Deep Learning - Reproducibility Study

DATA473 Assignment 2

Aarthy Sree Badrakalimuthu	Student ID:46238827
Raghuram Gudimetla	Student ID:59014141
Swapna Josmi Sam	Student ID:74281128

Introduction

Creating and training a deep or convolutional neural network from scratch would be a challenging task in terms of time as well as computational aspects. Thus, with the help of techniques such as fine tuning and transfer learning, most of the image recognition tasks are optimized using some pretrained networks such as ImageNet. Many researches have already conducted experiments to validate and observe the performance of these pretrained networks on new image classification tasks. This report recreates the results of some of the recent studies conducted to observe the performance of different neural networks that uses ImageNet models and understand their effects by varying the hyperparameters used.

There are three main topics the report focuses on recreating: one is dropout, Srivastava et al. [1] introduces dropout as a way to address the issue of overfitting a model, and thereby improving its performance on the test dataset. The second is on fine-tuning, Li et al. [2] have performed fine-tuning experiments to demonstrate the impact of hyperparameters on different networks using pretrained ImageNet model (on varied datasets). In this report, we have recreated experiments by varying momentum hyperparameter on $ResNet$ network using Stanford Dogs dataset and Caltech Birds datasets. The final experiment, talks about transfer learning. Kornblith et al. [3] have performed experiments to analyse the correlation between transfer accuracy from fixed feature extraction and fine-tuning using ImageNet weights against ImageNet Top-1 accuracy. We have recreated feature extraction experiment using Stanford Cars dataset and fine-tuning using Caltech101 dataset. In all the above-mentioned experiments, the results obtained are studied and compared with the corresponding results from the papers.

Datasets Used

The experiments were carried out with the same datasets as mentioned in the paper. Some of these datasets were directly available from $Keras$ or $TensorFlow$ library, while some others were downloaded from $Kaggle$. The following are the datasets used:

MNIST Dataset

This dataset was imported from the $Keras$ library and it consists of images of 10 digits (0 to 9) corresponding to the 10 classes. $60{,}000$ images are available in the training dataset and $10{,}000$ images in the test dataset. Dimensions of the images are 28×28 pixels in greyscale. This dataset was used in the dropout experiment.

Stanford Dogs Dataset

This dataset from $Kaggle$ has 120 breeds of dog's images from around the world and belongs to fine-grained image recognition category. There is a total of 20580 images, of those the authors have used 12000 images for training and the rest 8580 as test dataset, while conducting fine-tuning experiment. Each image is of varied pixel size. The percentage of images used for validation, were not mentioned by the authors. Hence, of the 12000 images in training set, 10% were used as validation set (1200 images).

Caltech Birds Dataset

This dataset has been imported from $TensorFlow$. This consists of 200 species of fine-grained bird images i.e. 200 classes. Training dataset consists of 3033 images, while test data consists of 3000 images. Dimension of these images are 500×500. Due to infrastructural constraints, only 20% of the training data and 10% test data has been used to reproduce the results. This data has been used to recreate the impact of hyperparameters while fine-tuning ImageNet.

Stanford Cars Dataset

This dataset from $Kaggle$ has 196 classes of car images and belongs to fine-grained object recognition category. The total number of images are 16185 of varied pixel sizes, of those the authors have used 8144 images for training and the rest 8041 as test dataset for fixed feature transfer learning experiment. Here, while all 8144 images were used for training, only $2\% \sim 160$ images were used for testing due to environmental constraints.

Caltech-101 Dataset

This dataset from $TensorFlow$, consists of pictures of objects belonging to 101 classes, plus one background clutter class. Each image is labelled with a single object. Each class contains roughly 40 to 800 images, totalling around $9k$ images. The training dataset consists of 3060 images and test data consist of 6084 images, with a dimension of 377×604. Due to infrastructural constraints, only 10% of the training data and 5% test data have been used to reproduce the results. This data has been used to reproduce the correlation between ImageNet and transfer accuracy.

For datasets available in $TensorFlow$ or $Keras$ library the experiments were conducted using $Google\ Colab$, else the experiments were conducted using $Kaggle$ notebook wherein the datasets can be imported directly.

Methods

The following are the three major categories of experiments conducted: comparison of performance of a network with and without dropout, influence of momentum hyperparameter while fine-tuning a network initialized with $ImageNet$ weights and comparison of transfer accuracy using fixed feature extraction and transfer learning/fine-tuning with Top-1 ImageNet accuracy.

