Exploring the Effects of Context in Image Classification

Alexander Lamson
University of Massachusetts Amherst
Amherst, MA
alamson@umass.edu

Christopher Raff
University of Massachusetts Amherst
Amherst, MA
craff@umass.edu

Abstract

Great success has been seen using CNNs (convolutional neural networks) to classify object categories in images. When we train and test CNNs, we apply the convolutional kernels to every pixel in the image. This means that when we train a CNN on a dataset of pictures of horses, for example, we are measuring the error with respect to all the pixels in the image, not just the pixels that belong to the horse.

What are the implications of this? Certainly it's useful to learn the context that an object often appears in. However, it can lead to misclassifying objects purely based on the situation. For example, a dog sitting in a car could be misclassified as a person, because people are frequently seen sitting in cars and dogs are not. We propose that a model trained with intentionally obscured or altered backgrounds could avoid this bias, and this is what we will explore.

Figure 1. A floret of broccoli on a car dashboard

1. Introduction

The problem is that in most cases context can help a model classify what a target object is in an image, but if the context is very unusual for that object then the model can get misclassify the image or decrease its probability assigned to the correct class. We seek to solve this problem by taking images where pixels belonging to the target class have been labeled and changing the non-target pixels in different ways. We trained models for each of the different data augmentation techniques to measure which model has the best performance when classifying images where the context is unusual. In order to validate on a dataset of images where the context is unusual, we needed to first create such a dataset. This is done by training a model on images where the target object is hidden and only the context is shown, then observing validation images where the model is confident, but predicts the incorrect class. We define "unusual context" in this way because it implies that the context is less probable given the distribution of contexts for that class.

There are two potential benefits to this approach. First, training a model using this data augmentation technique guarantees that the context and the object are completely separated in the model. In other words, the model should only be learning convolutional filters which activate highly for the object itself and not for objects which are commonly correlated with appearing in the same scene as that object, like an arm with a dumbbell.[3]

The second benefit of using this approach relates to the idea of dataset collection. When an academic group collects an image dataset, they will collect images by all means available to them, especially those methods which are convenient. Although many methods can be used to collect image data, there are no guarantees that the distribution of images collected in the dataset is truly representative of real life.

If images could be generated according to our approach, the dataset can contain images containing novel contexts which would be difficult to acquire using traditional techniques. This could potentially add robustness to the model for specific situations when the model is run during test time on unusual things happening in real life.

2. Background/Related Work

In 2015, Google published a blog post[3] that explored the problem of how to visualize what a neural network-based object classifier had learned in each layer. They performed gradient ascent on images of random noise to maximize the probability of a chosen target class. This would theoretically produce images where the model would most strongly assign probability to that target class. When they performed this experiment while targeting the "dumbbell" class, they found that it generated images containing a dumbbell but also an arm holding the dumbbell (see Figure 2). That shows that without specifically labeling what is the object and what is context it is possible for those two ideas to become mixed up within the model. By using a model that explicitly segments out the pixels belonging to the target class we can attempt to eliminate this problem.

Figure 2. Google's images created by performing gradient ascent of the image pixels with respect to the "dumbbell" class. Note the arms appearing in the visualizations.

In the paper "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" (CycleGAN[4]), a model is created which can translate unpaired images from one category to another. In one example, the model is used to translate a picture of a horse to a picture of a zebra. Although the horse successfully gains stripes like a zebra, the grass also changed from green to yellow because zebras are often seen in scenes with yellow grass, as opposed to horses which are commonly seen in fields of green grass. It could be claimed that this is a good effect that shows that the image transfer happened quite successfully, but only if altering the background is the intent of the technique. One can imagine attempting to train a model to do the same thing, but only minimally affecting the surroundings of the zebra.

In the paper "Context Augmentation for Convolutional Neural Networks"[1], techniques for generating images with artificial contexts were explored. This was used to improve classifier performance in situations where the number of images in the dataset was small and such data augmentation techniques were necessary to achieve good performance. This paper found that generating a background for an object by sampling from its class rather than from all classes resulted in a better performance of their classifier. This suggests that context is useful when validating on normal images, but it does not give insight into how the classifier would perform when tested on unusual images. Inspired by this paper, our novelty is that our data augmentation seeks to create classifiers that perform better in unusual contexts, rather than purely measuring performance against images with a typical distribution of contexts. We also automatically create a dataset of images with unusual contexts, and validate our models against this set of images.

