Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

On the Performance of GoogLeNet and AlexNet Applied to Sketches

Pedro Ballester and **Ricardo Matsumura Araujo**
Center for Technological Development, Federal University of Pelotas
Pelotas, RS, Brazil
pedballester@gmail.com, ricardo@inf.ufpel.edu.br

Abstract

This work provides a study on how Convolutional Neural Networks, trained to identify objects primarily in photos, perform when applied to more abstract representations of the same objects. Our main goal is to better understand the generalization abilities of these networks and their learned inner representations. We show that both GoogLeNet and AlexNet networks are largely unable to recognize abstract sketches that are easily recognizable by humans. Moreover, we show that the measured efficacy vary considerably across different classes and we discuss possible reasons for this.

Introduction

Convolutional Neural Networks (CNN) are considered the state-of-the-art model in image recognition tasks. Part of a deep learning approach to machine learning, CNN have been deployed successfully in a variety of applications, including face recognition (Lawrence et al. 1997), object classification (Szegedy et al. 2014) and generating scene descriptions (Pinheiro and Collobert 2013). This success can be partly attributed to advances in learning algorithms for deep architectures and partly to large labeled data sets made available, such as ImageNet (Russakovsky et al. 2015).

ImageNet is a large collection of hierarchical labeled images that is used in the ImageNet Challenge (Russakovsky et al. 2015). Two well known trained CNN implementations that use ImageNet are GoogLeNet (Szegedy et al. 2014) and AlexNet (Krizhevsky, Sutskever, and Hinton 2012). Both attain a low error when trained over the million of images contained in ImageNet.

GoogLeNet and AlexNet are often used in photo classification, as a large fraction of examples in ImageNet are composed of photos. In this case, they are able to generalize well and successfully classify out-of-sample examples.

Although the general approach is good enough to be deployed in commercial applications (e.g. Google Photos, Flickr), an important issue is understanding what is being learned and the limits in generalization capabilities. It is often noted that CNNs error rates are comparable to that of humans (Zeiler and Fergus 2014) when such comparison is made over a large data set such as ImageNet, but works on where humans and CNNs differ are still sparse.

In this paper we aim at analyzing the ability of GoogLeNet and AlexNet to classify sketches of well-known subjects. All subjects are included in ImageNet, although almost exclusively by photo representations. The sketches we use in this work are easily recognized by humans in most cases (Eitz, Hays, and Alexa 2012) and our initial hypothesis was that they would be able to recognize most subjects with performance comparable to that of humans.

This hypothesis stems from the feature extraction capabilities of CNNs, which is able to identify relevant features from examples and could in principle learn very abstract representations from photos that could be applied to sketches in much the same way humans do. We however show that both networks are largely unable to recognize most tested subjects, indicating that the learned representations are quite different from that of humans. We argue that such approach can be useful to assess classifiers' generalization capabilities, in particular regarding to the abstraction level of learned representations.

The main contribution of this work is to put forward an image recognition task where current state-of-the-art models differ significantly in performance when compared to humans. By training over less abstract examples (photos) and testing in more abstract examples (sketches) one may be able to better understand what is being learned by these models.

This paper is organized as follows. We begin by presenting related prior works, following with the statement of our goal and our proposed methodology, including descriptions of the models and data sets. We then show our analysis of the results. Finally, we conclude the paper and discuss some possible interpretations and future work.

Related Works

Convolutional Neural Networks have been applied to a variety of tasks with state-of-the-art performance in several applications. The most used architecture follows that of (LeCun et al. 1998), where a CNN was first applied to recognize hand-written digits.

The network has been improving since its creation with the development of new layers (Srivastava et al. 2014; Girshick et al. 2014) and the addition of classic computer vision techniques. CNN are often used in the ImageNet Challenge with many different variations

In (Eitz, Hays, and Alexa 2012) a data set of sketches

is provided, which we use here and detail in the next sections. In that paper, it was shown that humans could achieve a 73.1% accuracy on the data set and results on a classifier were provided, showing a 56% accuracy.

CNN applied to the same data set was explored in (Yang and Hospedales 2015), where a CNN was trained to recognize sketches yielding an accuracy of 74.9%, hence outperforming humans in the same task. The proposed approach however makes use of the strokes' order to achieve such high accuracy.