Dropout

Overfitting a model results in high a generalization error when it is used for predicting new unseen data. Srivastava, has solved this issue by randomly dropping units in the neural network during the training process [1]. He has shown that this approach improves the performance of the model significantly better in comparison with other regularizing techniques. Here, an experiment performed using MNIST dataset is recreated.

Experiment with and without Dropout

According to the research paper MNIST dataset was trained on a standard neural network consisting of 2 hidden layers with 800 neurons in each layer. Logistic activation function is used on the hidden layers. This is compared with the performance of a network with dropouts incorporated. Here, the same dataset is trained on a network with 3 layers consisting 1024 units each. The activation function used here is Logistic, but additionally, dropout is implemented with a dropout rate of 0.8 on the input layer and 0.5 on hidden layers.

The above-mentioned experiments from the research paper were conducted. However, some of the parameters such as batch size, epochs, learning rate, and optimizers that were used were not clearly mentioned in the paper. These parameters have been used on the basis of technical feasibility of the platform ($Google\ Colab$) used for training. A batch size of 32 for 10 epochs, with SGD optimizer were used to recreate the experiment. Dropout rates were followed according to the values given in the paper. Pre-processing steps of the dataset includes rescaling the pixel size between 0 and 1, and encoding the categorical variables using one-hot encoding.

Fine Tuning

Training the images from scratch is a tedious process, hence the norm is to initialize pre-trained weights from ImageNet for a CNN and further training is performed by fine-tuning only certain hyperparameters. In this section, fine-tuning experiments conducted by Li [2] in his paper are recreated and the results obtained are compared.

Fine-Tuning: Varying Momentum with Fixed Learning Rate

One of the experiments performed by the Li [2] in his paper, is to understand the effect of momentum on domain similarity. He argues that from literature, most authors have used a momentum of 0.9 while fine-tuning the network. This value is unchanged in most studies, irrespective of the network architectures or the target classification task. Hence, in his paper he has conducted a study to check if indeed a moment of 0.9 outperforms in comparison to zero momentum for a variety of datasets, using pre-trained $ImageNet$ weights. This comparison is done calculating the validation error, and the results from the paper, show certain datasets perform better with higher momentum and while certain others with zero momentum. We have recreated this experiment using Stanford Dogs dataset and Caltech Birds dataset.

Fine-Tuning Stanford Dogs Dataset

Since the percentage of validation dataset used is unknown, 0.1 percent of the train dataset is used, hence the $ResNet101$ is trained on 10800 images with rest of 1200 images are used for validation in each epoch. As shown in table 1, batch size was reduced to 4 and were run for 20 epochs, incrementally (due to environmental constraints). The pixel sizes of images in the dataset were resized as 224×224 to match the input requirements. The dataset was trained with a momentum of 0.9 and learning rate of 0.01 using SGD optimizer (optimizer used is not given in the paper) and the corresponding training accuracy, training loss, validation accuracy and validation loss are computed. The same experiment is repeated with a momentum of 0.0 and the parameters are computed once again.

Parameter	Li's Paper	Experiment Conducted
Network	$ResNet101$	$ResNet101$
Momentum m	0.9 and 0	0.9 and 0
Learning Rate η	0.01	0.01
Weight Decay λ	0	0
Batch Size	256	4
Epochs	300	20

Table 1: Parameters Used for Fine-Tuning Experiment using Stanford Dogs Dataset

Fine-Tuning Caltech Birds Dataset

Here, the performance of $ResNet101$ architecture on bird classifications are examined as given in the paper. The experiment was conducted on a reduced training dataset size of ~ 600 images and a test set size of ~ 300 images. The hyper parameters provided in the paper and the experiment conducted are as in table 2. Unfortunately, the experiment could not be replicated with 256 epochs considering infrastructure constraints. However, the learning rate, batch size and weight decay are the same as mentioned in the paper. This network structure was trained with multiple momentum values, i.e. 0.9 and 0.0 to evaluate the results. Pre-processing steps include rescaling the pixel size between 0 and 1 and encoding the dependent categorical variables.

