CSC-345 Coursework 1

Ted Taylor 976335

06/12/2020

0.1 Introduction

Image recognition has a vast array of applications: facial recognition, google lens and image organisation are just a few of the possibilities. The technology driving image recognition is improving rapidly as the technology begins to become more prevalent in our daily lives.

There are various approaches that can be taken when implementing image recognition, there are unsupervised methods such as K-Means and Gaussian clustering, supervised methods like Regression, Support Vector Machines and Neural Networks as well as dimensionality reduction to gain deeper insight into the structure of the data.

0.2 Methodology

To carry out these methods, appropriate feature extraction must take place. This will be carried out using Histogram of Oriented Gradients (HOG) extraction. HOG extraction focuses on the change in gradient between pixels [5] analysing the images gradient direction and magnitude to produce relevant data and provide an insight to the shape of an object in a picture. This is a good starting place for our implementation as the shape of the image is the key indication as to what the image represents in addition to factors such as colour.

0.2.1 K-means and Gaussian

K-Means and Gaussian approaches are used more for clustering data than classifying, so they are not very suitable for this problem and can often yield unexpected results. A good example of this would be the K-Means algorithm. K-Means can be very harsh in terms of decision making, it often does not account for outliers and can regularly miss cluster objects. Gaussian Mixture Modelling (GMM) is slightly better as it does not partition the data, rather it scales the data between likelihoods of belonging to a specific cluster. However, GMM suffers from the same drawbacks as K-Means as both are unsupervised.

0.2.2 LDA and PCA

Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are two other techniques that can be tested. After researching the use of these two methods in image recognition, primarily facial recognition, it is suggested that PCA is favourable on small data sets whereas LDA is better on larger datasets. This was studied in a paper published by Aleix Martinez and Avinash Kak Titled “PCA versus LDA”, which claimed the performance of both vary depending on the size of dataset and validated this claim in their paper [6]. As the dataset in question is rather large, I will disregard PCA and focus on LDA.

Although these two methods are unlikely to match up to other techniques it can be useful to offer insight into the structure of the data and how I should carry out the other approaches.

0.2.3 Support Vector Machines

Supervised techniques are more suitable when tackling a problem such as image recognition. Support Vector Machines (SVMs) are a good way to go and can be effective in applications such as facial recognitions. This is discussed in a paper published at Nanyang Technological University Titled 'Face Recognition by Support Vector Machines', concluding 'it appears that the SVMs can be effectively trained for face recognition'. The experimental results show that the SVMs are a better learning algorithm than the nearest center approach for face recognition. [1]. In addition, one paper published at NEX Laboratories America reported a 71.8% classification accuracy in the ImageNet image set [4]. The research suggests SVMs are one appropriate method which is relatively simple to implement when compared to other algorithms. SVMs also have further applications in areas such as motion recognition as detailed in a paper published by the Royal Institute of Technology (KTH) [8]..

0.2.4 Neural Networks

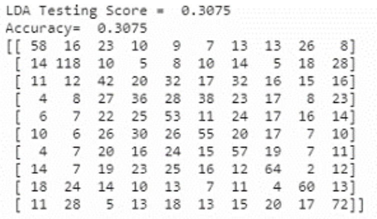
Deep learning techniques such as Convolutional Neural Networks (CNN) are often used in image recognition and have wide reaching application such as analysing images containing malaria or cholera to and decipher which is present. One study found a classification accuracy of their CNN to be as high as 94% [9]. However, neural networks can have a long execution time if the application is not correct. This will need to be avoided as having an algorithm that executes in a feasible amount of time on a large data set is just as important as the accuracy. Neural networks also work well to analyse very minute changes in images, performing well on datasets such as CUB-200-2011, a data set containing images of different species of bird. When implementing NNs or CNNs it is important to consider the order of the data as some of the libraries I will be using require data in a specific order. This means I will have to use a line of code such as np.transpose(trnData,(3,0,1,2) to re-order the train and test data to SxWxHxC . Also, neural networks often work better on shuffled data sets, so it would be a good idea to add a function such as the numpy shuffle() function to randomize the order of the data. The CNN requires the data to be fed slightly differently, so I will add a section in the code to properly organise the data. The main thing is to ensure the use of $keras.utils.to\_categorical(ytrain, 10)$ and $keras.utils.to\_categorical(ytest, 10)$ to ensure 10 categories are used.

