Paper Review

# A Comprehensive Study on Deep Image Classification with Small Datasets

Chandrarathne, Gayani and Thanikasalam, Kokul and Pinidiyaarachchi, Amalk
*Advances in Electronics Engineering, pp. 93-106. Springer, Singapore, 2020*

Reviewed by Nicholas Greene - ngreen29
Project 5: Vision Guided Mosquito Dissection for the Production of Malaria Vaccine
EN.601.656 CIS II Spring 2021

3/10/21

## Contents

## 1 Introduction and Relevance

The goal of Project 5: "Vision Guided Mosquito Dissection for the Production of Malaria Vaccine" is to develop several vision based methods for a robotic mosquito dissection system. The system is designed to automatically extract the salivary glands from mosquitoes, an important step required for the production of malaria vaccine.

Particularly, two out of the three major vision tasks of the project are to develop image classification solutions using both a classical image processing approach, and also a deep learning based approach. Both classification tasks involve determining if mosquito remains, which are left behind during multiple stages of the salivary gland extraction process, were successfully cleaned by an associated cleaning process. Figure 1 contains an example image for one of the cleaning tasks. Using both classical and deep learning methods for the same task allows the outputs to be cross-verified, resulting in greater overall confidence in the

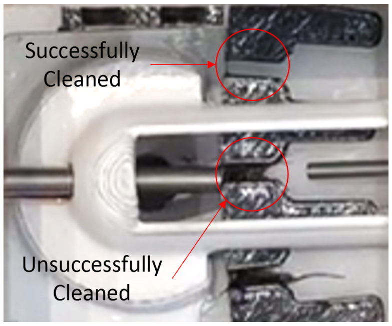


Figure 1: An example image from one of the cleaning tasks for the mosquito dissection system which is being developed at JHU

results. The success of these tasks is critical for maximizing the speed of the system, as well as for being a reliable metric when analyzing performance of the system after any changes, as well as for analyzing failure cases and frequency.

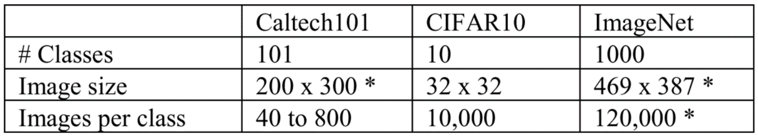
A major challenge for the development of the deep learning implementations is that there is a very small amount of training data, only on the order of a few hundred labeled training examples. Historically, the success of deep networks is associated with the number of training examples. There have been advancement, however, in achieving good performance with reducing data and reduced training times. The chosen paper *A Comprehensive study on deep Image Classification with Small Datasets* [1] paper investigates deep learning classification approaches in order to obtain better results with little training data.

# 2 Summary

## 2.1 Introduction and Background

The authors first provide the current context for deep learning with small datasets. They describe how Deep Convolutions Neural Networks are the most effective general solution for image classification and how Deep learning on large datasets is capable of achieving superhuman performance in visual recognition. They highlight that poor generalization due to both underfitting and overfitting is a major hurdle which arises when using deep learning on small datasets. To elaborate, a large deep network with millions of parameters can easily overfit to a small dataset, however a smaller model might not be flexible enough to model task adequately, resulting in underfitting. The limitations from using a smaller model to prevent overfitting on a small dataset means that state of the art performance can not be achieved.

The authors then explain this underfitting vs overfitting problem in greater technical detail: The depth of Convolutional Neural Netowrks (CNN) particularly the feature hierarchies that they extract are where their predictive power resides. Particularly, the success of networks like VGG made it clear that the depth of a network is critical to its performance [7]. This was further demonstrated after the success of ResNets, which were relatively simpler, but substantially deeper than previous networks [3]. In order to reduce the number of parameters in a model to prevent overfitting to a small dataset, the number of layers must decrease which inherently lowers the model's generalizability. The authors then briefly men-



| | Caltech101 | CIFAR10 | ImageNet |
|---|---|---|---|
| # Classes | 101 | 10 | 1000 |
| Image size | 200 x 300 * | 32 x 32 | 469 x 387 * |
| Images per class | 40 to 800 | 10,000 | 120,000 * |

