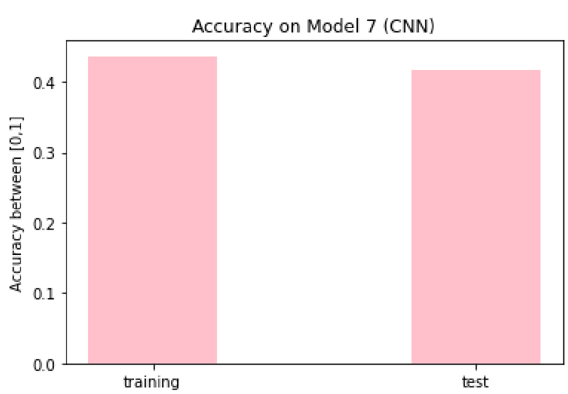
Model 4: Training and test accuracies of neural network trained on HOG features.



Model 5: Training and test accuracies of SVM model trained on HOG features with a RBF kernel and $C = 0.5$.



Model 6: Training and test accuracies of SVM model trained on HOG features with a RBF kernel and $C = 10$.



Model 7: Training and test accuracies of CNN trained on 32x32x3 images.

The final accuracies on the test data for models 1-7 respectively were: 0.5185, 0.5875, 0.6065, 0.536, 0.563, 0.6065, and 0.416. The training accuracies for models 1-7 respectively were: 0.7511, 0.9093, 1.0, 0.6635, 0.7709, 1.0, and 0.4365. The differences between test and training accuracy on models 1-7 were: 0.2326, 0.3218, 0.3935, 0.1275, 0.2079, 0.3935, and 0.0205. The only difference between Model 1 and Model 2 is the type of kernel used, and Model 2 which uses the RBF kernel did 6.9% better on the testing data than Model 1 which uses a linear kernel. Models 2, 3, 5, and 6 vary by their regularization parameters which are 1, 50, 0.5, and 10 respectively. The two models with the higher regularization parameters did 1.9% better than Model 2 and 4.35% better than Model 4. There was no change in accuracy between Model 3 and 6. Of the SVM models, Model 5 had the lowest difference between training and testing accuracy and Models 3 and 6 had the highest. The neural network model only had a higher test accuracy than the SVM model with the linear kernel, Model 1, but a lower test accuracy than the other models. However, the neural network had the lowest difference between training and testing accuracy as compared to the SVMs. The CNN model had the lowest accuracies, but also the lowest difference in accuracies of all the models.

Discussion:

Due to the comparison between Model 1 and 2, the RBF kernel seems to be a better choice. The comparison between Models 2, 3, 5, and 6 suggest that higher regularization overfits. Taking the overfitting into account, Model 5 is the best of the SVM models. Model 5 has the lowest difference between training and testing accuracy and is close to having the test accuracies that Models 2,3, and 6 have. However, all of the SVM models show evidence of overfitting.

Evaluation:

When compared to other machine learning algorithms, the SVM models perform better overall in terms of test accuracy with the neural network only performing better than one of the SVM models. However, both the neural network and the CNN perform better in terms of overfitting, especially the CNN. Changing the architectures of the neural network and CNN may improve the test accuracy. Also, increasing the number of epochs would likely improve the accuracy. Because of these results, the SVM is not a good choice for classifying the CIFAR-10 images because there are too many classes and the models overfit the data. The CNN had a lower accuracy than expected, especially given that it was the only model trained on the images and not the HOG features, but showed the highest potential.

References:

CIFAR10 : tensorflow datasets (no date) *TensorFlow*. Available at: https://www.tensorflow.org/datasets/catalog/cifar10 (Accessed: April 22, 2023).

Tyagi, M. (2021) *Hog(histogram of oriented gradients)*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f (Accessed: April 22, 2023).