Data augmentation with Neural Artistic Style

Anonymous Submission

Anonymous Affiliation

Abstract

This paper presents a novel approach to use the neural image style transfer as a data augmentation strategy for other image-based deep learning algorithms. The success of training deep learning algorithms heavily depends on a large amount of annotated data. Recent neural style transfer approch can apply the style of an image to another image without changing its high level semantic content, so we think it is reasonable to use this method as an data augmentation strategy in computer vision tasks. We explore stat-of-art neural style tranfer algorithms and build a novel approch to apply it as a data augmentation strategy on the image classification algotithms. We finally compare to and combine with the the traditional approaches to show the effectiveness of this method.

Keywords: Neural Style Transfer, Data Augmentation, Image Calssification.

1 Introduction

Deep Convolutional Neural Networks (CNN) have grown in popularity for performing image processing tasks, such as image classification, object detection and segementation. In the state-of-art network architechters, many different kinds of data augmentation strategies, like zooming, fliping, and cropping, have been used and found effective to improve the performance [Krizhevsky et al., 2012] [Simonyan and Zisserman, 2014] .

At the mean while, The Neural Algorithm of Artistic Style has been found can apply artistic style to a image without chaning the high level content of the original images [Gatys et al., 2016]. The basic idea of image style transfer is to jointly minimise the distance of the feature representations of a white noise image from the image content representation in one layer and the painting style representation defined on a number of layers of the Convolutional Neural Network.

For many applications, gathering raw data can be very time-consuming or difficult, especially for image and video classification tasks. Some critical image-based tasks like cancer detection [Kyprianidis et al., 2013] are hindered by this lack of data. Similarly, for many start-up company in the AI industry often have difficult to access enough data. Besides, for Machine learning algorithms, if we only use the raw data, the model is easy to get overffiting and does not have enought generalization. So we explore the effectiveness of distinct data augmentation techniques which can help us to address the menthioned problems.

The datasets used in this project are the caltech101 dataset[Fei-Fei et al., 2006] and caltech256 [Griffin et al., 2007]. in caltech101, pictures of objects belonging to 101 categories. About 40 to 800 images per category. Most categories have about 50 images. The size of each image is roughly 300 x 200 pixels. Caltech256 is a collection of 30607 images in 256 categories which is much bigger than caltech101. To evaluate the effectiveness of augmentation techniques, we split both datasets as 70% of images are used for training and 30% of images are used for validation.

We will apply the pre-trained vgg16 model and vgg19 model to perform a rudimentary classification. Troditional data augmentation techniques will be firstly used, and retrain our models. Next, we will make use of neural style transfer to augment the data by transferring styles from images in the dataset to a another image. Finally, we explore and propose a different kind of augmentation where we combine style transger and troditional methods together. We will finally evaluate the performance on validation dataset and use classification accuracy as the metric to compare these augmentation strategies.

2 Related Work

We will present the a overview of past work.(In progress, need to do more literature review)

2.1 Troditional Data Augmentation Tecniques

2.1.1 AlexNet

AlexNet from [Krizhevsky et al., 2012] is the winner of ILSVRC 2012 and the first model to make CNN popular in Computer Vision field. In this work, a 8 layers CNN model are introduced. Data augmentation techniques such as image translations, horizontal reflections, and patch extractions were used to avoid overfitting. ReLU and dropout are also used in this paper to avoid overfitting.

2.1.2 VGGNet

VGGNet is a simple but deep model created by [Simonyan and Zisserman, 2014]. This model strictly used 3x3 filters with stride and pad of 1, along with 2x2 maxpooling layers with stride 2.

- 3 3*3 conv layers back to back have an effective receptive field of 7x7.
- Used scale jittering as one data augmentation technique during training.
- But it took a very long time to train: Trained on 4 Nvidia Titan Black GPUs for two to three weeks.

2.1.3 ResNet

ResNet by [?] is a 152 layer network architecture that won ILSVRC 2015.

- The idea behind a residual block is that you have your input x, after conv layer, relu layer and normalization layer series, you will get fearure maps: F(x). That result is then added to the original input x: H(x) = F(x) + x, and then continue the training.
- **Naive increase of layers in plain nets result in higher training and test error.**
- The group tried a 1202-layer network, but got a lower test accuracy, presumably due to overfitting.

2.1.4 ZFNet

- Reconstruction in ZFNet: [?] introduced another reconstruction method with Unpooling, Rectification and Filtering are applied for visualization of an activation in some layers, But it can only reconstrut one activation one time (To examine a given convnet activation, all other activations in the layer are set to zero and pass the feature maps as input to the attached deconvnet layer).