Figure 2: Dataset Comparison

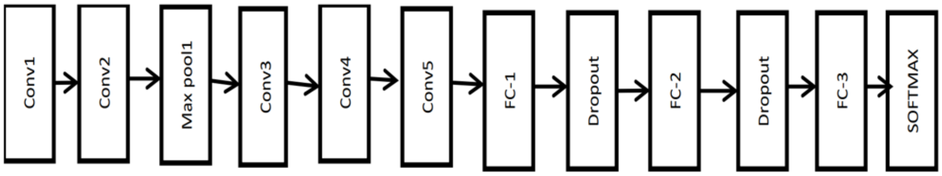


Figure 3: Best architecture for training on CIFAR10 from scratch

tion data augmentation before moving on to discuss transfer learning. Transfer learning is where models trained on large scale datasets such as Image Net can have their early layers reused in a new learning since the features learned in the earlier layers are more generic filters.

## 2.2 Experiments

The authors performed two main experiments using two small datasets, CIFAR10 and Caltech101. Caltech101 has 9,146 images and CIFAR10 has 100,000 images. See figure 2 for more details about the datasets [4] [6]. The ImageNet dataset mentioned in the figure was used for transfer learning which is described later [2]. For all models, the optimizer was Stochastic Gradient Descent with a learning rate was 0.001.

The first experiment studies how the number of convolutional layers in a deep learning architecture affects the predictive performance when training on a small dataset from scratch. For this they modify the VGG-16 architecture by removing a number of the convolutional layers from the end. The final convolution output is fed into three fully connected layers of size 512, 256 and the number of classes in the dataset, respectively. A total of eight networks with between 2 and 9 convolutional layers are trained on both the CIFAR10 and Caltech101 datasets. Max pooling layers are used in between the convolutional layers, and the fully connected layers use ReLU activation and dropout. An example of one of the architectures can be seen in figure 3.

The second experiment studies how the number of convolutional layers which are reinitialized and trained, rather than remaining ”frozen”, affects performance during transfer learning. The base network is VGG-16 pre-trained on ImageNet, where the last convolutional layer output is fed into three fully connected layers, identical to what was previously described. Finally, for each dataset, the transfer learning approch with the best performance was retrained one final time where the previously ”frozen” layers are also trained without re-initializing, and with 1/10th the learning rate. The VGG-16 layers which are used can be seen in figure 4

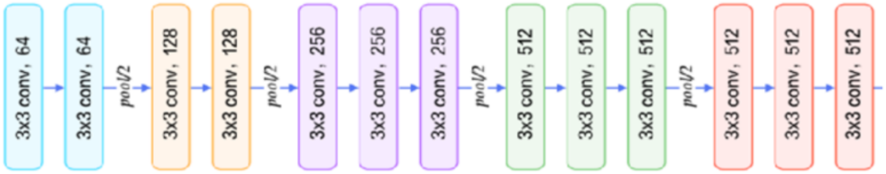


Figure 4: VGG-16 Convolutional layers. The kernel size and number of neurons are listed for each later.

## 2.3 Results

Figure 5 shows the results for training Caltech101 from scratch. Note that this is the smallest dataset, and the that only 4 convolutional layers was most effective.

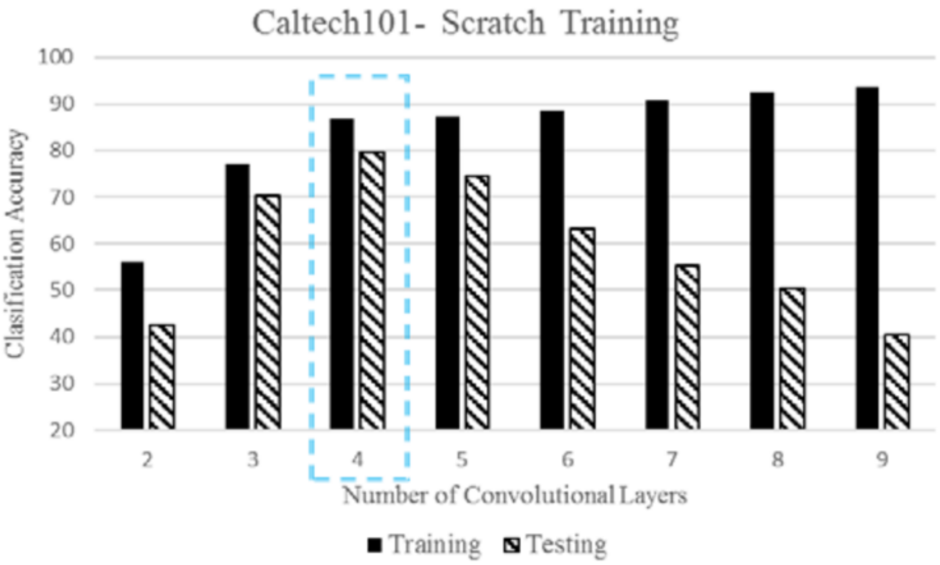


Figure 5: Train and test results of training on Caltech101 with from scratch versus the number of convolutional layers.