3. Approach

We plan to achieve improved context-free classification in the model by making a modification to the training data. Each image will have a mask marking out the target object. We will alter the pixels outside of that mask in different ways to explore the effect on the performance of the classifier in typical and unusual contexts. Every model trained will be initialized a pretrained ResNet18[2] architecture, with the last fully connected layer replaced such that there are 9 output classes. Only the weights of the last layer are altered during training for each model.

The different models we trained had the background pixels replaced with black, with random noise, or with crops from hand-picked "empty scenes".

At test time we won't be able to remove the background from the images, which means the network will still be producing results with respect to the background. However, unless the background itself looks like a target class, it shouldn't increase the probability of a particular class and so we believe this will produce reasonable results.

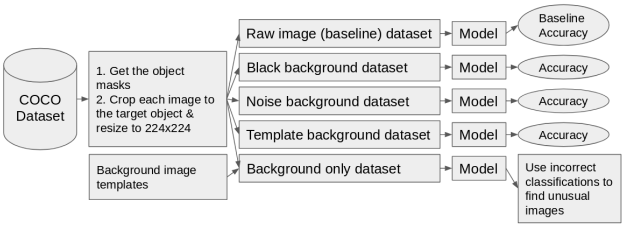


Figure 3. Image processing pipeline

3.1. Dataset

In order to generate our datasets, we require images where the pixels of each image are labeled as belonging or not belonging to the target class. This is so we can differentiate between the object and the context (pixels not belonging to the target class). A dataset of segmented, labeled images would achieve this goal, so we used the MS-COCO dataset. The dataset is well established and contains many labeled segmented images belonging to a variety of classes. We used this segmentation dataset to derive our own dataset of single-class images where we have information about which pixels belong to the object and which pixels belong to the background.

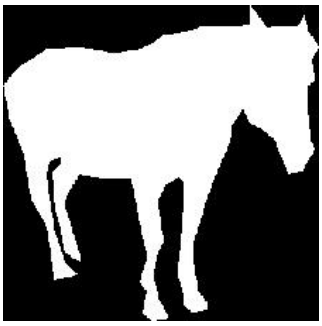
The classes we chose to classify were cats, dogs, horses, broccoli, carrot, cars, bicycles, boats and airplanes. We

chose these classes to have a variety of contexts, with several having overlapping contexts such as cars and bicycles.

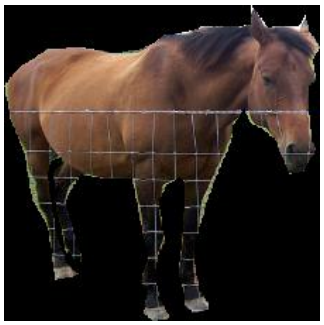
Figure 4. An example image from the dataset

Figure 5. Visualization of the image annotations

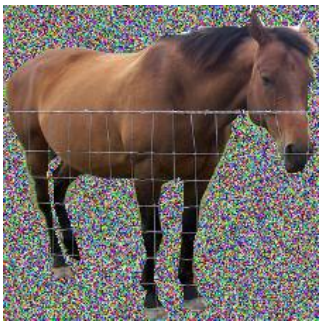
(a) A crop of one of the horses from Figure 4



(b) The horse's object mask



(c) The horse with the background set to black



(d) The horse with the background replaced with random colors

(e) The horse with the background replaced with an empty water scene

4. Experiments

Our initial attempt at creating a context agnostic classifier was to turn every pixel black which did not belong to the target class. This had very high validation accuracy, but performed poorly when tested on actual images. We believe this occurred because the model never saw a background in the training set, so when the model processed backgrounds in the test set, it resulted in convolutional kernels activating for elements of background of the scene, which caused misclassifications and lowered performance.

The next background alteration technique that we tried was turning all the background pixels into random colors. This performed slightly better than the black background classifier on real images, but still not well compared to baseline (see Table 1). Changing the pixels to black means that the background is extremely structured which resulted in poor performance at test time. Inversely, changing the pixels to noise causes extremely unstructured backgrounds which also result in poorly performance at test time. This is likely because it was also not representative of the backgrounds found in real images.

To resolve the problem of fitting to backgrounds that were structured differently than in reality, a new background alteration technique was used. The background was replaced with a random crop from a set of images which we had manually validated to be empty scenes. The intuition behind doing this was that the black and random noise pixels weren't sampled from reality, which meant that the classifier was still seeing backgrounds that were sampled from

a very different distribution at train time when compared to test time. Using image templates in this way is also closer to reality in that it is physically possible to see a cat in a desert, but it is almost certainly not possible to find a cat amongst random colored noise.

