RESEARCH ARTICLE OPEN ACCESS

# Towards Guidelines for Assembling Image Datasets for Transfer Learning Techniques

Elinda Kajo Meçe*, Evis Trandafili*,Sibora Theodhor*
**Department of Computer Engineering, Faculty of Information Technology, Polytechnic University of Tirana, Albania*
*Corresponding Author; Elinda Kajo Meçe*

**ABSTRACT**
With the latest breakthroughs in deep learning, the field of machine learning for image recognition has been attracting increasing attention. Neural networks for image similarity and image classification problems have yielded impressive results. Another successful breakthrough on the field is the application of transfer learning, to reduce training time. Collecting data and preprocessing datasets is the most expensive task of the process. Our work aims at observing the transferability of the dataset characteristics and outlining guidelines for the effort of image data collection on a transfer learning scenario.
**Keywords -** Machine Learning, Transfer Learning, Image Recognition, Classification, CNTK, ImageNet

---

DATE OF SUBMISSION: 27-12-2019 DATE OF ACCEPTANCE: 15-01-2020

---

## I. INTRODUCTION

In the last few years machine learning and the field of computer vision have made great progress. Models used to recognize images like neural networks were first described by Patterson, Dan W. in 1930 [15] but it's only in the last four or five years that the hardware advancements like GPU utilization have made it feasible to effectively run and discover their full potential.

This evolution of the hardware combined with several breakthroughs in the field of computer vision, like Krizhevsky, Sutskever, and Hinton [2] have laid out the path for tackling new challenges in the field of image recognition and image classification. Deep learning and models like convolutional neural network (CNN), has proven to achieve good performance in difficult visual recognition tasks - matching the human performance in some fields [17].

The layer of neurons in the CNNs are designed similarly to how vision in animals and humans works. One of the challenges of scaling such techniques is the hardware requirements. Toolkits like Theano, Tenserflow, CNTK are making use of GPU programming and leveraging the highly parallel computational architecture of the GPUs to parallelize the many matrix operations utilized on various computer vision techniques.

The need to overcome the high costs sacrificing little accuracy spawned the efforts that produced the next breakthrough: transfer learning [22].

The transfer learning approach is to get a pre-trained model (parameters and weights of an NN that has been trained with large dataset by others) and regulating the model with your dataset in the last layers [1]. By using transfer learning a significant amount of labeling effort can be saved, reducing the data gathering cost and most importantly saving a lot of training time.

Similar work has been conducted by [Bengio, Yosinski] who explored and indicated that the features of image recognition and classification techniques are transferable.

On our paper, we compose a set of experiments to investigate how two characteristics of datasets: the number of samples, and the ratio of sample affect precision and accuracy on image classification through transfer learning.

We generate several datasets of images, that are subclasses of ImageNet and take a systematic approach at determining on whether the task of collecting images should continue or it is sufficient. Also, we make observations about the ratio of the positive samples and negative samples concluding with a discussion of future work.

### OVERVIEW

A. Image Recognition/Classification

Image Classification involves taking a set of pixels representing a single image and as signing a tag to it from a given set of categories. This is one of the essential problems in Computer Vision that has many practical applications [11, 12, 13].

The first step of Image Classification process consists on the input, often referred as the training set. A set of M images, each labeled with one of K different classes. The next step is training the classifier or learning the model which means using the training set to learn what every one of the classes looks like. In the end, the quality of the classifier is evaluated by asking it to predict labels for a different set of images, never seen before by the classifier. To test the model, we can compare the true labels of these test images to the ones predicted by the classifier [10].

B. Using CNN for Image Recognition

Convolutional neural networks (CNNs) are widely used in image-recognition as they have several advantages compared to other techniques. Convolutional Neural Networks (CNN) are made up of multiple layers of feature-detecting "neurons" that have learnable weights and biases like ordinary Neural Networks [10]. Each layer has many neurons that respond to different combinations of inputs from the previous layers. The layers are constructed in the way that the first layer reveals a set of primitive patterns in data, the second layer reveals patterns of patterns and so on. For the pattern recognition, typically CNN's use 5 to 25 layers [18]. CNN's were a breakthrough discovery for advancements in image classification [2,4,8]. Deep networks usually integrate low/mid/high-level characteristics [4] and classifiers. "Levels" of characteristics can be enriched by the number of accumulated layers, which also determines the depth of the network.