Parameter	Li's Paper	Experiment Conducted
Network	$ResNet101$	$ResNet101$
Momentum m	0.9 and 0	0.9 and 0
Learning Rate η	0.01	0.01
Weight Decay λ	0	0
Batch Size	16	16
Epochs	300	50

Table 2: Parameters Used for Fine-Tuning Experiment using Caltech Birds Dataset

Transfer Learning

Transfer learning refers to extraction of features from the penultimate layer of a neural network (such as $ResNet$, etc.) and using these extracted features, for performing classification using conventional classification algorithms (such as Logistic Regression, SVM, etc.). These features can also be fine-tuned further on smaller fully connected networks.

In the paper by Kornblith [3] transfer learning performance is measured in three settings to assess if ImageNet Top-1 accuracy can be used as a predictor for transfer accuracy. These include: (1) training a logistic regression classifier on fixed feature extraction from the penultimate layer of the ImageNet pre-trained network, (2) fine-tuning the ImageNet pre-trained network, and (3) training the same CNN architecture from scratch on the new image task to compare. Here we have conducted experiments one and two.

Fixed Feature Extraction Experiment

In the paper by Kornblith [3], one of the experiments conducted transfer learning to perform a comparison of accuracy obtained as a result of feature extraction and applying logistic regression (fixed feature extraction) with $ImageNet$ Top 1 accuracy. The authors have tested this for 16 classification networks and have claimed that there is a strong correlation between these accuracies. Here, this experiment is performed on Stanford Cars dataset, using $ResNet$ and $MobileNet$, initialized with $ImageNet$ weights for fixed feature extraction. The images are resized as 224×224 pixels each. These features extracted are input to logistic regression model. As given in the paper, $L-BFGS$ solver with $L2$ Penalty is used for performing logistic regression (parameters used mentioned in table 3). The paper has not mentioned the number of iterations the experiment was run for, and if the input image pixels were scaled.

Parameter	Kornblith's Paper	Experiment Conducted
Network	$ResNet101$ and $MobileNet$	$ResNet101$ and $MobileNet$
Regularizer	$L2$	$L2$
Solver	L-BFGS	L-BFGS
Iterations Used	Not mentioned	10000
Weights	Initialized from $ImageNet$	Initialized from $ImageNet$
Pre-processing	Not mentioned	Scaled and resized to 224×224

Table 3: Parameters Used for Transfer Learning Experiment using Stanford Cars Dataset

Fine-Tuning Pre-trained ImageNet Network Experiment

This experiment was conducted to compare transfer learning accuracy by fine-tuning the ImageNet pre-trained network with Top-1 ImageNet accuracy. In the research paper, the authors examined the performance of $ResNet101$ architecture on Caltech object classifications. Batch size used here was 4 instead of 256 as given in the paper because of the infrastructure constraints. The other hyperparameters such as learning rate, weight decay was decided in the paper using grid search. In the current report, grid search was not feasible due to the huge dataset. Thus, the learning rate and weight decay used were 0.01 and 0, respectively. The optimizer used here was SGD. Penultimate layer of the network has been trained with 102 units classes and activation function used was $softmax$.

Results

The results from each of the experiments conducted, in comparison with the results from the paper are explained below.

Comparison of Dropout Results from Experiment and Paper

Using the dropout technique on MNIST images, have provided some interesting observations. Initially, MNIST dataset has been trained on the standard neural network using logistic regression without any dropout layers. Figure 1.1 explains the performance metrics of this model. As observed, the test error here was 0.072 i.e. 7.2%.

However, after adding dropout layers to this architecture still resulted in a test error of 0.07 i.e. 7% and did not significantly improve its performance. As seen in the Figure 1.2, there is only a slight increase in test accuracy which is almost negligible. The exact results were not reproducible for the test errors as stated in the research paper, which were 1.60% and 1.35%, for the networks without and with dropout respectively. However even in the paper, the margin of error difference between the models with and without dropout is very small.

