OBJECT RECOGNITION

Catalin Alexandru Mihalache

986965

Department of Computer Science
Swansea University

December 1, 2021

Introduction

Object recognition has increasingly become a new field of interest in machine learning over the last few years. It has been investigated to realize automatic assembly or inspection [1]. The main problem is making an efficient machine learning algorithm that can generalize well on various objects [1]. This leads to significant computation problems that can affect making more intelligent machines as the requirements are hard to meet [2]. Neuro-science has declared this as an open and critical complication for its studies. In this context, artificial neural networks were introduced to combat this obstacle. Those are computational networks based on the brain and nervous system [3], so they mimic human thinking when classifying objects. There have been breakthroughs in image labeling, object detection, scene classification [4], areas reported by different researchers worldwide. An application of this area, convolutional neural networks (CNN), has shown a performance breakthrough in object detection and scene classification [5]. Exceptional results have been demonstrated on large datasets, such as ImageNet, and medium datasets, such as CIFAR-100 [6]. Feature extraction is a crucial step of such algorithms. Feature extraction from images involves extracting a minimal set of features containing many objects or scene information from low-level image pixel values, therefore, capturing the difference among the object categories involved [7].

This experiment compared the effectiveness of a simple classification method, such as K-means, and the more advanced CNN approaches. Throughout the study, prediction accuracy and validation loss will be the metrics for CNNs, while for K-means only the accuracy. We will see if a more complex and demanding computational solution is worth a more simplistic and more straightforward approach. Due to the handmade CNNs, we will also discuss how a complicated model can improve the accuracy, such as DenseNet [8].

Methodology

For this study, feature extraction has only been used for K-means as the processing for CNNs is done by the convolutional layers using different size kernels. CIFAR-100 was the dataset used, with the test being run on both classes and superclasses of the dataset.

K-means

K-means algorithm is often used for data mining, compression, probability density estimation, and many others [9]. The most important task is to find the best value for k. In our approach, we have decided to go for the number of classes of each set. For example, k was 100 and 20 for fine labels, respectively for coarse labels. Before applying the algorithm, we had to preprocess the data. Feature extraction was used to help distinguish each object in the dataset. After the program finished the processing, the data had to be fit before predicting the actual accuracy results.

CNNs

Our approach on the classic CNNs had two convolutional layers, followed by a 2*2 max-pooling layer. For both layers, the filter size at the start was 16, with the kernel size being (2,2). While advancing in our study, we increased the filter size from 16 to 32, up until 128, and the final kernel size was (3,3). All of the convolutions use ReLU activation for consistency, with softmax for the last layer. The dropout, which helps avoid overfitting, was 0.5, while at the start, the program used 0.25.

Training

Adam optimizer has been used for each model and runs for 50 epochs. To reduce overfitting on the model, we have used EarlyStopping [10]. The value monitored by the system was validation loss, as it is the value that shows if our model is overfitting. The patience was set to three. This means that if the validation loss for epoch x is not smaller than the validation loss for epoch x-3, then the training stops. This also helps in reducing the computation power used, as it was already demanding for some machines. The validation data was the testing set, which means that the model is training in relation to it. The dataset contains 50k training images and 10k images for testing. Data augmentation was not used for this experiment. Furthermore, the models used a validation split of 20%.

Classification Results

The primary evaluation metric we will be looking at is the percentage of correct predictions made by the model. We compared our results with a benchmark of 39.43% accuracy for super-classes and 24.49% for fine classes.

K-means

As expected, the accuracy of k-means is not as high as someone would like to recommend its use in real-life usage. Because the classification is purely made on the data set, and thoroughly thinking, like a human brain, is not implemented, leading to low accuracy results as shown in Table 1.

K-means	Accuracy
fine labels	0.62%
coarse labels	7.07%

Table 1 – k-means accuracy

CNNs

For each CNN model tried, it has outperformed the fine label benchmark. As shown in Table 2, the model's accuracy is directly proportional to the filter and kernel size. The maximum accuracy of this model was 44% which is almost double the benchmark performance. Validation loss was 2.21.