## II. TRANSFER LEARNING

Transfer learning is the process of using a pre-trained model (the parameters and weights of a pre-trained network) and adjusting the model with your dataset.

When working with images, researchers typically follow the flow established by Krizhevsky, Sutskever and Hinton [2]. They train the model in a broad dataset, like ImageNet than remove the last layer, or the last few layers of the network, re-initializing them, and then re-running the whole training process. The first couple of layers can be frozen or re-trained.

When the base dataset is bigger than new dataset, transfer learning can be an optimal tool to training without overfitting a large network as target. The state-of-the-art results are obtained based on that fact, when transferring from higher layers (Donahue et al., 2013a; Zeiler and Fergus, 2013; Sermanet et al., 2014), these networks layers compute features that are very general. The importance of transfer learning is emphasized further by those results [1].

## III. A FORM OF MEASURE ON DATASET TRANSFERABILITY

In their work, Pan and Fellow [3] suggest that transfer learning could work well without retraining on the target dataset, saving so from the work to collect and label data on the domain to transfer the model. All their examples come from text datasets.

In their paper [1] – (Yosinski J, Clune J, Bengio Y, and Lipson H), give a detailed description of the transfer learning techniques and their performance. The two network architectures they have put together are identical: CNN with 8 layers, where the first three layers are the transferable layers, and the last 5 layers are retrained on the target datasets.

They benchmark the accuracy of models where the 3 first layers are kept frozen, and only the last layers are further fine-tunes, with the accuracy of the models where the error from the last layers back-propagates to the first features, and customizes them further, and report that the differences in the performance are negligible.

With regards to selecting a dataset that does well for transfer learning scenarios, the guidelines are very informal. Typically, they are expressed with suggestions along the lines of: "if the domain of images is the same, or similar enough you could receive good results with transfer learning". Although we don't aim yet at resolving the problem of a scientific process that measures and quantifies the model transferability between two problems, this set of test gives insights on how well the model is expected to perform a new dataset based on a few of its characteristics, like the number of samples and how balanced are the two classes that we'll classify. Through the following experiments, we wanted to point out not only the domain transferability, but also the transferability of the characteristics of the dataset, and more importantly to offer that as a guideline in the selecting of images for the new dataset.

## IV. DATASETS AND RUNS ARCHITECTURE

Our experiments are based on the transfer learning guides that come with CNKT 2.0. We are running on an Azure GPU N_series VM.

Hardware characteristics of the Virtual Machine

- NVIDIA Tesla K8 GPU
- 56 GB of memory
- Intel Xeon CPU E5-2690 v3 2.6GHz processor

The model that we are reusing is ResNet_18 [14] that is bundled with the CNTK tools.

A. Experiment 1- Dogs dataset

The dataset we choose for the first experiment is a collection of dog images and comes from the Stanford Vision Lab [16] for the positive samples, and outdoor images coming from the ImageNet 2011 “outdoors” synset.

The results of the experiment are summarized in the tables below.

**Table 1** Dog’s Test Dataset

| **Test** | **Dog** | 200 |
|---|---|---|
| | **Non-dog** | 348 |

**Table 2** Bigger Dataset

| **Train** | | **Results** | | | |
|---|---|---|---|---|---|
| **Positive (Dog)** | **Negative (non-dog)** | **True Positives** | **False Positives** | **True Negatives** | **False Negatives** |
| **2281** | 2376 | 4.5 | 3.4 | 12 | 21 |
| Precision = **0.899** | | Recall = 0.845 | | Accuracy = 0.908 | |

**Table 3** Medium unbalanced Dataset

| **Train** | | **Results** | | | |
|---|---|---|---|---|---|
| **Positive (Dog)** | **Negative (non-dog)** | **True Positives** | **False Positives** | **True Negatives** | **False Negatives** |
| **800** | 2376 | 151 | 49 | 346 | 2 |
| Precision = **0.755** | | Recall = 0.987 | | Accuracy = 0.901 | |

**Table 4** Small unbalanced Dataset

| **Train** | | **Results** | | | |
|---|---|---|---|---|---|
| **Positive (Dog)** | **Negative (non-dog)** | **True Positives** | **False Positives** | **True Negatives** | **False Negatives** |
| **200** | 2376 | 29 | 171 | 348 | 0 |
| Precision = **0.145** | | Recall = 1 | | Accuracy = 0.688 | |

