A Comparison of Machine Learning Object

Classification Approaches

Final Report

**Team 37**

Alex Conroy  
Jordan McCallum  
Nebojsa Ajdarevic

Student Numbers

08324841

09155881

10434348

CAB420 Assignment 2

Queensland University of Technology (QUT)

GPO Box 2434, Brisbane QLD 4001, Australia

May 3, 2020

Introduction

Image processing and object recognition is fast becoming one of the most important areas of research within Deep/Machine Learning. The core concept of object recognition is to train a computer to do see things as humans do. It has potential applications in a wide variety of industries from law enforcement to manufacturing, healthcare or agriculture. Giving machines the ability to recognise objects based on what they look like can potentially be implemented in lifesaving ways, allowing self-driving cars to avoid hitting a small child.

Research Question and Motivations

This paper will be looking to address the progress of deep learning models from 2006 to present (2020), while expanding on premises first introduced in the conference paper ‘In defense of Nearest-Neighbor based image classification’at the *IEEE Conference on Computer Vision and Pattern Recognition* (Boiman, Shechtman , & Irani, 2008). This will be achieved by running the models on a common community standard dataset for this period, Caltech 256. The paper will set out to create multiple deep learning models using methodologies posed in the conference paper, including the addition of a modern model from the same family. Then compare and evaluate them against the benchmarks provided in the dataset’s creation documents.

The general aim of these models will be to accurately recognise objects and assign them to different categories given what the model has identified as the best match for the object. Given the important applications that this type of analysis can be used for, described above. It is important that potential models are accurate, computationally efficient, and does not take too long to train.

Related Work

With the introduction of challenging datasets like Caltech-256 and the increasing real world need of accurate object recognition many approaches have been recommended within the research. More traditional Machine Learning approaches such as the Scale Invariant Feature Transform (SIFT) algorithm have been found to be slow and really only a benefit to smaller datasets where the Deep Learning methods struggle to gain good accuracy scores (Ramya & James , 2019)

Papers quite often focus on comparison of different variants of a specific method such as a Convolution Neural Network (CNN), or Linear and Quadratic Support Vector Machines (SVM) (Tropea & Fedele, 2019) for object recognition tasks like Caltech-256.

In recent years there have been a significant number of papers presented on the comparison of different classifier methods within the context of object recognition. In this regard Convolution Neural Network’s (CNN) have been the focus of many researchers when it comes to image classification due to its inherent ability to understand and extract features of images and classify them from a large number of image data automatically, thus reducing the need for any feature engineering (Yan & Lui, 2016).

SVM’s have been shown to be effective at large scale object recognition tasks (Lin, et al., 2011) use a parallel averaging stochastic gradient descent (ASGD) 1 vs all SVM on an ImageNet 1000-class classification task. Even on such a large-scale data, the authors report relatively fast convergence and relatively low training times. This makes them an often-used comparison for classification problems.

This paper will look to focus and extend on the principles of a number of recent works, in particular the report will focus on a conference paper by Boiman et al (2008) in which SVM classifiers were compared to a Naive-Bayes nearest-neighbour (NBNN) based classifier to discuss the benefits, and undervaluation of the non-parametric NN family. The performance was then compared on several challenging databases, including Caltech-256, with positive results recorded. This approach will then be extended to include a modern Convolution Neural Network alongside a secondary NN model and an SVM like the one produced by the Boiman et al (2008) paper. These will all be compared to the benchmark results provided by the Californian Institute of Technology with the original dataset paper (Griffin, Holub, & Perona, 2007). This is to hopefully give a ‘yardstick’ measurement of the progress over the last fourteen years.

Data

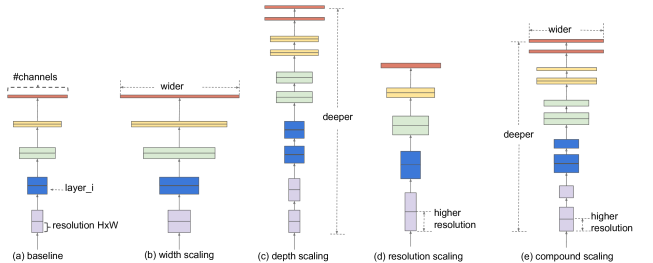
The dataset being used for this project is the Caltech 256 dataset. It contains images over 30,000 objects in 256 distinct object categories with each category containing a minimum of 80 images and a mean of 119. The dataset is collated from two different popular image databases ensuring for a variety of orientations, lighting conditions, and backgrounds. The dataset was organised in a read-to-use format, this means that the need for cropping or other processing should not be needed in most instances.