2.1.5 Neural Style Transfer

The algorithm of [Gatys et al., 2016] is the first method that use Gram matrices to to represent the style and use some layer to represent the content, and then reconstruct the stylized image by minimizing the loss by gradient descent with backpropagation. The basic idea of image style transfer is to jointly minimise the distance of the feature representations of a white noise image from the image content representation in one layer and the painting style representation defined on a number of layers of the Convolutional Neural Network . The author found that replacing the maximum pooling operation by average pooling yields slightly more appealing results. Adjust the trade-off between content and style to create different images. The different initialisations do not seem to have a strong effect on the outcome of the synthesis procedure In this work, the author consider style transfer to be successful if the generated image âĂŸlooks likeâĂŹ the style image but shows the objects and scenery of the content image. VGG network is applied in this work.

Although the the work by [Gatys et al., 2016] can produce impressive stylized images, there are still soem efïňĄciency issues "since each step of the optimization problem requires a forward and backward pass through the pretrained network" [**?**]. We are going to apply the Style Transfer as a data agumentation strategy, so the performance is of great importance. Based on the previous work, many Fast Neural Style Transfer methods has been proposed.

2.2 Perceptual Losses and FeedForward Network

Based on the algorithm proposed by [Gatys et al., 2016], [**?**] introduced a much faster approach.

- Their system consists of two components: an image transformation network and a loss network
  - The image transformation network is a deep residual convolutional neural network parameterized by weights W; it can transform input images to output images.
  - Each loss function computes a scalar value to measure the diïňĂerence between the output image and a target image, including Feature Reconstruction Loss and Style Reconstruction Loss.

2.3 N-Styles FeedForward Network

Although the work by [**?**] are much faster than descriptive methods, their limitations are also obvious: each generative network is trained for a single style, which means that we need to train multiple networks for different styles, so it is time consuming and not flexible. Based on the observation, [**?**] proposed an algorithm to train a conditional style transfer network for multiple styles. Their work "stems from the intuition that many styles probably share some degree of computation, and that this sharing is thrown away by training N networks from scratch when building an Nstyles style transfer system." By using their method and tuning parameters of an conditional instance normalization, we "can stylize a single image into N painting styles with a single feed forward pass of the network with a batch size of N."

3 Method

We propose two different approaches to data augmentation. The first approach is Traditional transformations. Traditional transformations consist of using a combination of affine transformations to manipulate the training data. For each input image, we generate a âĂİduplicateâĂİ image that is shifted, zoomed in/out, rotated, flipped, distorted, or shaded with a hue. Both image and duplicate are fed into the neural net. The second approach Neural Style Transfer. For each input image, we select a style image from a subset of different styles from famous artists: wave, scream, rain-princess. A styled transformation of the original image is generated. Both original and styled image are fed to train the net.

<A image of the Traditional transformations should be put here>

3.1 Datasets and Features

There are lots of image dataset. Since we need to do some analysis in the future, the image quality should be good. Here are some popular dataset avaliable online.

- Caltech 256: http://www.vision.caltech.edu/Image_Datasets/Caltech256/ (1.2 GB. image size/quality is fine)

- Caltech 101: http://www.vision.caltech.edu/Image_Datasets/Caltech101/ (131MB, image quality)

- CIFAR10 / CIFAR100: image size is too small (32*32)

- ImageNet(more than 100GB. select specific classes: how to select would be a problem)
- Tiny ImageNet(image Size is too small 64*64)
- Pascal VOC2007 and VOC2012: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html(around 2 GB)
- Coil-100/Coil-20: images are too small.

Conclusion: The first dataset can be Caltech101, which is a small but effective dataset. This dataset has been tested in the original VggNet Paper. I can take it as my first dataset and do some experiment. I can also do some testing in Caltech256 and VOC2007/2012 after I get some results from Caltech101.

3.2 Baseline Network

The first model I use will be vgg16. Then I can also try vgg19. The output would look like:

Model	**Dataset**	**result**
Vgg16	Caltech101	...
Vgg19	Caltech101	...
Vgg16	Caltech256	...
Vgg19	Caltech256	...

3.3 Neural Style Transfer with Traditional Transformations

The implementation follows [Engstrom, 2016], which uses roughly the same transformation network as described in Johnson, except that batch normalization is replaced with Ulyanov's instance normalization, and the scaling/offset of the output tanh layer is slightly different. We use a loss function close to the one described in Gatys, using VGG19 instead of VGG16 and typically using "shallower" layers than in Johnson's implementation. Empirically, this results in larger scale style features in transformations.

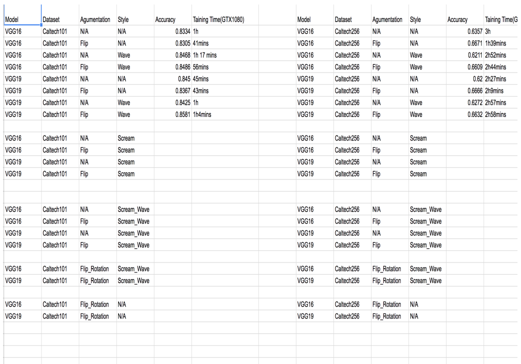
<A picture of the network archtecture should be put here>