**Table 5** Medium balanced Dataset

| **Train** | | **Results** | | | |
|---|---|---|---|---|---|
| **Positive (Dog)** | **Negative (non-dog)** | **True Positives** | **False Positives** | **True Negatives** | **False Negatives** |
| **800** | **723** | 171 | 29 | 170 | 0 |
| Precision = **0.855** | | Recall = 1 | | Accuracy = 0.921 | |

**Table 6** Small balanced Dataset

| **Train** | | **Results** | | | |
|---|---|---|---|---|---|
| **Positive (Dog)** | **Negative (non-dog)** | **True Positives** | **False Positives** | **True Negatives** | **False Negatives** |
| **200** | **200** | 162 | 38 | 146 | 14 |
| Precision = **0.81** | | Recall = 0.92 | | Accuracy = 0.823 | |

**Fig. 1** a zoom into the false negatives

B. Experiment 2 – Sheep's dataset
Duration: 20 epochs

**Table 7** Sheep's Test Dataset

| | | |
|---|---|---|
| **Test** | **Sheep** | 55 |
| | **Non-sheep** | 40 |

**Table 8** Bigger Dataset

| **Train** | | **Results** | | | |
|---|---|---|---|---|---|
| **Positive (Dog)** | **Negative (non-dog)** | **True Positives** | **False Positives** | **True Negatives** | **False Negatives** |
| **596** | **920** | 46 | 4 | 35 | 4 |
| Precision = **0.98** | | Recall = 0.94 | | Accuracy = 0.957 | |

**Table 9** Smaller Dataset

| **Train** | | **Results** | | | |
|---|---|---|---|---|---|
| **Positive (Dog)** | **Negative (non-dog)** | **True Positives** | **False Positives** | **True Negatives** | **False Negatives** |
| **322** | **647** | 54 | 1 | 36 | 3 |
| Precision = **0.938** | | Recall = 0.938 | | Accuracy = 0.91 | |

**Fig. 2** the false positive image - a sheep that gets recognized as no-sheep

**Fig. 3** false negative, non-sheep being identified as sheep

Plotting the variation of precision for the two set of experiments, the balanced datasets sample size variation in the first chart, and the unbalanced dataset sample size variation in the second chart.

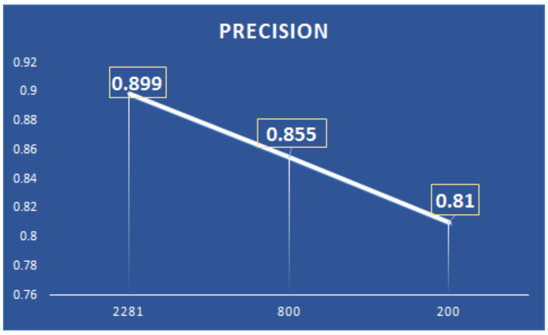


**Fig. 4** balanced dataset precision vs data points

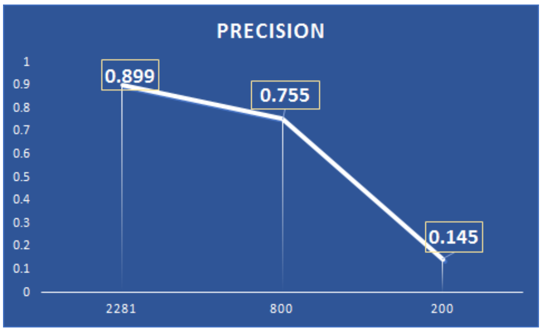


**Fig. 5** unbalanced dataset precision versus data points

## V. CONCLUSIONS AND FUTURE WORK

In our paper we gave an overview of the current landscape on image classification and recognition; and an introduction to the application of transfer learning in the field. Transfer learning is a recipe often described as a faster solution for a dataset when an already trained model exists in a similar domain.

The definition of "similarity" between the domains, therefore the datasets, is subjective and so far, the decision is taken by the data scientists based on his experience with image similarity, and visual inspection of a sample of the data from both datasets.

We suggest that there should be systematic and defined ways to quantify and score the transferability of the model from one dataset to another to avoid paying the penalty of using transfer learning and getting unsatisfying results.

We explore neural networks and transfer learning by retraining Restnet_18 previously trained on ImageNet [19] on new datasets of dogs and sheep, whose domains are subcategories of ImageNet.

Our first dataset is a collection of images of dogs gathered from the Stanford vision lab [16].