0.3 Results

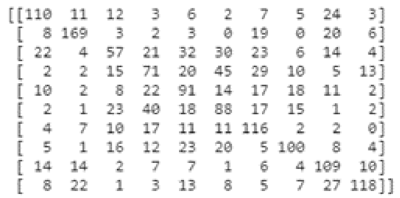
I began by testing PCA to see if this unsupervised method could provide insight into the characteristics of the data. My results showed PCA with *n_components* = 2 produces a score of 14272, I then tested with *n_components* = 10 yielding a score of 14632. This shows how much data was able to be retained while massively reducing the dimensions. This is probably because the data is often related which benefits dimensionality reduction, showing it can be a very useful tool when approaching problems such as this.

I then moved on to supervised approaches starting with LDA. I attempted to classify the data using LDA but only yielded a score of 0.375. The figure 1 confusion matrix shows the large number of false positives and false negatives produced. The main categories to suffer from this were categories 3,4,5,6,7. This is likely because they are all animal images and have a relatively similar structure making them hard to differentiate.

After implementing SVM it is clear my hypothesis was incorrect and that SVM it is not the preferred method. Although it is a good technique for image recognition, SVMs suffer from the 'curse of dimensionality' meaning the execution time is very long for large data sets. The curse of dimensionality describes the difficulty of analysing data in larger dimensional space (such as this data set), causing computational time to suffer greatly taking over 15 minutes to complete execution. The results I achieved in my prediction were however good. Achieving an accuracy of 0.5145, the figure 2 confusion matrix shows the number of true positive and true negatives (along the diagonal) to be relatively high when compared to the number of false positives and false negatives.



```
LDA Testing Score =  0.3075
Accuracy=  0.3075
[[ 58  16  23  10   9   7  13  13  26   8]
 [ 14 118  10   5   8  10  14   5  18  28]
 [ 11  12  42  20  32  17  32  16  15  16]
 [  4   8  27  36  28  38  23  17   8  23]
 [  6   7  22  25  53  11  24  17  16  14]
 [ 10   6  26  30  26  55  20  17   7  10]
 [  4   7  20  16  24  15  57  19   7  11]
 [ 14   7  19  23  25  16  12  64   2  12]
 [ 18  24  14  10  13   7  11   4  60  13]
 [ 11  28   5  13  18  13  15  20  17  72]]
```



```
[[110  11  12   3   6   2   7   5  24   3]
 [  8 169   3   2   3   0  19   0  20   6]
 [ 22   4  57  21  32  30  23   6  14   4]
 [  2   2  15  71  20  45  29  10   5  13]
 [ 10   2   8  22  91  14  17  18  11   2]
 [  2   1  23  40  18  88  17  15   1   2]
 [  4   7  10  17  11  11 116   2   2   0]
 [  5   1  16  12  23  20   5 100   8   4]
 [ 14  14   2   7   7   1   6   4 109  10]
 [  8  22   1   3  13   8   5   7  27 118]]
```