Figure 6. Example images created by pasting objects into crops of empty scene template images.

Figure 7. Picture of a dog with the corresponding saliency map. When using the template model, the activations appear over the dog, showing that the model thinks that the dog pixels are most important in deciding that this is a picture of a dog.

The first time we trained all these models and tested them, our target classes included couches and beds. These two classes were replaced due to two issues. First, couches and beds often occupy the entire frame of the image in the original dataset, which effectively resulted in that object becoming the "context" for the image. The second issue was that pictures of couches and beds from the dataset often had dogs or cats on them. This meant that many images contained pixels with overlapping labels of both cat and couch. In other words, pixels belonging to a cat were labeled "bed". To fix these issues, we simply replaced these classes with boats and bicycles. Boats were typically found in bodies of water, which was dissimilar from the other classes. Bicycles were typically found on streets and urban areas, similar to cars, although there was very little class overlap.

4.1. Creating a dataset of unusual-context images

It was necessary to create a set of unusual context images to tell how the models performed compared to the baseline classifier in the unusual contexts we care about. The first attempt at doing this was to train a classifier on the context-only images and extract the images where the model performed poorly. It was trained on the context-only images by setting pixels to black if they belonged to the target class. This created a classifier that effectively became a silhouette classifier. In other words, it used the shape of the outline of the object to classify the image, rather than the colors of the pixels in the remaining context. This is visible by looking at the saliency map in Figure 8.

Figure 8. Picture of a horse with the corresponding saliency map. Note how the saliency activations are grouped on the edges of the horse.

This silhouette problem was resolved by blacking out all the pixels in a bounding box around the target object, as opposed only blacking out the the specific pixels belonging to the object. Now that object was hidden by a rectangle, the model couldn't use the shape of object to determine its class. To allow the classifier to continue to use the scene context, the bounds of the resulting image was widened to allow a gap of scene context one third the size of the bounding box around the edge. An example can be seen in Figure 9.

4.2. Results

Model	Raw Image Acc.	Val. Acc.	"Unusual Acc."
Baseline	87%	87%	88%
Hand-picked Crops only	81%	80%	87%
Hand-picked Crops w/ raw	87%	82%	86%
Black	71%	82%	79%
Noise	74%	83%	77%

Table 1. Accuracies for different models

5. Conclusion

We were able to perform comparably to baseline's performance on our unusual set. As shown in Table 1 our best

Figure 9. Image of the updated technique for generating context-only images. The target class is bicycle.

Class	Both	Ours	Baseline	Neither
Airplane	72%	1%	20%	7%
Bicycle	85%	4%	4%	7%
Boat	70%	12%	7%	1%
Broccoli	88%	1%	7%	4%
Car	81%	5%	7%	7%
Carrot	70%	3%	15%	12%
Cat	74%	3%	10%	11%
Dog	74%	8%	9%	9%
Horse	81%	4%	5%	10%
Grand Total	79%	5%	8%	8%

Table 2. Per-class accuracy over the raw-image validation set. The column name refers to which model(s) correctly identified the class.

Class	Both	Ours	Baseline	Neither
Airplane	100%	0%	0%	0%
Bicycle	100%	0%	0%	0%
Boat	75%	15%	5%	5%
Broccoli	89%	2%	5%	4%
Car	88%	3%	5%	3%
Carrot	67%	3%	12%	18%
Cat	91%	0%	0%	9%
Dog	67%	20%	13%	0%
Horse	73%	9%	0%	18%
Grand Total	82%	6%	6%	6%

Table 3. Per-class accuracy over the unusual validation set. The column name refers to which model(s) correctly identified the class.

(a) The baseline classifier struggled with monochromatic images, while our classifier usually performed well on them.

(b) Pictures of figures, toys, and other representations of objects tended to be found by the model to have an unusual context.

(c) Occluded objects tended to be found by the model to have an unusual context.

Figure 10. Examples of automatically discovered unusual-context images

performing model is the one trained on images with backgrounds from hand-picked crops. We can make the model less sensitive to context by excluding the raw images in the training set. When we do this, we observe that the performance in typical contexts gets worse, but the performance in unusual contexts gets better. This shows how context is usually helpful in determining what category an object is. As our top two models suggest, there is a balance between general performance and performance in the unusual case as we expected (see Table 1).