On the first set of experiments we varied the number of image samples in the positive and negative category by 11 times, going from sets of 200 images to 2200 positive samples and retraining/scoring the model gave an increase on accuracy of only 0.08.

We therefore suggest that efforts to collect more data for transfer learning scenarios, where the new dataset type is a subset of the bigger dataset, might often not be necessary.

The other metric that we explore in this paper is the organization of samples into the positive and negative categories. In machine learning problems that make no use of neural networks, where the features are well defined, and not self-learned, the principle of "having more data helps increase the accuracy of the model" is generally true.

On the second set of experiments, we varied the number of the images in the positive and negative sample groups and noticed that the ratio between the positive samples and the negative samples has a direct effect on the results.

Going from the 1:1 ratio between positive and negative samples, to a 1:43 ratio causes the precision to drop by 0.144, and a 1:10 ratio causes a 0.754 precision loss.

We conclude that the self-learned features in the negative samples, outnumber the ones in the positive samples in proportion to the dataset, and weigh in the prediction in undesired way.

We therefore suggest keeping the classes of the samples balanced, for higher precision.

Some other ideas we plan to apply in the future as part of our current work done on dataset characteristics transferability identification are techniques like: measuring the amount of neurons that get activated in every layer of the existing model, when inputting an image from the new dataset, and other quicker comparison of two images, creating composite scores to measure transferability quicker than running the full training process.

## REFERENCES

[1]. Yosinski J, Clune J, Bengio Y, and Lipson H. How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014.

[2]. A. Krizhevsky, I. Sutskever, and G. Hinton. "Imagenet classification with deep convolutional neural networks". In NIPS, 2012

[3]. SinnoJialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. IEEE Trans. on Knowl.and Data Eng. 22, 10 (October 2010), 1345-1359.

[4]. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks.In ECCV, 2014.

[5]. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks

[6]. Deep Learning of Representations for Unsupervised and Transfer Learning

[7]. Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun (2015) Deep Residual Learning for Image Recognition - Deep Residual Learning for Image Recognition

[8]. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.

[9]. Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun Deep Residual Learning for Image Recognition

[10]. Li.F, Karpathy.A, Johnson.J (2016) Convolutional Neural Networks for Visual Recognition

[11]. Jwan Al-doski, Shattri B. Mansor1 and Helmi Zulhaidi Mohd Shafri, "Image Classification in Remote Sensing" Department of Civil Engineering, Faculty of Engineering, Universiti Putra Malaysia 43400, Serdang, Selangor, Malaysia

[12]. WG SU1,2,3, FZ SU2, CH ZHOU2 1 Virtual Satellite Construction and Application for Image Classification, University of Chinese Academy of Sciences, Beijing 100049, China

[13]. Dan Cires¸an, Ueli Meier and Jurgen Schmidhuber. 2012, Multi-column Deep Neural Networks for Image Classification ¨ IDSIA-USI-SUPSI Galleria 2, 6928 Manno-Lugano, Switzerland

[14]. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun 2015 Deep Residual Learning for Image Recognition

[15]. Patterson, D. W. Introduction to artificial intelligence and expert systems. Englewood Cliffs, N.J.: Prentice Hall. (1990).

[16]. Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. 2011. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
[17]. Russakovsky, O. et al. 2014. "ImageNet Large Scale Visual Recognition Challenge." International Journal of Computer Vision.
[18]. Samer Hijazi, Rishi Kumar, and Chris Rowen, IP Group, Cadence Using Convolutional Neural Networks for Image Recognition
[19]. J Deng, W Dong, R Socher, LJ Li, K Li, L Fei-Fei 2009. Imagenet: A large-scale hierarchical image database - Computer Vision and Pattern Recognition, CVPR
[20]. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013a). Decaf: A deep convolutional activation feature for generic visual recognition. Technical report, arXiv preprint arXiv:1310.1531.
[21]. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. In International Conference on Learning Representations (ICLR 2014). CBLS.
[22]. Analysis SinnoJialin Pan1, Ivor W. Tsang2, James T. Kwok1 and Qiang Yang1, Domain Adaptation via Transfer Component, Department of Computer Science and Engineering Hong Kong University of Science and Technology, Hong Kong

Elinda Kajo Meçe, et.al. "Towards Guidelines for Assembling Image Datasets for Transfer Learning Techniques" *International Journal of Engineering Research and Applications (IJERA*), vol.10 (01), 2020, pp 08-13.