Recently, some studies aim at understanding Deep Neural Networks behavior in adverse circumstances. (Szegedy et al. 2013) presents a study on how small changes in images can drastically change the resulting classification. In the opposite direction of our work, (Nguyen, Yosinski, and Clune 2014) presents images that are completely unrecognizable by humans but are classified with high certainty by the networks.

Goals and Methodology

Our main goal is to understand whether a concept learned by CNN mostly in photos are transferable to more abstract representations of the same object. Our general methodology consists of using CNN trained over the ImageNet data set and applying the resulting model to the sketch data set, registering the classification for each example in each category and analyzing the results.

It must be observed that we do not include the sketches in the training set, which is known to provide good results (Yang and Hospedales 2015). Hence, we are not interested in building better classifiers but rather understand often used models and how they work when applied across different representations.

Models

We use two well-known trained CNNs, GoogLeNet (Szegedy et al. 2014) and AlexNet (Krizhevsky, Sutskever, and Hinton 2012). Both networks have participated in ImageNet Contest in different years with good results.

The networks differ in general architecture. GoogLeNet has Inception Modules, which perform different sizes of convolutions and concatenate the filters for the next layer. AlexNet, on the other hand, has layers input provided by one previous layer instead of a filter concatenation. Both networks have been tested independently and use the implementation provided by Caffe (Jia et al. 2014), a Deep Learning framework.

Test Data Set

We use the TU-Berlin sketch (Eitz, Hays, and Alexa 2012) data set, a crowd-sourced collection of sketches containing 20,000 sketches of everyday subjects divided into 250 categories. For this study, We chose some wide categories that are contained in WordNet, which is used to label ImageNet's examples. The chosen categories are: airplane, bird, car, cat, dog, frog, horse, ship and truck.

Furthermore, we grouped ImageNet's less specific labels into broader sets that matched that of the sketches. Hence, for example, "seagull" was relabeled as "bird" and so were the many different dog breeds into a single "dog" class. Table 1 shows how many ImageNet classes were included in the same TU-Berlin class. This results in an easier task for the classifiers and accounts for the high abstraction levels of sketches. In the end, each category contains 80 sketches.

Table 1: Classes in ImageNet and TU-Berlin Relationship

Sketch class	# of ImageNet classes
Airplane	3
Bird	60
Car	5
Cat	5
Dog	118
Frog	3
Horse	1
Ship	1
Truck	3

Evaluation

Both CNN output a probability distribution over possible classes for the input. Two different methods were used to evaluate the results. The first one considers only the top 10 most probable classes and the second one register the position of the correct class in the full probability ranking.

In the first method, we rank the networks' outputs by their probability and consider only the top ten most probable classes. We then count how often each class appears for each image of each target category.

This method allows to assess whether the correct output is being given a high, and useful, probability, but also allows for observing qualitatively how consistent the results are for each category. In this latter case, one expects that for each category the top 10 does not vary significantly.

In the second method, we construct descriptive statistics over the position of the correct class in the probability ranking. The higher the position in the ranking, the better the classification - ideally, the correct class would be located in the first position.

We calculate the mean and standard deviation for each category. A low mean corresponds to a higher average position in the ranking, while a low standard deviation is evidence of consistency of the output for different instances of the same category. This also allows for capturing the best and worst instances of each category which we use to discuss possible reasons for the observed results.

Results

Analysis of Top 10 Ranking

We begin by analyzing the top 10 rankings. Tables 2 through 10 depict the frequency of occurrences of each categorization over the available images for each category. For instance, in Table 2, for the 80 available images labeled as "airplane", AlexNet had "safety pin" in the top 10 most probable classifications for 79 of them.

