CSC-345 Assignment

Thomas Briggs
Student ID: 984745

December 2021

1 Introduction

As cameras and smartphones are becoming more ubiquitous the amount of image data available is growing at a rapid rate. Previously difficult tasks such as object recognition are now becoming increasingly easier with the use of machine learning algorithms. In this paper we are going to test a variety of machine learning techniques on the CIFAR100 dataset (Krizhevsky et al., n.d.).

As we are becoming closer and closer to a world a driver less cars, object recognition is becoming a very important field to be studied, as we will be requiring fast and reliable object detection (Gupta et al., 2021). There are many different approaches available for us to use in classification space, we will be looking at a few examples and comparing their results.

We will be comparing a Support Vector Machine against Neural Networks and Convolutional Neural Networks, comparing the accuracy of the models over different stages of training and seeing how fast the model begins to overfit to the data or where the model still has room for improvement.

2 Methods

Models to be compared

Support Vector Machine

Support vector machines have shown success in classifying images quickly (Lin et al., 2011) on large datasets, as they are very resilient to erroneous samples in the data. In order to use SVM, we need to turn the image into a feature vector. In order to do this we are using histogram of oriented gradients analysis, this will allow us to use more conventional classification algorithms on the image data. Due to time and memory restrictions, we need to lower the dimensions of the feature vector. This will be done by using principle component analysis, to create new feature vector that maximise the variance of the original features. Training the SVM of a range of PCA dimensions will give us an idea of how many feature dimensions are required before we get diminishing returns, only increasing model complexity with no increase in accuracy. Due to memory restrictions we are unable to train the SVM model on the full dataset, instead we opt to train the model on a randomized sample of the training data, giving a equal chance of each category of sample to show. This should allow us to still get a accurate result for prediction without favoring one category.

Neural Network

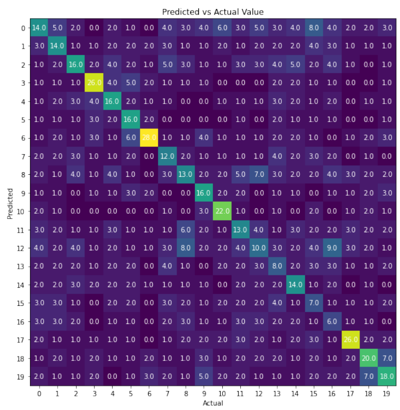
Using the same feature extraction technique for SVM, once we have the feature vectors we can then build a neural networks to be able to recognize patterns in the features to be able to detect patterns and objects. We will be able to achieve this by using a relatively straight forward neural network, with having a few layers, building up to larger and larger width. As the output space is not that large, only 20 categories, we will need to be careful to make sure we don't give our network too much width, this will result in the network learning the training data too much. This is called overfitting, and will lead to the network learning specific patterns that are found in our training data and not general patterns that can be applied to unseen data. We will be keeping metrics of the model as it trains, where we will be able to see where it is no longer decreasing it's loss and unseen data, but continues to improve it's loss in the training data.

Convolutional Neural Network

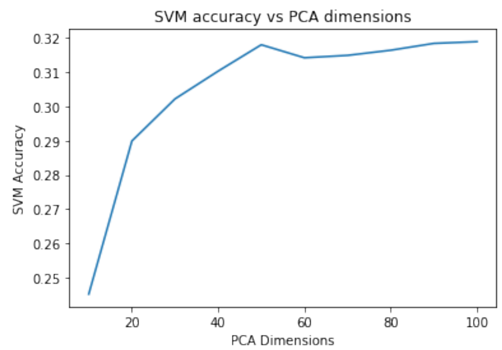
The convolutional neural network will in practive be quite similar to useing the features extracted from the images by the use of HOG, then passing it through a neural network, as the layers in the cnn first will use kernal convolutions around the image to apply filters to the image that will begin to recognize patterns, then once that is done, the image will be flattened out, and passed into dense layers, which will be very similar to that of the neural network, where we will be able to give a probability prediction for the category of the given image.

3 Results

Support Vector Machine



(a) A confusion matrix to show the ground truth of an image vs the model prediction for each category.

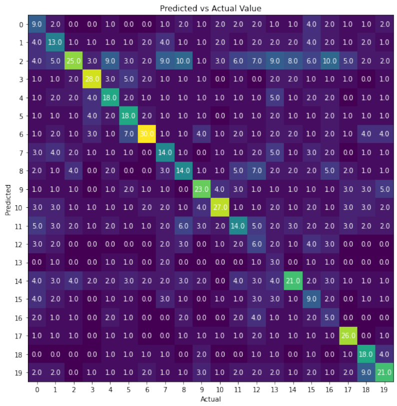


(b) A graph to show training accuracy vs the number of features dimensions given by the PCA algorithm.