Figure 6 shows the results for training CIFAR10 from scratch. Note that this dataset is an order of magnitude larger than Caltech101, and it is most successful with an aditional convolutional layer. The association between dataset size and the number of parameters before overfitting occurs is clear here.

The test results from transfer learning with a different numbers of frozen layers is shown in figure 7. Not that once again, the overfitting problem is very clear here. Caltech101, an order of magnitude smaller than CIFAR10, performs best when retraining only the last two convolutional layers and the fully connected layers. The best results for CIFAR10 were for retraining the last five convolutional layers and the fully connected layers.

Finally, the result of unfreezing and training all of the parameters are shown in figure 8. Once again, it seems that the overfitting problem occurs for the smaller Caltech101 dataset due to the increased number of parameters, as performance goes down significantly. It does not occur for the larger CIFAR10 dataset where performance is improved.

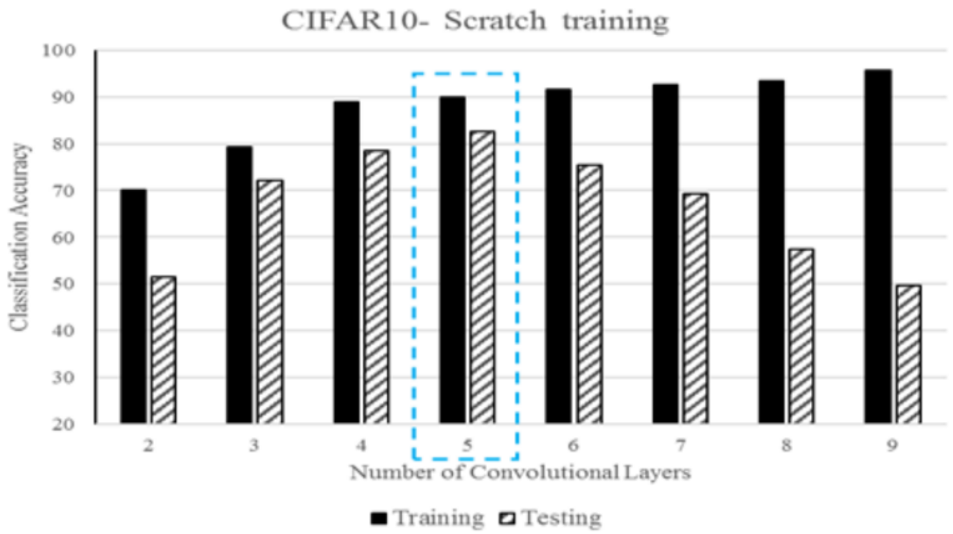


Figure 6: Train and test results of training on CIFAR10 with from scratch versus the number of convolutional layers. The best result is highlighted by the rectangle

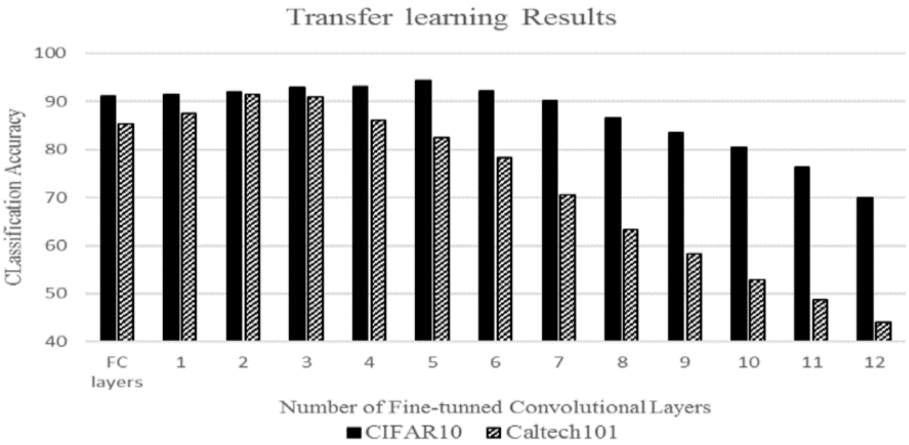
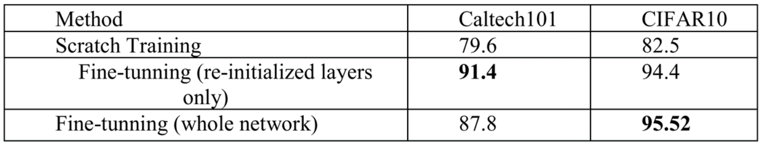