When analyzing each table independently, one can observe that for no category does the correct label appear

Table 2: Outputs for Airplane class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Safety pin	79	Hook, claw	80
Hook, claw	75	Safety pin	65
Corkscrew, bottle screw	68	Bow	60
Walking stick, stick insect	54	Nematode	55
Envelope	51	Hair slide	50
Nail	51	Stethoscope	44
Hair slide	46	Nail	32
Necklace	32	Walking stick, stick insect	30
Chain	28	Corkscrew, bottle screw	30
Nematode	27	Binder, ring-binder	29

Table 3: Outputs for Bird class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Safety pin	73	Hook, claw	80
Hook, claw	69	Stethoscope	61
Corkscrew, bottle screw	53	Bow	60
Necklace	50	Nematode	55
Harvestman, daddy longlegs	45	Safety pin	43
Chain	42	Necklace	39
Walking stick, stick insect	37	Binder, ring-binder	36
Envelope	37	Chain	34
Nematode	34	Corkscrew, Bottle screw	27
American egret, Egretta albus	21	Nail	26

Table 4: Outputs for Car class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Safety pin	78	Hook, claw	80
Hook, claw	69	Hair slide	68
Hair slide	68	Safety pin	63
Chain	66	Binder, ring-binder	55
Envelope	62	Stethoscope	52
Corkscrew, bottle screw	47	Plate rack	45
Necklace	47	Envelope	35
Binder, ring-binder	45	Nematode	34
Stretcher	39	Necklace	33
Whistle	35	Chain	30

Table 5: Outputs for Cat class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Hook, claw	69	Hook, claw	80
Necklace	63	Stethoscope	72
Chain	53	Nematode	55
Envelope	49	Binder, ring-binder	47
Safety pin	40	Envelope	46
Harvestman, daddy longlegs	40	Bow	45
Garden spider	33	Chain	43
Nematode	27	Necklace	42
Corkscrew, bottle screw	26	Safety pin	37
Paper towel	24	Cassette	32

Table 6: Outputs for Dog class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Hook, claw	75	Hook, claw	79
Safety pin	68	Stethoscope	73
Chain	65	Nematode	72
Envelope	61	Bow	61
Necklace	61	Chain	58
Corkscrew, bottle screw	53	Binder, ring-binder	51
Nematode	44	Safety pin	43
Whistle	31	Hair slide	43
Hair slide	27	Envelope	32
Binder, ring-binder	26	Necklace	31

Table 7: Outputs for Frog class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Safety pin	69	Hook, claw	79
Necklace	66	Stethoscope	65
Chain	63	Nematode	61
Envelope	59	Envelope	57
Hook, claw	58	Binder, ring-binder	54
Binder, ring-binder	43	Necklace	43
Corkscrew, bottle screw	36	Safety Pin	42
Hair slide	33	Bow	41
Fire screen, fireguard	33	Chain	40
nematode	30	Hair slide	39

among the most frequent classifications. Nonetheless, both networks are quite consistent, having high counts for a small subset of classes. In other words, the networks provide consistently incorrect classifications.

By observing all tables a pattern emerges, where essentially the same classifications are being given for all images in all different categories. AlexNet essentially sees a safety pin in all cases, while GoogLeNet sees a hook/claw. For other less frequent classifications there are still a large overlap across different categories.

The reason for this behavior seems to be that most classifications are for objects that contain simple, thin traces in their composition, such as safety pins and bow strings. It is therefore understandable that the networks may mistake a drawing with a few lines with these objects. But it is also clear that this is a major departure from what humans offer as classifications for the same sketches.

Analysis of Position of Correct Label

We now turn our attention to the position in the ranking of the correct label. A correct classification is one where the correct label is the most probable classification output by the network, i.e. the first position in the rank composed of 1000 labels. Tables 11 and 12 summarize the results.

Both networks behave in a very similar way (the Pearson correlation coefficient between means is of 0.92) and none perform adequately in a consistent manner. On average, networks are behaving better than random guessing, but results are hardly useful in practical terms. Even when considering the best ranked image of each class the networks perform poorly, with the exception of a few Bird sketches that were correctly classified.

The best results occur for Bird and Dog classes, which are two of the largest classes by number of examples contained in Imagenet, with many different sub-classes (see Table 1). This suggests that having a larger set of examples does help in identifying more abstract representations. It is not clear, however, what helps the networks correctly predict individual instances.

Figures 1 and 2 show the best and worst ranked sketches for the Bird class. Qualitatively it seems that the best ranked images are slightly more realistic in their proportions, including more details of feathers and some texture. In any case the sketches in both groups are not that different and even the worst ranked ones are arguably easily recognizable by humans. Similar observations can be made for the other categories.