Due to the clean nature the caltech-256 dataset is presented in, very minimal data pre-processing was required. Since colour was not a core component to class classification the data was converted into grayscale to decrease model training time. Data was split into train, test and validation sets containing 56%, 30% and 14% of the data, respectively.

Due to the number of classes in the dataset, a receiver operating characteristic (ROC) curve will be used to illustrate the ability of the models, as opposed to a confusion matrix.

Methodology

The first model to be used will be a Convolution Neural Network (CNN). Specifically, the model will use EfficientNet, the model was released approximately 12 months ago by Google (Tan & Le, 2019). The model has set records for both accuracy and computational efficiency. EfficientNet uses multidimensional scaling to avoid diminished returns on computational efficiency. The model also employs compound scaling to attempt a synergy of dimension scaling, this is achieved by applying a constant ratio to each dimension under scale, again this is done in the name of efficiency. These measures have helped the model gain a reported 5x reduction in parameters required to achieve the same accuracy rating as comparable CNNs. Furthermore, EfficientNet has displayed similar magnitude gains in time when using heat maps to capture items in a picture. Overall, it is believed that this model will make an excellent choice for the modern CNN for comparison in this classification challenge.



*Figure 1 EfficientNet model with compound scaling example*

The specific layout of the EfficientNet is quite complicated, however it contains the basic neural network architectures. It contains batch normalization layers to reduce overfitting, max pooling layers to reduce dimensionality, convolution layers and depthwise convolution layers to get a deeper understanding of the image. all of this was then flattened and passed to a fully connected layer that turns these into outputs. This can be seen in figure 2 below.

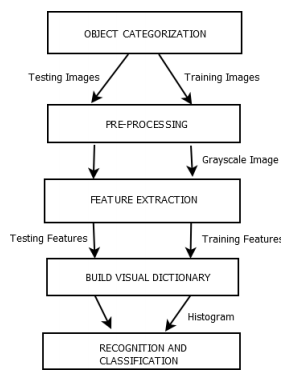


*Figure 2 EfficientNet framework built for the Caltech-256 database*

The second model is a Bernoulli Nearest Neighbour based model. Nearest Neighbour models have historically been underused for classification tasks in favour of their more appealing counterparts, deep networks. However, Boiman (2008) et al paper was published with the intention of bringing them back to the forefront of research. The Boiman et al (2008) paper used a simple Naive-Bayes nearest neighbour (NBNN) classifier which, despite its lack of training, ranks amongst the top of the line networks for classification accuracy. The model produced in this task is our inspired continuation of applying NN family models to classification problems. As such the construction of a Bernoulli Nearest Neighbour model, this will form the third of three NN models to be compared to the SVM model described below along with the dataset benchmarks.

The Bernoulli model (BNN) events are treated as independent Boolean inputs. In a similar fashion to the NBNN classifier this model is popular mostly with document classification. As such there are no instances of this method being used on the dataset. The probabilistic formula for the BNN is derived from the likelihood function shown in figure 3. The formula basically states that the likelihood of a class() being responsible for generating a data point () is the sum of probabilities based on the Bernoulli/Binomial distribution.

The last model is to be a multiclass Support Vector Machine (SVM). SVM’s are supervised learning algorithms which are commonly used for classification and regression tasks. SVM’s use hyperplanes or sets of hyperplanes to classify or regress variables. SVM’s aim to maximize their margin sizes by increasing the space between classes, as a result, they are robust and not affected by outliers. SVM’s generally perform classification effectively, even with high dimensionality. As a result, an SVM is one of the benchmarks used by California University of Technology for their Caltech-256 dataset.



