Comparing machine learning methods for image classification of CIFAR10 dataset

Thomas Blaine

2033755

Swansea University
Prifysgol Abertawe

Department of Computer Science

19th December 2022

Introduction

In this report, I will compare the performance of two machine learning methods for image classification on a subset of the CIFAR-10 dataset. The subset of CIFAR-10 dataset consists of 10,000 32x32 colour training images and 1,000 test images, with 10 classes of objects. The first method is support vector classification (SVC) using histogram of oriented gradients (HOG) features of each image. HOG features are commonly used for object detection tasks and have been shown to be effective at capturing important visual features in images [1]. The second method is a convolutional neural network (CNN), a type of deep learning model that has achieved great performance on a wide range of image classification tasks [2].

I evaluate the performance of both methods on the CIFAR-10 dataset and compare the results in terms of accuracy, training time, and model complexity. My experimental results show that the CNN model achieved an accuracy of 0.763, while the SVC model using HOG features achieved an accuracy of 0.53. My goal is to determine which method is more effective at classifying the objects in the CIFAR-10 dataset and to understand the trade-offs between the two approaches. In this report, I will provide analysis of the results and discuss the implications of these findings.

Method

In this study, I compared the performance of two machine learning methods for image classification on the CIFAR-10 dataset. The first method was support vector classification (SVC) using histogram of oriented gradients (HOG) features. The second method was a convolutional neural network (CNN).

Machine Learning method choices

Support vector classification (SVC) was one machine learning algorithm used for image classification on the CIFAR-10 dataset because it is a powerful method that has been widely used for image classification tasks [1]. SVC works by finding a hyperplane in feature space that maximally separates different classes of data. By using a suitable kernel function, SVC can capture complex non-linear relationships between the features and the target class, which is important for image classification tasks where the visual features may be highly non-linear.

Convolutional neural networks (CNNs) were the other method chosen for image classification on the CIFAR-10 dataset. This is because they have been demonstrated great performance on image classification tasks[2]. CNNs are a type of deep learning model that are specifically designed to process image data. They consist of multiple convolutional layers, which are used to learn feature hierarchies from the raw data, and fully connected layers, which are used to make predictions based on the learned features. CNNs can learn highly discriminative features from the data, which makes them well-suited for tasks such as image classification where the features may be extremely complex.

Feature extraction

One of the key steps in applying machine learning methods to image classification is feature extraction, which involves identifying and extracting important visual features from the raw images. For the SVC method, I used the histogram of oriented gradients (HOG) as a feature descriptor. HOG features are commonly used in object detection tasks and have demonstrated to be effective at capturing important visual features such as edges and gradients in images [1].

To extract HOG features from the images, I used scikit-learn implementation of HOG feature extraction. This involved dividing each image into small 3x3 cells computing a histogram of gradient orientations for each cell. The resulting features were then concatenated to form a feature vector for each image.

SVC approach

After choosing this approach I began training multiple models using different parameters such as pixels per cell and kernel choice. After using the linear kernel and achieving an accuracy score of 0.393. The RBF kernel was chosen next due to non-linear relationships between HOG features and target classes. This then achieved an improved score of 0.538.

CNN approach

In this CNN model, I used a combination of convolutional and fully connected layers to learn hierarchical features from the data then make predictions. The model consists of 2 convolutional layers followed by a max pooling layer, and two fully connected layers. The convolutional layers use a kernel size of (3, 3) and are activated using the 'relu' activation function. The padding parameter is set to 'same', which means that the output has the same dimensions as the input[3]. The kernel_initializer parameter is set to 'he_uniform', which initializes the weights of the kernels using the he uniform initialization method [3].

The max pooling layers are used to downsample the spatial dimensions of the feature maps[3], which can reduce the computational cost of the model and improve its generalisation ability. The dropout layers are used to randomly set a fraction of the input units to 0 at each update during training, which can help prevent overfitting and improve the model's generalization ability.

The model also includes two fully connected layers, the first with 128 units and the second with 10 units, corresponding to the number of classes in the CIFAR-10 dataset. The fully connected layers are activated using the 'relu' activation function and are followed by a 'softmax' activation function in the output layer, which allows the model to output class probabilities[3].

The model is compiled using the Adam optimizer with categorical cross-entropy loss function. The Adam optimizer is a gradient-based optimisation algorithm that adapts the learning rate based on the historical gradient information, while the categorical cross-entropy loss is a measure of the difference between the predicted and true class probabilities [4].

VGG blocks are a type of CNN architecture that was developed by the Visual Geometry Group (VGG) at the University of Oxford. The VGG architecture is characterized by its use of small, 3x3 convolutional filters and a stack of convolutional layers, which allows it to learn rich feature hierarchies from the data. Initially one VGG block was used in my implementation, but this was changed to 2 due to improved accuracy [5].

Results

The SVC model was trained using an RBF kernel and achieved an accuracy of 53.8% on the test set. The CNN model was trained using 4 convolutional layers and 2 fully connected layers and achieved an accuracy of 76.3% on the test set.