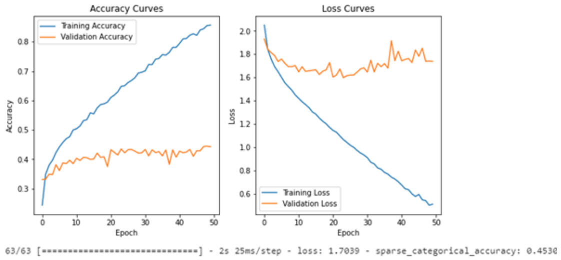
,

Figure 1. LDA Results _____ Figure2. SVM Results

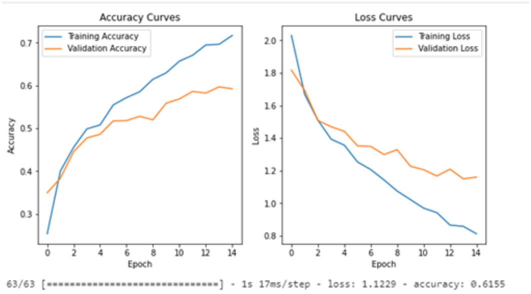
I then progressed to deep learning techniques; I first tested the suitability of a Neural Network (NN). Which did not perform as well as I had hoped so quickly moved on.

Finally, I tested the performance of a Convolutional Neural Network (CNN). The CNN performed the best out of all the methods I have proposed. My CNN achieved and accuracy of 0.618. This was achieved using various tweaks to the model to provide the best outcome. The first tweak I used was to add an additional filter layer (124 filters) (more filters are added to deeper networks as discussed in this article from pyimagesearch.com [7]. I also altered the model's padding parameter, padding allows us to specify whether the special dimensions can be reduced [7], and so I added padding = "same" to my CNN model. Before these tweaks were added I was achieving 0.55 proving these sorts of tweaks can have a great impact on performance. Finally, I changed the activation from Relu, which I initially used, to LeakyRelu. After carrying out some research I found that LeakyRelu allows for the use of small (non-zero) values, this is important because as the dataset is normalised there are a large amount of these sort of values [3].

The CCN and the addition of a convolution layer likely performs better for two reasons. First, the convolution layer provides simpler processing, improving time complexity dramatically. Second, the convolution layer benefits datasets with a lot of related data, images are a great example of this.



```
63/63 [==============================] - 2s 25ms/step - loss: 1.7039 - sparse_categorical_accuracy: 0.4530
```



```
63/63 [==============================] - 1s 17ms/step - loss: 1.1229 - accuracy: 0.6155
```

Figure 3. NN Results (Left) Figure 4. CNN Results (Right)

0.4 Conclusion

As predicted a supervised approach is favourable to this problem, especially since we are given labels in our dataset. Our SVM and LDA implementation yielded reasonable accuracies, but suffered from extremely long execution times, partly due to the 'curse of dimensionality'. Out of the two, SVM would be the preferred option as it had a similar running time to LDA, but a higher accuracy of 0.514 (whereas LDA yielded 0.315). Similarly, PCA is useful for gaining insight into the dataset. However deep learning techniques, namely CNNs, are the most suitable approach to this problem as they work well on datasets that contain related data and execute in reasonable time.

The accuracy of these algorithms can be improved with various tweaks and changes. For example, a paper published by Harbin University of Science and Technology, China, concluded that an improved SVM using two mixed kernel functions of Wavelet and Gaussian functions improved efficiency and accuracy [2]. In addition to this, more computing power will improve the results for SVM and LDA which currently suffer from long execution times, however this us due to the limitations of my hardware. If this experiment were to be carried out again, I would recommend more computing power and find out what other tweaks can be made to improve the methods.

Bibliography

[1] Guodong Guo, S. Z. Li, and Kapluk Chan. Face recognition by support vector machines. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 196–201, 2000.

[2] Hao Huanrui. New mixed kernel functions of svm used in pattern recognition. *Cybernetics and Information Technologies*, 16(5):5 – 14, 01 Oct. 2016.

[3] Keras. Keras - leakyrelu layer. keras.io. Accessed: 07/12/2020.

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

[5] Satya Mallick. Histogram of oriented gradients. Huffington Post. Accessed: 07/12/2020.

[6] A. M. Martinez and A. C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[7] Adrian Rosebrock. Keras conv2d and convolutional layers. pyimagesearch. Accessed: 07/12/2020.

[8] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, 2004.

[9] Boukaye Boubacar Traore, Bernard Kamsu-Foguem, and Fana Tangara. Deep convolution neural network for image recognition. *Ecological Informatics*, 48:257 – 268, 2018.