Figure 7: Transfer learning test accuracy for CIFAR10 and Caltech101 versus the number of unfrozen convolutional layers



| Method | Caltech101 | CIFAR10 |
|---|---|---|
| Scratch Training | 79.6 | 82.5 |
| Fine-tunning (re-initialized layers only) | **91.4** | 94.4 |
| Fine-tunning (whole network) | 87.8 | **95.52** |

Figure 8: Best performance for training from scratch, transfer learning, and transfer learning where no layers are frozen. Note that the latter case was only trained once per dataset, using the best transfer learning approach.

# 3 Critical Assessment

Overall I felt that this paper was a bit weak. I did not feel that I got as much out of it as the title *A Comprehensive Study on Deep Image Classification with Small Datasets* would suggest. There were a number of technical details which were never mentioned, such as what loss function was used, whether regularization was used, how the data was preprocessed, or the exact location the max pooling layers for all architectures. The biggest flaw, however, was that the paper did not cover enough scenarios, and it did not make use of many modern techniques which were considered standard practice by the time the paper was published in 2020.

I will provide some specific details to substantiate these critiques. Firstly, the authors only used one base model, VGG-16, for their experiments. Using a single model does not seem thorough enough to make any generalizable claims. Additionally, VGG-16 is about seven years old now, and state of the art performance on ImageNet is currently more than 15% better than VGG-16's performance. Some standard techniques which were not used are a better performing optimizer such as ADAM [5], batch normalization, or even data augmentation to name a few. The lack of at least a meaningful discussion for why data augmentation was excluded from the experiments is particularly egregious considering how much of a benefit it provides to small datasets. The paper did not attempt to quantify the relationship between the number of trainable parameters for each model and the size of the dataset. This would have been helpful in an application by providing a quantifiable estimate for how complex a model should be for a given small dataset. Finally, the authors did not address any of these points in a future work section.

I have some suggestions for additional future experiments which I feel would be particularly valuable. It would be interesting to run similar experiments for datasets on the order of 1000 and 100 training examples. The smallest dataset at approximately 10,000 images is still quite large for many applications, particularly in the medical field. Additionally, since there seems to be a limit on the number of parameters in a model in relation to the dataset size, it would be interesting to see how changing the input resolution of the images would affect performance. Downsizing the images could significantly reduce the number of parameters required for a given architecture, without sacrificing too much information. This could allow for deeper networks before the overfitting parameter limit is reached, potentially combating any underfitting which may occur due to the shallower networks required for smaller datasets. Finally, it would also be useful to experiment with removing parameters from the the fully connected layers rather than the convolutional layers. In fact, most of the parameters are in the fully connected layers. Removing just one fully connected layer could allow for a much deeper set of convolutions for the same total number of parameters. This might improve performance by reducing underfitting without increasing overfitting.

# 4 Conclusion

In conclusion, despite the critiques mentioned, this paper provides useful insight into how the number of convolutional layers affects performance on a small dataset for both training from scratch, and for transfer learning. This has direct applications for my project's deep learning image classification task as the dataset will only containt about 300 images. I will try using a substantially smaller number of convolutional layers, between 2 and 5, as performance could potentially be better, and I would not have initially tried using so few layers.

# References

[1] Chandrarathne, G., K. Thanikasalam, and A. Pinidiyaarachchi (2020). A comprehensive study on deep image classification with small datasets. In *Advances in Electronics Engineering*, pp. 93–106. Springer.

[2] Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[3] He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

[4] Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[5] Kingma, D. P. and J. Ba (2015). Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[6] Krizhevsky, A., G. Hinton, et al. (2009). Learning multiple layers of features from tiny images.

[7] Simonyan, K. and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.