*Figure 3 SVM Model*

The specific SVM will be a “combined bag of features”. This approach is designed to emphasize better recognition and classification accuracy. The approach will be modelled after the paper ‘Object Recognition Using SVM Based Bag of Combined Features’ by Mehboob et al (2019). The paper outlines the use of Root Scale invariant feature transforms (SIFT) and speeded-up robust features (SURF) to produce state of the art performance benchmarks in object recognition on the Caltech-101 dataset. The aim of this model is to extend this to the Caltech-256 dataset and compare this model to the other two methods mentioned above.

Evaluation and Discussion

The CNN was the first model trained. Due to the complicated nature of deep networks and the large size of the dataset, this model was very computationally intensive. The EfficientNet CNN model trained for ~4.5 mins on the Caltech-256 dataset. However, unlike the other two models this was run through the GPU rather than the processor directly, this created a 26 sec time for each epoch. It achieved an accuracy of 13%. When the dataset was created the state-of-the-art model released as a benchmark was a Spatial pyramid matching algorithm and it had an accuracy of 34% on the entire dataset. Since, a state-of-the-art CNN model can achieve accuracy scores of up to 91% (Cao, Wu, Chen, Cui, & Feng, 2019), while this model has achieved significantly less than this. The shortfalls and possible improvements of the model will be discussed below.

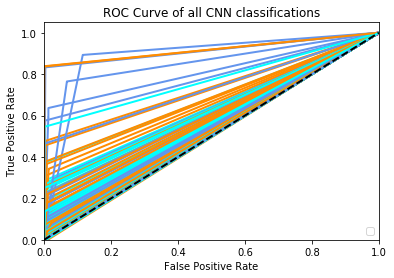
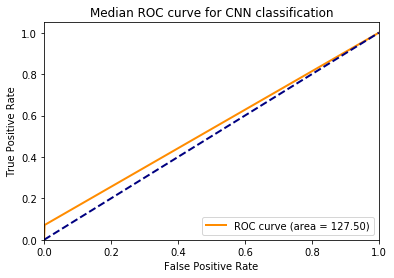


Figure 4 ROC for CNN

Figure 4 clearly shows that the model is extracting detail from some classes but barely learning anything for others. The model underperforms expectations for a variety of reasons. Since we increased the image input size from 64x64 to 128x128, and this did not change the final accuracy results, it appears that this network is not starved for meaningful features to distinguish. Batch size was varied from 8 to 32, shuffled the input data at every training session, and had almost identical accuracy results for the training, validation and testing sets, this suggests that the model did not overfit. The model was trained for 100 epochs on all data. An exploding loss & no increases in accuracy over time was observed, it seems that time to fit was not a constraint.

This might come down to the loss function being a softmax cross entropy with logits, with the addition of a softmax activation to the final model layer hindered performance greatly. However, since this loss function applies only when training, if interference was made without a softmax activation on the final layer, loss would explode, and predictions would be wildly out. A possible point for future works would be to remove softmax activation for training and only use it for testing. The resulting dataset had a 13% test accuracy and a 22% test accuracy. Given this new development, it is possible that there was some over-fitting to some specific labels. This would make sense, as even when shuffling all data every training session, it is entirely possible for some classes to come out way over-represented in the training set. This possibility would be exacerbated by the imbalance in the available data, with some labels having a fractionally massive number of images.

Through these ruled out causes, and considering the lack of accuracy increase over time, it is suggested that the constraints on performance were due to two main factors. One, the network did not have a high enough dimensionality to make decisions complex enough to distinguish between 256 categories. Two, the architecture of the model was such that convolution blocks between filters were identical, and it is possible that a more tailored network to the dataset would have helped in accuracy greatly.

The SVM was the second model constructed. SVM was trained for over 2-hour, on a single thread of the processor. It achieved an accuracy of 3%. Again, the model vastly underperforms when compared to Mehboob, Abbas, and Rauf’s study (2018) which achieved a correct classification rate of 65% and 72% at 15 and 30 training images per class.

