Effectiveness of Style Transfer as a Data Augmentation

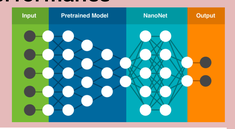
Jason Wang, Luis A. Perez
CS231n: CNNs for Visual Image Recognition, Stanford University

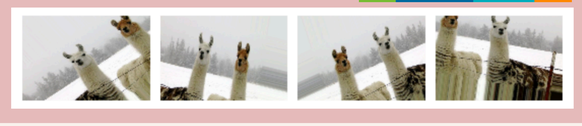
Background

We explore the effectiveness of "data transfiguration" as a data augmentation technique for image classification tasks. ML models are often limited by the amount of task data available, such as self driving cars lacking sufficient nighttime driving data or small startups lacking enough customer data. The lack of data can lead to poor performance

Current Methods:

- Transfer learning
- Simple data augmentation





Problem

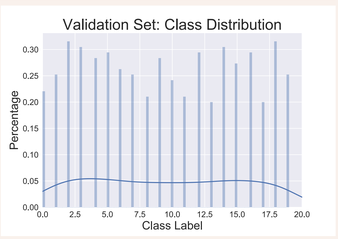
We tackle the problem of data augmentation with a novel technique, possible due to advances in Generative Adversarial Networks (GANs), which we call "data transfiguration".

We artificially mimic a reduced data set and compare previous data augmentation techniques as well as transfer learning, to our new method of data transfiguration.

We evaluate the performance of VGGNet, ResNet, and others on a constrained image classification task for a fixed number of epochs with each type of data augmentation.

Datasets

We use a subset of the ImageNet dataset consisting of 20 distinct classes with 251 image samples per task for 5020 total images, of size 64x64x3. We augment the data set using six distinct styles.



Methods

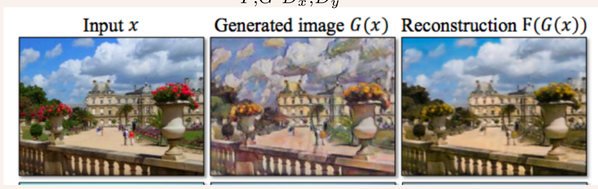
We begin by training a GAN for unpaired image to image translation, as described in [1]. The main idea is to enforce "cycle-consistency" on the generator network given unpaired images. We follow [1] closely.

$$\mathcal{L}_{\text{cyc}}(G,F) = \mathbb{E}_{x\sim p_{\text{data}(x)}}\|F(G(x)) - x\|_1 + \mathbb{E}_{y\sim p_{\text{data}(y)}}\|G(F(x)) - y\|_1$$

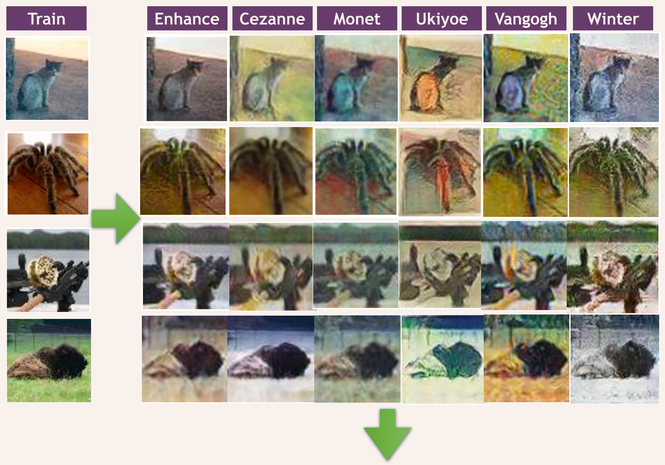
$$\mathcal{L}_{\text{GAN}}(G,D) = \mathbb{E}_{x\sim p_{\text{data}(x)}}\log(1 - D(G(x)) + \mathbb{E}_{y\sim p_{\text{data}(y)}}\log(D(y))$$

$$\mathcal{L}(G,F,D_x,D_y) = \mathcal{L}_{\text{GAN}}(G,D_x) + \mathcal{L}_{\text{GAN}}(F,D_y) + \lambda\mathcal{L}_{\text{cya}}(G,F)$$

$$G^*, F^* = \arg\min_{F,G}\max_{D_x,D_y}\mathcal{L}(G,F,D_x,D_y)$$



With our network trained, we can now perform augmentation on our dataset. We take each input image and generate six new images from it, each augmented by one of six generators G.



VGGNet - (C5 + FC)
AlexNet - FC
AlexNet - (C5 + FC)

Preliminary Results

We evaluate the success of style transfer by feeding the style transfer augmented dataset and the standard augmentation into a second network. The networks we evaluated were VGG16 with various number of layers untrained and a small fully untrained network.

Normal	Transfiguration

Validation Accuracy		
Model-layers trained	Original Data	Augmentation
AlexNet-FC	**0.928**	0.924
Alex-C5+FC	0.552	**0.689**
VGG-C5+FC	0.601	**0.748**

Figure III: Best Val Accuracy for all Models

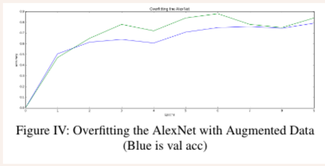


Figure IV: Overfitting the AlexNet with Augmented Data (Blue is val acc)

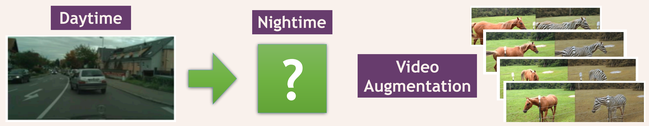
WIP

Future Work

Current results are promising. We see an increase with the augmented data set, which can match the performance of normal augmentation techniques. Our starting hypothesis that "data transfiguration" using CycleGAN and similar style transfer techniques would be effective has proven correct.

Next Steps:

- Experiment with video classification
- Test techniques in RL settings (self-driving cars)



References

1. J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR, abs/1703.10593, 2017.
2. Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin. Improved relation classification by deep recurrent neural networks with data augmentation. CoRR, abs/1601.03651, 2016
3. S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. Understanding data augmentation for classification: when to warp? CoRR, abs/1609.08764, 2016.
4. C.N. Vasconcelos and B. N. Vasconcelos. Increasing deep learning melanoma classification by classical and expert knowledge based image transforms. CoRR, abs/1702.07025, 2017.
5. Finetune alexnet with tensorflow 1.0: https://github.com/kratzert/finetune_alexnet_with_tensorflow. Accessed 2017-05-19.