Without Dropout

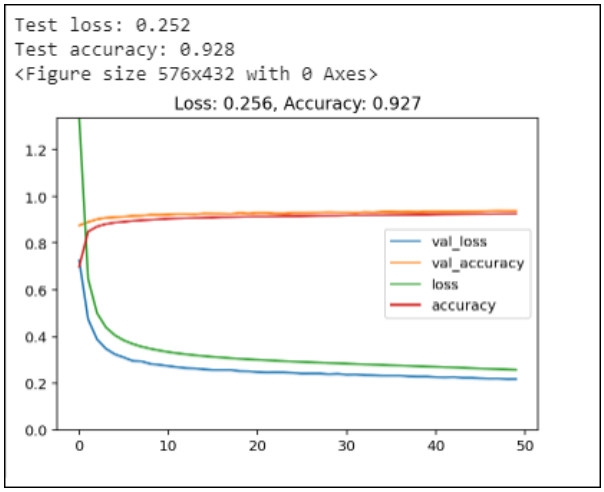

Figure 1.1: Loss and Accuracy Per Epoch

With Dropout

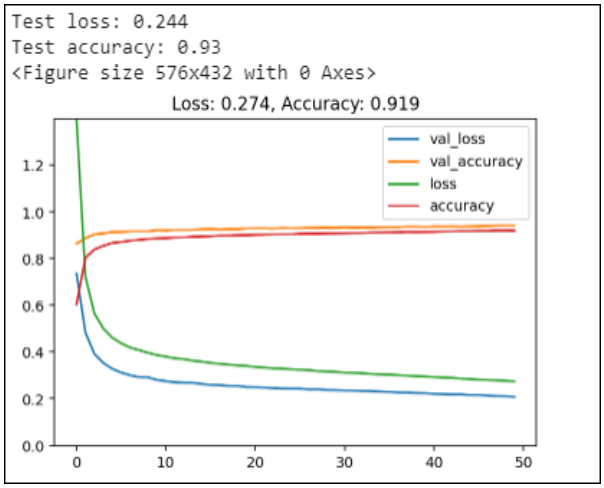

Figure 1.2: Loss and Accuracy Per Epoch

Fine Tuning Results for Stanford Dogs Dataset

The below graphs are plotted to represent the error, accuracy and loss per epoch. As seen from the plots in Figure 2, both the train and validation accuracy increase, as the loss reduces for each epoch trained in both experiments.

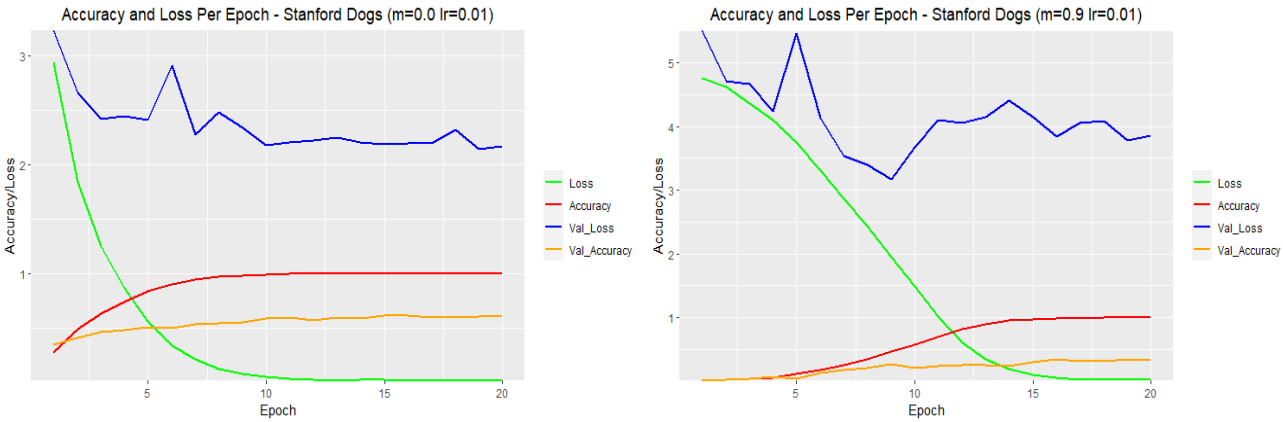

Figure 2: Accuracy and Loss as a Function of Epoch

Since the validation errors are compared in the paper, this along with the training errors are computed from the accuracy obtained and plotted for both scenarios with momentum of 0.9 and zero momentum as shown in Figure 3 below. It can be seen that with zero momentum, the validation error is less comparatively with a momentum of value 0.9