4 Experiments and Results

5 Introduction

6 Conclusion

Data augmentation has been shown to produce promising ways to increase the accuracy of classiïňĄcation tasks. While traditional augmentation is very effective alone, other techniques enabled by CycleGAN and other similar networks are promising. We experimented with our own way of combining training images allowing a neural net to learn augmentations that best improve the ability to correctly classify images. If given more time, we would like to explore more complex architecture and more varied datasets. To mimic industrial applications, using a VGG16 instead of SmallNet can help us determine if augmentation techniques are still helpful given complex



Model	Dataset	Agumentation	Style	Accuracy	Taining Time(GTX1080)		Model	Dataset	Agumentation	Style	Accuracy	Taining Time(G
VGG16	Caltech101	N/A	N/A	0.8334	1h		VGG16	Caltech256	N/A	N/A	0.6357	3h
VGG16	Caltech101	Flip	N/A	0.8305	41mins		VGG16	Caltech256	Flip	N/A	0.6671	1h39mins
VGG16	Caltech101	N/A	Wave	0.8468	1h 17 mins		VGG16	Caltech256	N/A	Wave	0.6211	2h52mins
VGG16	Caltech101	Flip	Wave	0.8486	56mins		VGG16	Caltech256	Flip	Wave	0.6609	2h44mins
VGG19	Caltech101	N/A	N/A	0.845	45mins		VGG19	Caltech256	N/A	N/A	0.62	2h27mins
VGG19	Caltech101	Flip	N/A	0.8367	43mins		VGG19	Caltech256	Flip	N/A	0.6666	2h9mins
VGG19	Caltech101	N/A	Wave	0.8425	1h		VGG19	Caltech256	N/A	Wave	0.6272	2h57mins
VGG19	Caltech101	Flip	Wave	0.8581	1h4mins		VGG19	Caltech256	Flip	Wave	0.6632	2h58mins
VGG16	Caltech101	N/A	Scream				VGG16	Caltech256	N/A	Scream		
VGG16	Caltech101	Flip	Scream				VGG16	Caltech256	Flip	Scream		
VGG19	Caltech101	N/A	Scream				VGG19	Caltech256	N/A	Scream		
VGG19	Caltech101	Flip	Scream				VGG19	Caltech256	Flip	Scream		
VGG16	Caltech101	N/A	Scream_Wave				VGG16	Caltech256	N/A	Scream_Wave		
VGG16	Caltech101	Flip	Scream_Wave				VGG16	Caltech256	Flip	Scream_Wave		
VGG19	Caltech101	N/A	Scream_Wave				VGG19	Caltech256	N/A	Scream_Wave		
VGG19	Caltech101	Flip	Scream_Wave				VGG19	Caltech256	Flip	Scream_Wave		
VGG16	Caltech101	Flip_Rotation	Scream_Wave				VGG16	Caltech256	Flip_Rotation	Scream_Wave		
VGG19	Caltech101	Flip_Rotation	Scream_Wave				VGG19	Caltech256	Flip_Rotation	Scream_Wave		
VGG16	Caltech101	Flip_Rotation	N/A				VGG16	Caltech256	Flip_Rotation	N/A		
VGG19	Caltech101	Flip_Rotation	N/A				VGG19	Caltech256	Flip_Rotation	N/A		

Figure 1: Results.

enough networks that already deal with many overïňĄtting and regularization problems. Finally, although GANs and neural augmentations do not perform much better than traditional augmentations and consume almost 3x the compute time or more, we can always combine data augmentation techniques. Perhaps a combination of traditional augmentation followed by neural augmentation further improves classiïňĄcation strength.

Given the plethora of data, we would expect that such data augmentation techniques might be used to beneïňĄt not only classiïňĄcation tasks lacking sufïňĄcient data, but also help improve the current state of the art algorithms for classiïňĄcation. Furthermore, the work can be applicable in more generic ways, as âĂİstyleâĂİ transfer can be used to augment data in situations were the available data set is unbalanced. For example, it would be interesting to see if reinforcement learning techniques could beneïňĄt from similar data augmentation approaches. We would also like to explore the applicability of this technique to videos. SpeciïňĄcally, it is a well known challenge to collect video data in different conditions (night, rain, fog) which can be used to train selfdriving vehicles. However, these are the exact situations under which safety is the most critical. Can our style transfer method be applied to daytime videos so we can generate night time driving conditions? Can this improve safety? If such methods are successful, then we can greatly reduce the difïňĄculty of collecting sufïňĄcient data and replace them with augmentation techniques, which by comparison are much more simpler.

References

[Engstrom, 2016] Engstrom, L. (2016). Fast style transfer. `https://github.com/lengstrom/fast-style-transfer/`. commit xxxxxxx.

[Fei-Fei et al., 2006] Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

[Gatys et al., 2016] Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE.

[Griffin et al., 2007] Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kyprianidis et al., 2013] Kyprianidis, J. E., Collomosse, J., Wang, T., and Isenberg, T. (2013). State of the" artâĂİ: A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics*, 19(5):866–885.

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