Accuracy	Number of layers	Layer size	Kernel size	Dropout
34%	2	16	2,2	0.25
37%	2	32	2,2	0.5
40%	2	64	3,3	0.5
44%	2	128	3,3	0.5

Table 2 – CNN accuracy on fine labels

To highlight the model's effectiveness for the fine label set, we plotted the training and validation accuracies and losses in Figure 1 alongside the confusion matrix in Figure 2.

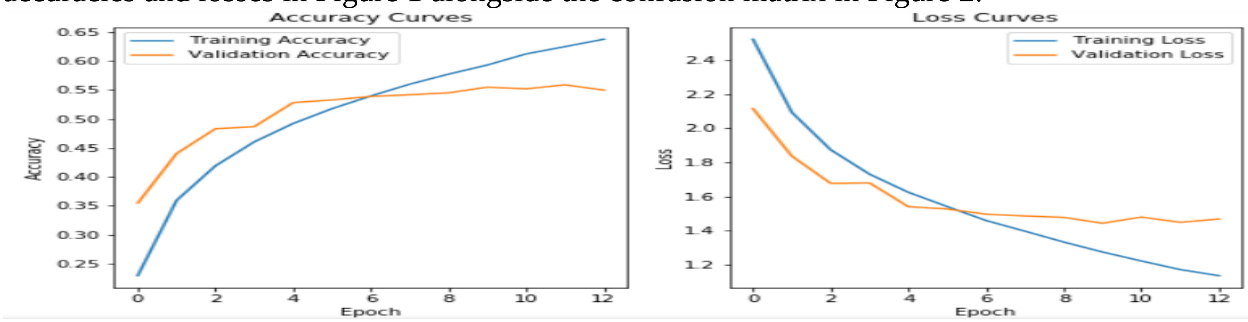

Figure 1 – accuracy and loss curves of CNNs for fine labels

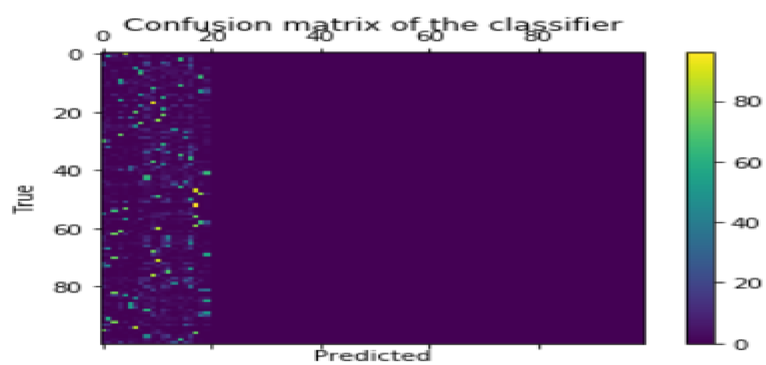

Figure 2 – confusion matrix of CNNs for fine labels

The same case applied to the benchmark of coarse labels. The accuracy of our model was 20% better than the benchmark. The relation is still directly proportional, however as the difference between 64 and 128 filters was almost insignificant, we have decided to keep only 68 filters to reduce the computation power. Validation loss for this model is 1.47.

Accuracy	Number of layers	Layer size	Kernel size	Dropout
40%	2	16	2,2	0.25
49%	2	32	2,2	0.5
56%	2	64	3,3	0.5
57%	2	128	3,3	0.5