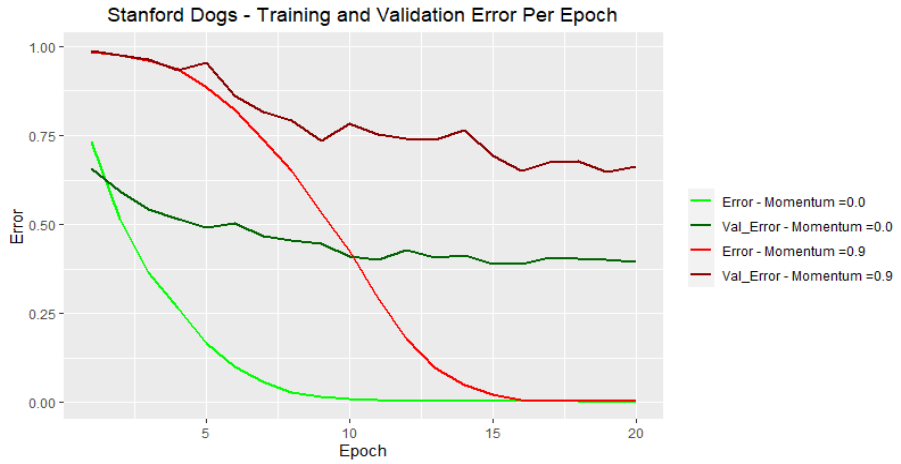

Figure 3: Train Error and Validation Error as a Function of Epoch

Comparison of Stanford Dog Fine-Tuning Results from Experiment and Paper

The table 4 below compares the result obtained from the paper and as a result of the experiment performed. Although the batch size was smaller and $ResNet101$ was trained for a small number of epochs (compared to the parameters from the paper), the results obtained does concur with that of the results from the paper, which is validation error with zero momentum is much smaller than the validation error with a momentum of 0.9. This implies not all training should be done with a standard value of momentum as 0.9, as in this scenario zero momentum works best.

STANFORD DOG DATASET	Result from Li's Fine-Tuning Paper	Result from Experiment Conducted
Validation Error with Momentum 0.0	10.87% after 300 epochs (256 batches)	39.5% after 20 epochs (4 batches)
Validation Error with Momentum 0.9	17.41% after 300 epochs (256 batches)	66.4% after 20 epochs (4 batches)

Table 4: Validation Error Comparison for Stanford Dog Dataset

Comparison of Caltech Birds Fine-Tuning Results from Experiment and Paper

Experiment with a momentum of 0.9 and zero have been used to observe its effectiveness on the training data. Figure 4.1 provides the variation in errors for both. With zero momentum, there was a sharp decrease of error within a few initial iterations. The decrease in error was stable throughout the iterations given the momentum value of zero. It was interesting that the results observed in the current experiment are almost similar to the actual results of the research paper. Figure 4.2 describes the actual results observed using the same parameters. From the experiment, the model with zero momentum outperforms model with a momentum of 0.9 and concurs with the results obtained from the paper as well.

Experimental Observations	Actual Observations
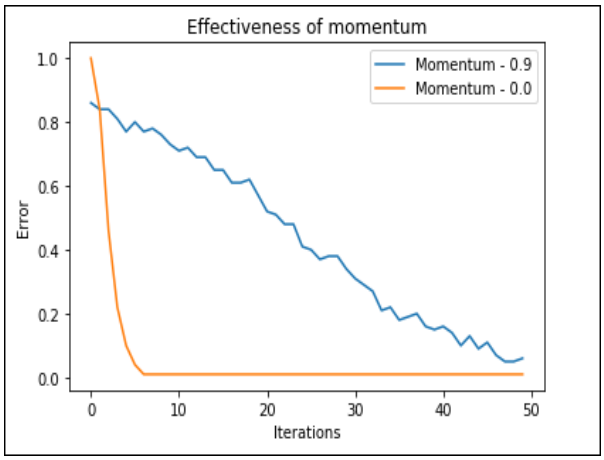	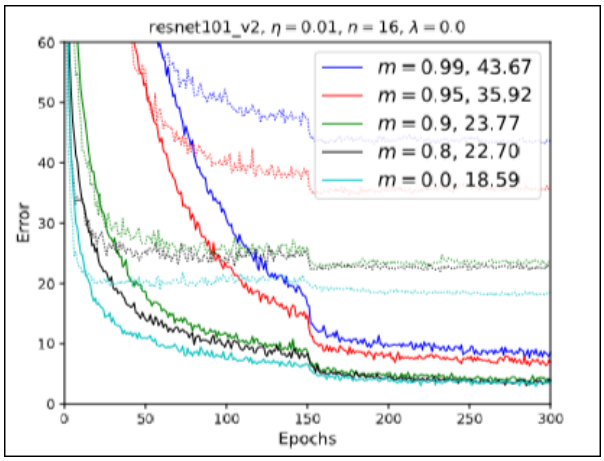
Figure 4.1: Error Per Epoch	Figure 4.2: Error Per Epoch