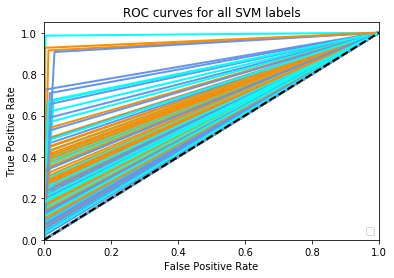
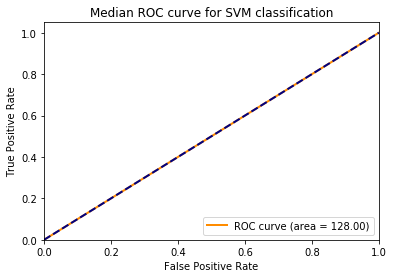


Figure 5 ROC for SVM

It can clearly be seen that the SVM underperforms expectations in figure 5, but predicts some classes really well. There are a few reasons for this underperformance. Due to the complexity of our dataset feature extraction is important as it will give valuable information for matching these images. Feature extraction using the bag of features method has proven to be a powerful tool (Mehboob, Abbas, & Rauf 2018). Existing approaches use a variety of feature extraction tools and methods such as Root-SIFT and SURF algorithms to construct a Visual vocabulary for dictionary representation. SIFT aids in reliable matching amongst different views of the same object, a major issue in our dataset. They then use feature quantisation and clustering to reduce the dimensionality of the dataset. The Images are then Represented by Frequencies of Visual Words, and only then is the data fit by a multiclass SVM.

The last model constructed was the Bernoulli Naïve Nearest Neighbour model. The model training time varied from approx. 5 – 10 mins using the same single thread of the processor as the SVM model. This model is the fastest of all the models trained. The accuracy achieved by the BNN was 4.5%. While this was an achievement from the BNN given the resources taken. It is apparent that the model is much more suited for a dataset with a significantly smaller number of classes. This is understandable from a mathematical point of view as the Bernoulli algorithm is one of a binary nature.

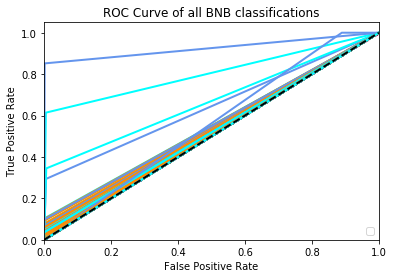
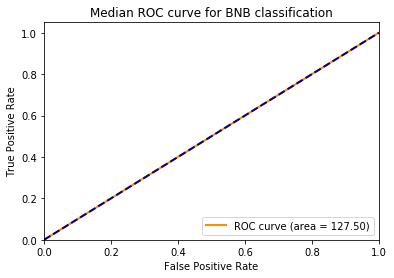


Figure 6 ROC for BNN

The BNN clearly lacks predictive power as cane seen by the ROC curve above. There are several possible things that could be done to improve the BNN. Image filtering or semantic segmentation could potentially have a large impact on the accuracy. However, with a general accuracy as low as this one. It is the recommendation of this report that any efforts to improve this model be used instead to investigate more appropriate models or improve one of the other models investigated in this report.

In the ROC curves above it is clear that the EfficientNet CNN is the most accurate model. However, due to its size and the number of parameters the CNN was run on a GPU with 8 GB of dedicated RAM. Therefore, for computational efficiency the Bernoulli Naïve Nearest Neighbour was the clear winner here. Once this was taken into consideration, the discrepancy in accuracy was still too great in the favour of the EfficientNet CNN. This is backed up by the researched related work, in which a significant number of the most recent articles focus on the use of CNNs. It is for these reasons that the EfficentNet CNN is the model that would be recommended for further analysis for the dataset.

A final accuracy comparison of the different models proposed is shown alongside the original benchmark for the Caltech-256 dataset as provided by Griffin, G., Holub, A., & Perona, P. (2007) in Table 1.

|  |  |
| --- | --- |
| Model | Accuracy |
| EfficientNet CNN | 13% |
| Bernoulli Naïve Bayesian | 4.5% |
| SVM | 3% |
| Benchmark (SPM) | 34% |

*Table 1 Model Comparisons*

These results are very transparent in showing that all results achieved in this report are significantly below the benchmark accuracy. However, a number of recommendations have been given as to how to improve the models. In particular a number of promising avenues for improvement were given for the best performing model, EfficientNet CNN. With these improvements in place it is believed that a significantly closer accuracy could be achieved to this benchmark.