These results are evidence that the networks can recognize some sketches successfully, albeit they seem to require less abstract representations for that and seem to be rather sensitive to small changes in such abstraction level. Again,

Table 8: Outputs for Horse class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Safety pin	74	Hook, claw	79
Chain	69	Bow	74
Corkscrew, bottle screw	64	Stethoscope	66
Hook, claw	59	Chain	63
Envelope	57	Safety Pin	60
Necklace	50	Binder, ring-binder	59
Hair slide	46	Nematode	55
Walking stick, stick insect	40	Hair slide	52
nematode	31	Corkscrew, bottle screw	38
Stretcher	30	Nail	36

Table 9: Outputs for Ship class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Safety pin	71	Hook, claw	74
Envelope	50	Bow	45
Hook, claw	47	Safety pin	37
Nail	46	Plate rack	29
Walking stick, stick insect	46	Nail	27
Plate rack	44	Walking stick, stick insect	27
Folding chair	33	Stethoscope	27
Harvestman, daddy longlegs	31	Binder, ring-binder	26
Corkscrew, bottle screw	30	Matchstick	23
Bow	29	Harvestman, daddy longlegs	23

Table 10: Outputs for Truck class

AlexNet's Output	Frequency	GoogLeNet's Output	Frequency
Safety pin	79	Hook, claw	80
Envelope	69	Safety pin	74
Hook, claw	66	Binder, ring-binder	66
Corkscrew, bottle screw	50	Plate rack	51
Binder, ring-binder	50	Stretcher	46
Stretcher	50	Hair slide	42
Chain	49	Stethoscope	38
Buckle	44	Bannister, handrail	33
Fire screen, fireguard	29	Bow	28
Lighter, light, igniter	23	Swing	26

this is a departure from how humans classify sketches

Conclusions

In this paper we presented results of applying Convolutional Neural Networks trained using images from Imagenet on a sketch data set. Our goal was to observe whether training these networks on mostly realistic representations of subjects (photos) is enough to classify more abstract representations (sketches). The work presented here contributes to the understanding of the applicability of CNN in domains that are different but related to that of the training set.

We showed that both AlexNet and GoogLeNet are largely unable to consistently classify the shown sketches. The drawing lines are often confused for objects with simple and thin lines, such as safety pins. Nonetheless, both networks are able to perform better than random guess and display better performance for some categories. When analyzing such categories we found that categories with more training images are better classified and we observed that instances with more correct classifications are qualitatively less abstract.

Overall, we conclude that the test networks are not useful to perform sketch classification and their classifications are different from the classifications offered by humans. This is in contrast to results using photos where CNN are able to approach human accuracy. These results provide initial evidence that these networks are overfitting on the abstraction level of the training images, making them unable to transfer the learned concepts to those simpler, more abstract representations.

Table 11: Position of the correct label (AlexNet)

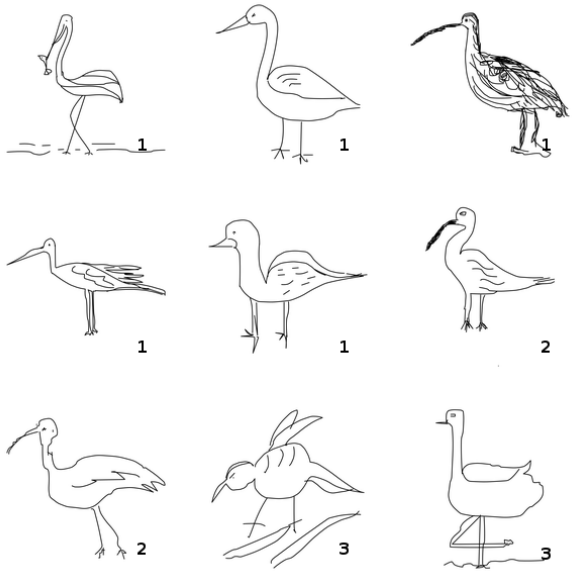
Expected Class	Best Ranked	Worst Ranked	Mean	Std. Deviation
Airplane	20	749	195.81	135.03
Bird	1	127	26.98	26.59
Car	13	376	133.01	83.35
Cat	148	635	387.34	98.52
Dog	9	324	82.75	61.67
Frog	387	942	711.1	123.44
Horse	453	931	723.55	126.56
Ship	38	693	342.55	165.92
Truck	177	689	422.4	110.42