Transfer Learning Fixed Feature Extraction Results

The experiment was performed using $ResNet101V2$ and $MobileNetV1$ with the final layer removed, and the weights were initialized using $ImageNet$. The scaled pre-processed 224×224 extracted features are trained with logistic regression classifier with $L-BFGS$ solver and $L2$ Regularizer for 10000 iterations.

Comparison of Results from Experiment and Paper

The accuracy for $ResNet$ and $MobileNet$ obtained from experiment are slightly less in comparison with that of the paper (table 5). This could be as a result of smaller subset of test data used (only 2% i.e. ~160 images from 8041 images), also there are unknown parameters such as number of iterations performed or if the images were pre-processed by the authors. However, as given in the paper the experiment shows $MobileNet$ transfer accuracy is slightly less in comparison with the $ResNet$.

NETWORK	Result from Kornblith's Transfer Learning Paper	Result from Experiment Conducted
ResNet	Accuracy: 59% (From Figure 2 in page 2664) -iterations not mentioned	Test Accuracy: 49.89% Train Accuracy: 99.85% -after 10,000 iterations
MobileNet	Accuracy: 53% (From Figure 2 in page 2664) -iterations not mentioned	Test Accuracy: 46.58% Train Accuracy: 99.85% -after 10,000 iterations

Table 5: Transfer Accuracy Comparison for Stanford Cars Dataset

Figure 5, shows the transfer accuracy in comparison with Top-1 ImageNet accuracy.

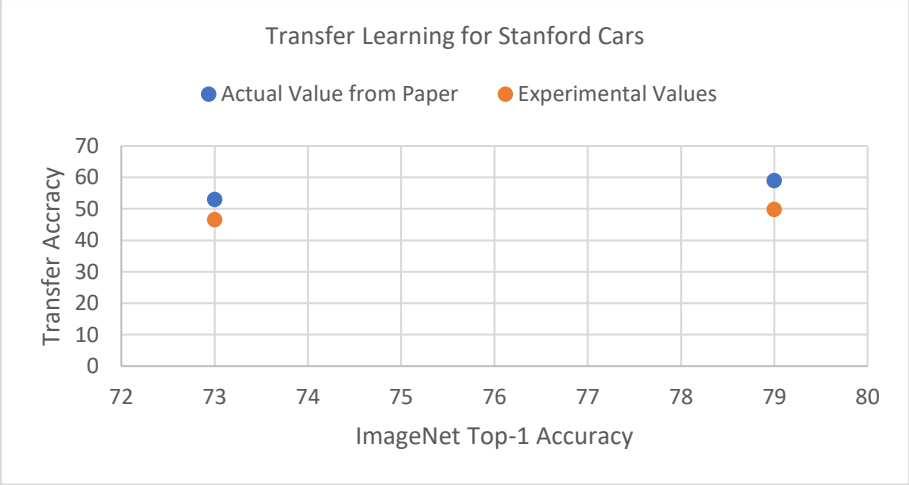

Figure 5: ImageNet Accuracy as a Predictor for Transfer Accuracy

Fine-Tuning Transfer Accuracy Comparison of Results from Experiment and Paper

As seen from table 6, the accuracy after 28 epochs was 82%. This was less when compared to the actual results, as we were unable to perform a large number of epochs. This variation could also be due to the inability to perform the grid search to obtain the optimal values for learning rate and weight decay. Nevertheless, the result obtained was comparatively similar to the original experiment as reported. Figure 6, shows accuracies transfer (orange bar), fine-tuning and random initialization accuracies (from paper), while the bar chart on the right the experimental result.