## Future works

CNN’s seem to have the upper hand in the literature. While future work within the object classification field can continue in many directions one model appears to be an excellent candidate for further exploration. Deep-Clustering is the combination of CNN’s and clustering and is a relatively new technique originally presented by Facebook AI team (Caron, Bojanowski, Joulin, & Douze, 2018). The method implies iterative clustering of deep features and using the cluster assignments as pseudo-labels to learn the parameters of the Сonvnet. This more complicated method achieved top results on the ImageNet dataset. Furthermore, due to the unsupervised nature of this type of model, it translates very well to other datasets, or used to build on work of Scheirer et al (2013) on open set recognition.

Conclusion

Due to the inherent difficulty of this dataset basic one-pronged approaches without heavy alteration and tailoring are extremely unlikely to perform well on the dataset. Due to time constraints we were unable to heavily tune or perfect the approach of any of the models presented. As a result, our models underperform. The EfficientNet could be improved by fine tuning hyperparameters and model parameters, along with the changes to approach described in the evaluation section. The SVM could be improved by adding some sort of feature extraction tool like SIFT and SURF. The BNN could be improved with input filtering. However, it is the recommendation of this report to avoid BNN for any real future work. Similarly given the current state of research, it is recommended that SVM also not be used for a dataset of this nature without due cause. As stated previously with regards to understanding if it is possible for a general machine learning user to be able to reproduce the work of state-of-the-art models from 2006, this report has been unable to substantiate this claim. But it is the opinion of the report that with further investigations and tuning the EffiicentNet CNN could be capable of achieving this outcome.

# References

Boiman, O., Shechtman , E., & Irani, M. (2008). In defense of Nearest-Neighbor based image classification. *IEEE Conference on Computer Vision and Pattern Recognition.* Anchorage: IEEE. doi:10.1109/CVPR.2008.4587598

Cao, J., Wu, C., Chen, L., Cui, H., & Feng, G. (2019). An Improved Convolutional Neural Network Algorithm and Its Application in Multilabel Image Labeling. *Computational Intelligence and Neuroscience*. doi:10.1155/2019/2060796

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep Clustering for Unsupervised Learning of Visual Features. *European Conference on Computer Vision.* Munich: Facebook.

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 Object Category Dataset.* Pasadena: California Institute of Technology .

Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., . . . Huang, T. (2011). Large-scale image classification: Fast feature extraction and SVM training. *CVPR 2011.* Providence: IEEE.

Mehboob, F., Abbas, M., & Rauf, A. (2018). Object Recognition Using SVM Based Bag of Combined Features. *SAI 2018: Intelligent Computing* (pp. 497-510). London: Springer.

Ramya, P., & James , A. (2019). Object Recognition and Classification Based on Improved Bag of Features using SURF AND MSER Local Feature Extraction. *1st International Conference on Innovations in Information and Communication Technology (ICIICT)* (pp. 1-4). IEEE.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., & Boult, T. E. (2013). Toward Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(7), 1757 - 1772. doi:10.1109/TPAMI.2012.256

Tan, M., & Le, Q. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.*

Tropea, M., & Fedele, G. (2019). Classifiers Comparison for Convolutional Neural. *IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT).* Cosenza: IEEE.

Yan, H., & Lui, X. (2016). Image Classification via fusing the latent deep CNN feature. *ICIMCS'16: Proceedings of the International Conference on Internet Multimedia Computing and Service*, (pp. 110-113).

Appendices

|  |  |  |  |
| --- | --- | --- | --- |
| Group Member | Contribution | Contribution % | Digital Signatures |
| Alex Conroy | Research, Troubleshooting, Report writing | 33.3 | ALEX CONROY |
| Nebojsa Ajdarevic | Troubleshooting, Research, Report writing | 33.3 | NEBOJSA AJDAREVIC |
| Jordan McCallum | Modelling, Visualisations | 33.3 | JORDAN MCCALLUM |

*Table 2 Contributions*