Table 12: Position of the correct label (GoogLeNet)

Expected Class	Best Ranked	Worst Ranked	Mean	Std. Deviation
Airplane	27	739	377.16	203.05
Bird	1	178	41.6	44.44
Car	11	453	183.14	88.00
Cat	129	456	276.43	68.12
Dog	78	269	187.83	42.99
Frog	265	897	602.76	132.99
Horse	508	941	780.41	83.77
Ship	24	851	403.04	213.15
Truck	34	755	322.30	160.33

Of course, CNN are perfectly able to classify sketches if they are trained over adequate examples containing sketches (Yang and Hospedales 2015), but that they are unable to transfer the learned concepts to sketches when trained using photos is an interesting outcome since it arguably differs from the ability of humans to do so. Our results show that one possible solution is to train with even more data,

Figure 1: Best ranked birds from AlexNet. Images presented to the network and their respective best ranked correct guess.



possibly with more diversified examples. More interesting is the need to improve learning algorithms and network architectures to make them better cope with this kind of task. We argue that using different representations for training and testing, in particular by using sketches, is an useful method to evaluate concept learning.

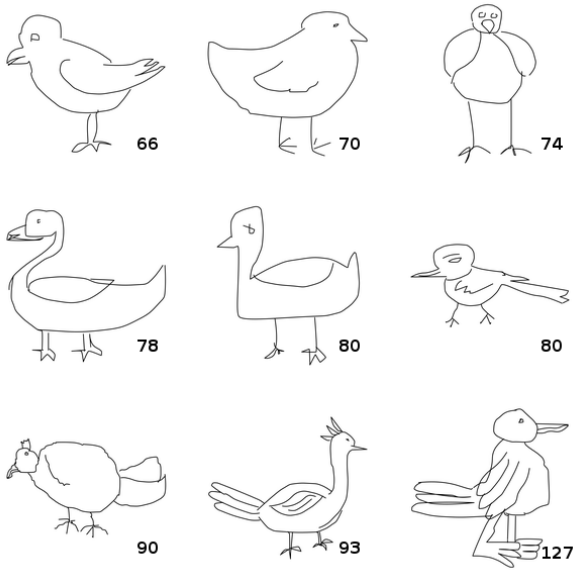
In this direction, an important question is whether training over sketches representing objects that are not present in the test set could help the network learn more adequately. For instance, it could be argued that humans would not be able to identify sketches if they were not exposed to sketches before - i.e. sketch identification is a skill in itself and needs to be learned. It must be noted that both data sets used contain a number of sketch-like images, but very few are related to particular objects and are used to represent concepts such as painting or art.

We plan on improving the analysis by testing the networks on sketches and images with different levels of details, using techniques such as kurtosis measurement of wavelet coefficients (Ferzli, Karam, and Caviedes 2005) to provide quantitative measurements of detail, relating that to classification accuracy. Also as future work we plan on training CNN over more constrained categories and varying the diversity of the examples, including untargeted sketches, relating that to classification performance.

Acknowledgment

This work is partly supported by CNPq (Brazilian National Research Council). We thank the anonymous reviewers for their contribution.

Figure 2: Worst ranked birds from AlexNet. Images presented to the network and their respective best ranked correct guess.



References

Eitz, M.; Hays, J.; and Alexa, M. 2012. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)* 31(4):44:1–44:10.

Ferzli, R.; Karam, L. J.; and Caviedes, J. 2005. A robust image sharpness metric based on kurtosis measurement of wavelet coefficients. In *Proc. of Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics*.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

Lawrence, S.; Giles, C. L.; Tsoi, A. C.; and Back, A. D. 1997. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks* 8(1):98–113.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Nguyen, A.; Yosinski, J.; and Clune, J. 2014. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv preprint arXiv:1412.1897*.

Pinheiro, P. H. O., and Collobert, R. 2013. Recurrent convolutional neural networks for scene parsing. *CoRR* abs/1306.2795.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 1–42.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2014. Going deeper with convolutions. *CoRR* abs/1409.4842.

Yang, Y., and Hospedales, T. M. 2015. Deep neural networks for sketch recognition. *CoRR* abs/1501.07873.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*. Springer. 818–833.