Table 3 - CNN accuracy on coarse labels

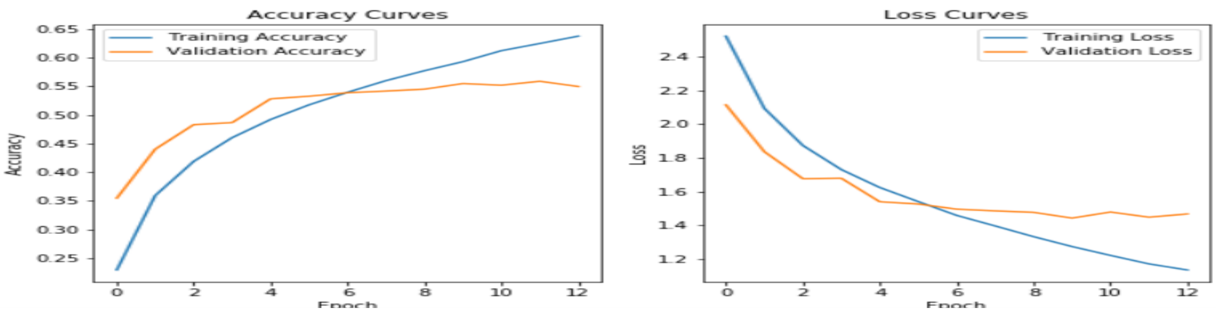

Figure 3 - accuracy and loss curves of CNNs for coarse labels

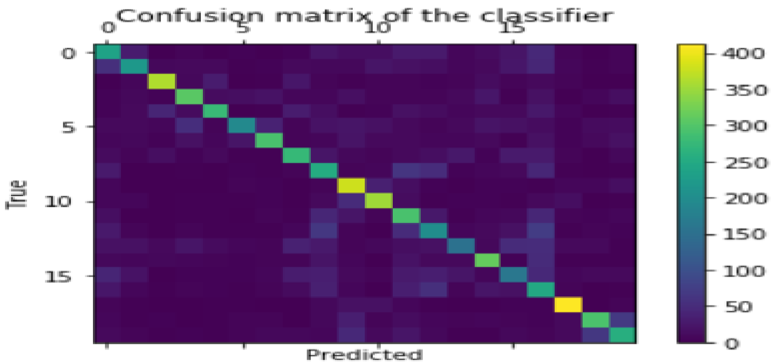

Figure 4 – confusion matrix of CNNs on coarse labels

Conclusion

The methods were all run on just one machine, a personal laptop, with decent processing power, but the truth may be that rerunning this on a different system with higher volumes of power would set up better results. From this study, one can extract that simplified algorithms do not work on complex problems such as object recognition. In the end, the Neural Network solution appears to be the most promising, and exploring the potential of more complex Convolutional Neural Networks [8] could turn out to be more accurate outcomes if researched rigorously. Minor things, such as specifying the correct learning rate [11], might also play a role in further improving the accuracy.

References

[1] M. Oshima and Y. Shirai, "Object Recognition Using Three-Dimensional Information," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 4, pp. 353-361, July 1983, doi: 10.1109/TPAMI.1983.4767405.

[2] Poggio, Department of Brain and Cognitive Sciences, McGovern Institute for Brain Research, Center for Biological and Computational Learning, “Models of object recognition”, 2000, Pages 1546-1726, https://doi.org/10.1038/81479

[3] Steven Walczak, Narciso Cerpa, Artificial Neural Networks, Editor(s): Robert A. Meyers, Encyclopedia of Physical Science and Technology (Third Edition), Academic Press, 2003, Pages 631-645, ISBN 9780122274107, https://doi.org/10.1016/B0-12-227410-5/00837-1

[4] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V.,& Garcia-Rodriguez, J. (2017) “A Review on Deep Learning Techniques Applied to Semantic Segmentation.”

[5] Wang, Y., & Wu, Y. “Scene Classification with Deep Convolutional Neural Networks.”

[6] Sultana, F., Sufian, A., & Dutta, P. (2019). Advancements in Image Classification using Convolutional Neural Network. Arxiv.org. https://doi.org/10.1109/ICRCICN.2018.8718718

[7] Neha Sharma, Vibhor Jain, Anju Mishra, An Analysis Of Convolutional Neural Networks For Image Classification, Procedia Computer Science, Volume 132, 2018, Pages 377-384, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.198

[8] Huang, G., Liu, Z., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. ArXiv.org. https://arxiv.org/abs/1608.06993

[9] Greg Hamerly, Charles Elkan, Learning the k in k-means, Department of Computer Science and Engineering University of California, San Diego La Jolla, California 92093-0114.

[10] “Tensorflow”, Oracle, accessed December 1, 2021, https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping.

[11] “Setting the learning rate of neural network.”, Jeremy Jordan, accessed December 1, 2021, https://www.jeremyjordan.me/nn-learning-rate