Data augmentation with Artistic Style

Anonymous Submission

Anonymous Affiliation

Abstract

This paper presents a novel approch to use the image style transfer by CNN as a data augmentation strategy for 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 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 build a novel approch to apply it on the current image classification algotithms we compare to and combine with the the traditional approaches to show the effectiveness of this method.Pre-trained Vgg16 and Vgg19 are the baseline network.architecture.

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

1 Introduction

Deep Convolutional Neural Networks (CNN) have grown in popularity for performing image processing tasks like image classification, object detection and segementation. In the state-of-art network architechters, many different kinds of data augmentation strategies have been used and found effective to improve the performance. 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 [**?**].

The motivation for this problem is both broad and speciïňĄc. Specialized image and video classiïňĄcation tasks often have insufïňĄcient data. This is particularly true in the medical industry, where access to data is heavily protected due to privacy concerns. Important tasks such as classifying cancer types [14] are hindered by this lack of data. Techniques have been developed which combine expert domain knowledge with pre-trained models. Similarly, small players in the AI industry often lack access to signiïňĄcant amounts of data. At the end of the day, weâĂŹve realized a large limiting factor for most projects is access to reliable data, and as such, we explore the effectiveness of distinct data augmentation techniques in image classiïňĄcation tasks.

The datasets we examine are the caltech101 dataset and caltech256. Caltech101 consists of 9000 images and 102 categories, while caltech256 contains 257 categories and ***** images. To evaluate the effectiveness of augmentation techniques, we split both data set as 70% of images are used for training and 30% of images are used for validation.

We will apply the pre-trained vgg16 and vgg19 to perform a rudimentary classification. Troditional data augmentation techniques will be firstly used for, and retrain our models. Next, we will make use of CycleGAN [19] to augment our data by transferring styles from images in the dataset to a ïňĄxed predetermined image such as Night/Day theme or Winter/Summer. Finally, we explore and propose a different kind of augmentation where we combine neural nets that transfer style and classify so instead of standard augmentation tricks, the neural net learns augmentations that best re-

1 duce classiïňĄcation loss. For all the above, we will measure classiïňĄcation performance on the validation dataset as the metric to compare these augmentation strategies.

2 Related Work

This section provides a brief review of past work that has augmented data to improve image classiïňĄer performance and the state of the art techniques of neural style transfer.

2.1 Troditional Data Augmentation Tecniques

2.1.1 AlexNet

AlexNet from [?] 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.

2.1.2 VGGNet

VGGNet is a simple but deep model created by [?]. 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

Method. How to keep high level content. how to apply styles. the formulars.

3.1 Datasets and Features

3.2 Baseline Network

We propose two different approaches to data augmentation. The ïňĄrst approach is generate augmented data before training the classiïňĄer. For instance, we will apply GANs and basic transformations to create a larger dataset. All images are fed into the net at training time and at test time, only the original images are used to validate.

3.1. Traditional Transformations Traditional transformations consist of using a combination of afïňĄne transformations to manipulate the training data [9]. For each input image, we generate a âĂİduplicateâĂİ image that is shifted, zoomed in/out, rotated, ïňĆipped, distorted, or shaded with a hue. Both image and duplicate are fed into the neural net. For a dataset of size N, we generate a dataset of 2N size.

3.2. Neural Style Transfer For each input image, we select a style image from a subset of 6 different styles: Cezanne, Enhance, Monet, Ukiyoe, Van Gogh and Winter. A styled transformation of the original image is generated. Both original and styled image are fed to train the net. More detail about the GANs and style transfer can be viewed on the cited paper [19].

Figure II: Neural Style Transfer via CNN

More details about the architecture of the layers will be described in the Experiments section. We implement a small 5-layer CNN to perform augmentation. The classiïňĄer is a small 3-layer net with batch normalization and pooling followed by 2 fully connected layers with dropout. This is much similar to VGG16 in structure but smaller in interest of faster training for evaluation. We arenâĂŹt aiming for the best classiïňĄer. We are exploring how augmentation tricks improve classiïňĄcation accuracy, reduce overïňĄtting, and help the networks converge faster.

3.3 Neural Style Transfer with Traditional Transformations More details about the architecture of the layers will be described in the Experiments section. We implement a small 5-layer CNN to perform augmentation. The classiïňĄer is a small 3-layer net with batch normalization and pooling followed by 2 fully connected layers with dropout. This is much similar to VGG16 in structure but smaller in interest of faster training for evaluation. We arenâĂŹt aiming for the best classiïňĄer. We are exploring how augmentation tricks improve classiïňĄcation accuracy, reduce overïňĄtting, and help the networks converge faster.

4 Experiments and Results

There are three sets of images that we experimented with. Each dataset is a small dataset with two classes. A small portion of the data is held aside for testing. The remaining images are divided by a 80:20 split between training and validation.

Our ïňĄrst data set is taken from tiny-imagenet-200. We take 500 images from dogs and 500 images from cats. 400 images for each class is allocated to the training set. The remaining 100 in each class forms the validation set. The images are 64x64x3. RGB values are also normalized for each color in the preprocessing step.

The second data set is also taken from tiny-imagenet200 except we replace cats with goldïňĄsh. The reason for this change is that goldïňĄsh look very different from dogs whereas cats visually are very similar. Hence CNNs tend to have a harder time distinguishing cats. Finally, cats and dogs have similar styles whereas images from the goldïňĄsh tend to have very bright orange styles.

Lastly, the ïňĄnal dataset is 2k images from MNIST, 1000 from each class. We perform the task of distinguishing 0âĂŹs from 8âĂŹs. MNIST images are 28x28x1 and are in gray scale. Again, images are normalized in the preprocessing step. MNIST is much more structured than imagenet so that digits are always centered. The motivation is that MNIST provides a very simple dataset with simple images. Are patterns in the more complex images also observed in simpler images?

To test the effectiveness of various augmentation, we run 10 experiments on the imagenet data. The results of the experiments are tabulated in the following table. All experiments are run for 40 epochs at the learning rate of 0.0001 using Adam Optimization. The highest test accuracy at all the epochs is reported as the best score. Once we obtained the augmented images, we feed them into a neural net that does classiïňĄcation. We name this neural net the SmallNet since it only has 3 convolutional layers paired with a batch normalization and max pool layer followed by 2 fully connected layers. The output is a score matrix for the weights for each class. The layers of the network is detailed below although the speciïňĄc net is not very important. Any net that can reliably predict the classes sufïňĄces. Hence, one can replace this net with VGG16 with ïňĄne-tuning on the fully connected and last convolution layers to allow for sufïňĄcient training. Consider the variable $x \in \mathbb{R}$,

$$f(x) = x^2 + 2 \tag{1}$$

Equation (1) is a polynomial of order 2.

An example of table is shown Table 1.

x	y	z

Table 1: Example of table.

5 Conclusion

Data augmentation has been shown to produce promising ways to increase the accuracy of classification 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

enough networks that already deal with many overfitting 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 classification strength.

Given the plethora of data, we would expect that such data augmentation techniques might be used to benefit not only classification tasks lacking sufficient data, but also help improve the current state of the art algorithms for classification. 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 benefit from similar data augmentation approaches. We would also like to explore the applicability of this technique to videos. Specifically, 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 difficulty of collecting sufficient data and replace them with augmentation techniques, which by comparison are much more simpler.

References

[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.