Overall, the CNN model outperformed the SVC model on the CIFAR-10 dataset, achieving a higher accuracy by 22.5%. The loss score for the CNN was 0.665. This suggests that the CNN model was able to better learn and generalize the features of the images in the dataset, likely due to its ability to automatically learn hierarchies of features with convolutional layers.

It is worth noting that the performance of both models can potentially be improved by further tuning the hyperparameters or increasing the size of the training set. The larger dataset would preferably been used but due to computational resources available the smaller dataset had to be used.

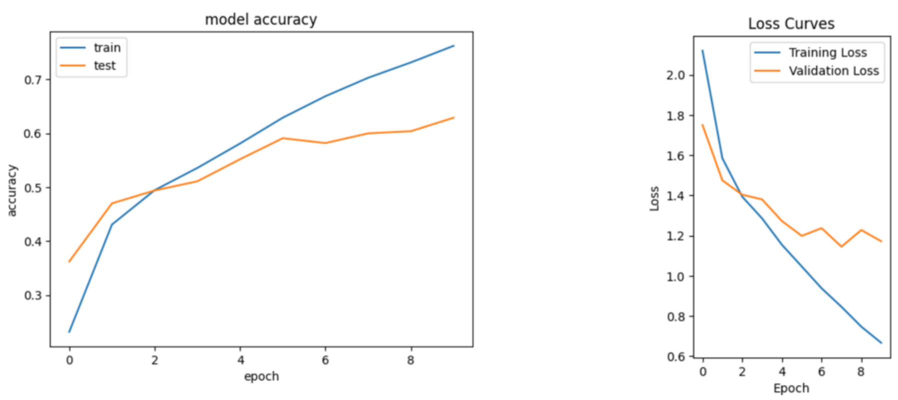

Figure 1: Accuracy and loss curves for CNN implementation

Confusion matrices

```
print(confusion_matrix(tstLabels, y_pred))
```

```
[[57  4  8  2  6  1  3  2 15  2]
 [ 5 61  4  1  3  0  5  1 14  6]
 [12  1 31 10 12 11 11  5  5  2]
 [ 3  1  9 42  8 16 10  6  2  3]
 [ 8  1  4  8 60  3 11  2  2  1]
 [ 1  1 12 18  7 45  9  5  1  1]
 [ 2  3  9  6 10  4 63  1  1  1]
 [ 0  1  6  9 10 13  0 56  2  3]
 [15 11  1  0  4  1  2  0 61  5]
 [ 6  4  3  0  8  3  2  3  9 62]]
```

Figure 2: Confusion matrix for test data on SVC model.

```
print(confusion_matrix(true_y, predictedLabels))
```

```
[[59  4  8  1  3  4  0  1 13  7]
 [ 0 72  3  2  0  1  2  4  4 12]
 [ 8  0 52  5 11 10  5  7  1  1]
 [ 4  0 12 52  5 11  8  4  1  3]
 [ 2  0  6  6 58  4  9 10  1  4]
 [ 0  2 12 21  4 46  4  8  0  3]
 [ 0  0  9  4  6  2 76  1  0  2]
 [ 1  0  6 10  6  5  1 70  0  1]
 [ 6  6  3  1  4  0  2  0 75  3]
 [ 1 19  1  0  0  1  1  4  4 69]]
```

Figure 3: Confusion matrix for test data on CNN model.

Conclusion

Overall, my results showed that the convolutional neural network (CNN) model outperformed the support vector classifier (SVC) model on the CIFAR-10 dataset. The CNN model achieved a higher accuracy as well as a lower loss score during the training process. These findings suggest that the CNN model was able to better learn and generalize the features of the images in the dataset, likely due to its ability to automatically learn hierarchies of features using convolutional layers.

These results align with previous research that has demonstrated the effectiveness of deep learning models, such as CNNs, for image classification tasks. In contrast, the SVC model, which used a RBF kernel, was less accurate and had a higher loss score on the test set.

Overall, this comparison of SVC and CNN models on the CIFAR-10 dataset highlights the potential advantages of using deep learning models for image classification tasks. Future research could explore the impact of different hyperparameter settings or larger training sets on the performance of these models.

Bibliography

[1] H. S. Dadi and G. K. Mohan Pillutla, “Improved face recognition rate using hog features and SVM classifier,” IOSR Journal of Electronics and Communication Engineering, vol. 11, no. 04, pp. 34–44, 2016.

[2] M. Hussain, J. J. Bird, and D. R. Faria, “A study on CNN transfer learning for image classification,” Advances in Intelligent Systems and Computing, pp. 191–202, 2018.

[3] “Module: Tf.keras.layers :   tensorflow V2.11.0,” TensorFlow. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers. [Accessed: 18-Dec-2022].

[4] [K. Team, “Keras documentation: Optimizers,” Keras. [Online]. Available: https://keras.io/api/optimizers/. [Accessed: 18-Dec-2022].

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” Proceedings of the British Machine Vision Conference 2014, 2014.