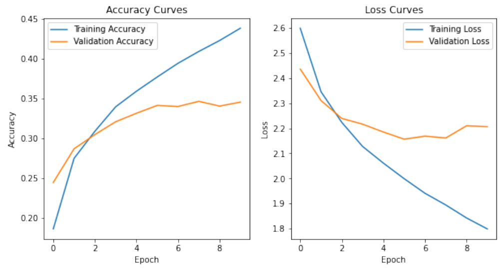
After training the SVM on a different number of feature dimensions from the PCA analysis we can quickly see the accuracy curve flatten as we increase anywhere above 50 dimensions. Evaluating the model on the testing set achieved a average accuracy of around 31.45%. As can be seen in the confusion matrix the model does seem to have slight favourite around certain categories which may be a side affect having to train the model on a random subset of the original training data, instead of being able to train the model on it all. Overall the accuracy of the model isn't bad considering how fast and model was the train, and it only used around 20% of the training data.

Neural Network

The neural network showed a slight increase in average accuracy over the SVM with a accuracy on the testing dataset of 34.04%. However as can be seen in the training accuracy and loss curves, The model after only



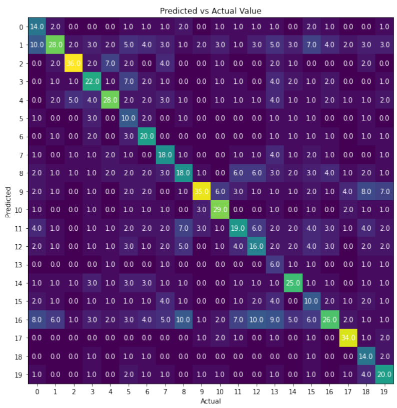
(a) A confusion matrix to show the ground truth of an image vs the model prediction for each category.



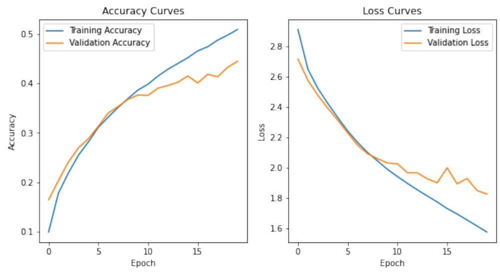
(b) A graph to show the training and loss curves of the model with its training accuracy vs the validation accuracy.

2 to 3 epochs, has dramatically decreased in its accuracy gains, and the validation loss curve even starting to get worse. This is indicating that the model is beginning to overfit to the testing data, and no longer learning general patterns from the dataset.

Convolutional Neural Network



(a) A confusion matrix to show the ground truth of an image vs the model prediction for each category.



(b) A graph to show the training and loss curves of the model with its training accuracy vs the validation accuracy.

The convolutional neural network has shown the best results with a average accuracy on the testing dataset of 42.8%, aswell as the accuracy curve of the training suggests, given more time to train, the model may be able to still improve as although the validation accuracy is dropping faster than the training accuracy there are still noticeable improve from each epoch.

4 Conclusion

Review and Analysis

The support machine vector did not provide the best result however the training time was very fast compared to most other methods and was able to be done on the full dataset. Using PCA combines with SVN, we were able to relativly quickly train the model on a subset on the training data, however we found that we were very quickly getting diminishing results with the number of feature dimensions returned by the PCA analysis.

The neural network showed disappointing results it began to overfit the training data very quickly, and showing poor accuracy on unseen data.

As shown in the results the convolutional neural network provided the best result and still showed room for further accuracy given more time to train.

Improvements

The depth of the research could be greatly improved by including a wide range of machine learning techniques, as well as a variety of neural network and convolutional neural network architectures to give more insight to what is happening inside the network and where there are still great improvements to be gained to stop the network from over fitting to the data.

References

Gupta, A., Anpalagan, A., Guan, L., & Khwaja, A. S. (2021). Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 100057.

Krizhevsky, A., Nair, V., & Hinton, G. (n.d.). Cifar-100 (canadian institute for advanced research). http://www.cs.toronto.edu/~kriz/cifar.html

Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Cao, L., & Huang, T. (2011). Large-scale image classification: Fast feature extraction and svm training. *CVPR 2011*, 1689–1696.