NETWORK	Result from Kornblith's Transfer Learning Paper	Result from Experiment Conducted
ResNet	Accuracy: 92% (From Figure 6 in page 2666) -iterations not mentioned	Accuracy: 82.07%

Table 6: Transfer Accuracy Comparison for Caltech101 Dataset

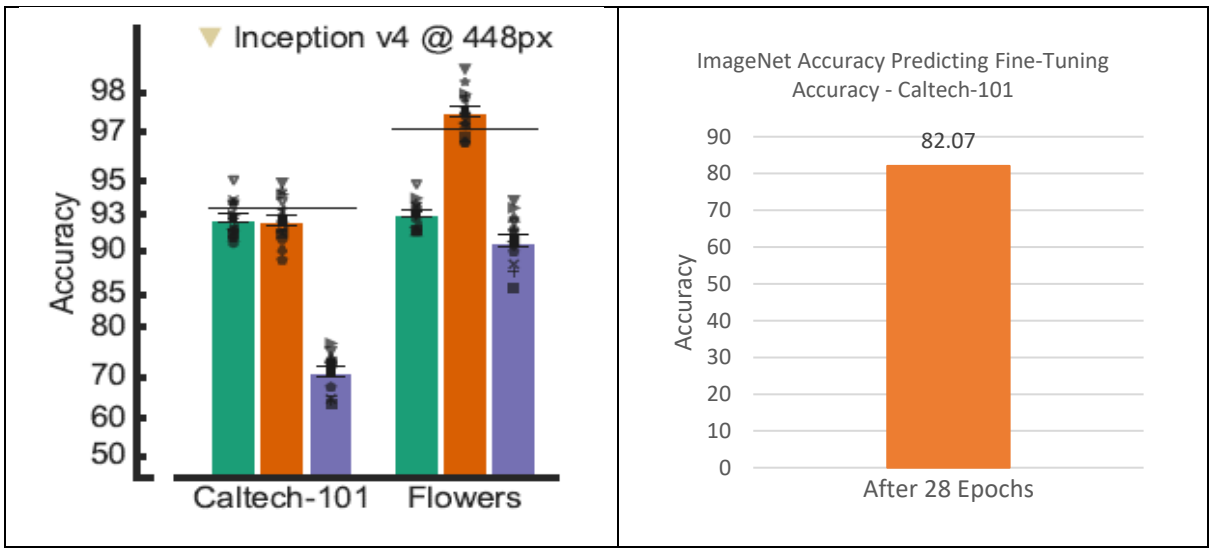

Figure 6: Comparison as a Predictor for Transfer Accuracy

Discussion

In almost all the experiments, the results obtained were not exact as those discussed in the papers. This is largely as a result of environmental constraints, since performing experiments with a complete dataset or iterating for a large number of epochs were infeasible. Also, in some of the experiments conducted, the parameters were not explained in detail. For example, in case of dropout experiment, there was no clarity about the number of epochs, learning rate and optimizer used. As a result, there was a discrepancy in the results. Also, the optimizer used for fine-tuning, the number of iterations for logistic regression in case of transfer learning were not provided. Also, in some cases the authors have used parameter grid to obtain better accuracies, which could not be verified due to environmental constraints.

However, the results in general appear to concur with the findings from all 3 papers. In case of dropout experiment, the network with dropout, had a low error (although very small) in comparison with network without dropout. In fine-tuning experiment, as stated by the authors Li [2], it is not advisable to have a momentum of 0.9 always, as the hyperparameters are dataset dependent and sensitive to source and target domain. With transfer learning experiment, the transfer accuracy varies with using different networks for the same extracted features and appears in-line with the ImageNet Top-1 accuracy, indicating it ImageNet accuracy could be used as a predictor for transfer network's performance as suggested by Kornblith [3].

Conclusion

Different experiments for dropout, fine-tuning and transfer learning were recreated. These experiments were conducted for different datasets using different networks, by varying their hyperparameters. Their performance, in terms of accuracy and loss were measured. These results were compared with the original experiments performed by the authors. In general, the inference based on the results obtained from the experiments does concur with those of the authors.

References

[1] Srivastava, Nitesh, et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting" 2014 *Journal of Machine Learning Research*. no. 15, pp. 1929-1958.

[2] Li, Hao, et al. "Rethinking Hyperparameters for Fine-tuning" *2020 International Conference on Learning Representations (ICLR)*, 2020, pp. 2661-2671.

[3] Kornblith, Simon, et al. "Do Better ImageNet Models Transfer Better?" *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, doi:10.1109/cvpr.2019.00277.