Based on the accuracies in Table 2, the boat class has the most unusual contexts out of the raw image validation set. Accuracies on the unusual set, shown in Table 3 suggest that dogs and horses also have contexts that the baseline classifier struggles with.

Some images that were automatically put in the unusual set were not the unusual contexts we expected. As shown in Figure 10, they included toys and occluded objects. Another unexpected result is that our classifier seemed to perform well on monochromatic images, while the baseline classifier struggled with them.

The results also show that artificially created back-

grounds that are extremely structured (black) or extremely non-structured (noise) result in poor performance when a classifier is trained on those backgrounds. Interestingly, noisy backgrounds make a better general classifier than black backgrounds, but black backgrounds make a better classifier for our unusual set.

The validation accuracy for the hand-picked crops model is lower than its accuracy on real images. This indicates that our approach of training on images you don't expect to test on can lead to similar or better results compared to the baseline performance. However, introducing raw images to the hand-picked crops dataset hurts the performance of the classifier in unusual cases. Perhaps the performance of the classifier on the unusual set could be improved by using a larger set of "empty scene" image templates.

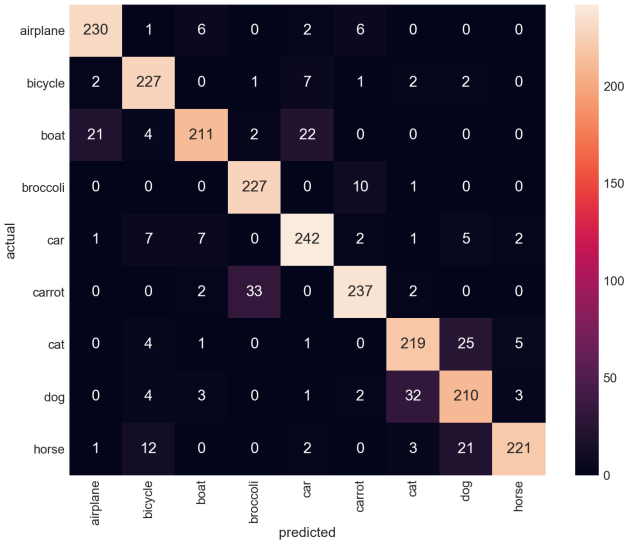


Figure 11. Baseline confusion matrix over the raw validation images

5.1. Code

All code is available at our github repository [1].

5.2. Future work

Further exploration could measure how the empty scenes' size, quantity and types of contexts can effect the performance of the classifier.

The model can perform comparably to baseline. We found that changing the ratio of generated images to real images in the training set can affect how the model performs on typical images or unusual-context images. This ratio hyperparameter could be tuned to find a desired balance between performance on typical images and unusual-context images.

[1] https://github.com/chrisraff/context-free-network

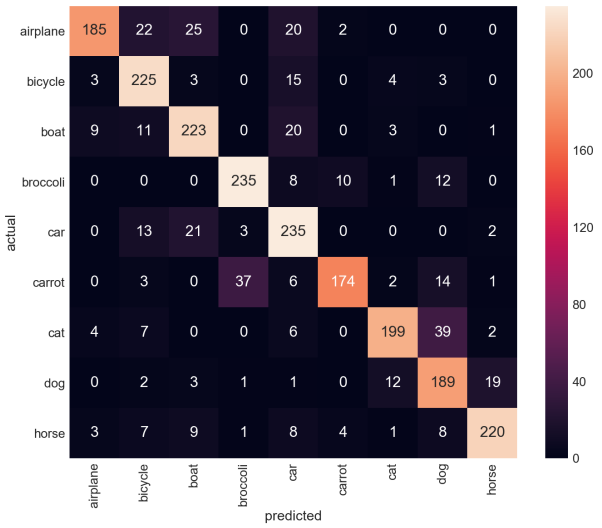


Figure 12. Our model's confusion matrix over the raw validation images

To understand how well our model has learned the difference between context and the labeled object, it could be useful to have the model generate images of the target classes and compare them to what the baseline model generates. This could be done with either a GAN (Generative Adversarial Network) or by gradient ascent on noise. In other words, if a model were to have the problems of arms appearing in such images generated for a dumbbell class, like in Figure 2, then it could be observed if the arms disappeared from the generated images by using our technique.

References

[1] A. Dundar and I. Garcia-Dorado. Context augmentation for convolutional neural networks. *arXiv preprint arXiv:1712.01653*, 2017.

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

[3] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. `https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`. Accessed: 2018-11-17.

[4] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.