Training Neural Networks to be More Biased Towards Shape than Texture

Mir Tanvir Islam
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
tanvir.islam04@northsouth.edu

Shafin T.Mashfu
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
shafin.mashfu@northsouth.edu

***Abstract*—In the field of computer vision with neural networks, a widely accepted notion about Neural Networks is that it recognizes objects by mainly detecting the shape of the objects. However, according to some recent studies on Convolutional Neural Networks, image textures might play a more critical part. A stylized image dataset was created and compared with the performance of the CapsuleNet-based image identification network on the stylized images and original images. In this study, the same architecture was trained to learn shape-based recognition using two other variant of the CIFAR and ImageNet datasets called 'Stylized-CIFAR' and 'Stylized-ImageNet'. This allows the CapsuleNet to perform more in accordance with the recognition strategies that human observers use. Learning shape-based recognition allows it to analyze a wide array of image distortions and draw out robustness that was previously unseen. Overall, object detection performance was also improved through shape-based recognition.**

***Index Terms*—AdaIN, MNIST, CapsuleNet**

I. Introduction

Convolutional Neural Networks(CNNs) provide us with an impressive performance on several complex perceptual tasks; such as object recognition (Krizhevsky et al., 2012) and semantic segmentation (Long et al., 2015). One way to decipher this process is to assume that CNNs start with low-level features (e.g. edges). Then it moves on to more complicated shapes until it captures the entire complex image of the object. According to Kriegeskorte (2015), "the network acquires complex knowledge about the kinds of shapes associated with each category. [...] High-level units appear to learn representations of shapes occurring in natural images" (p. 429). This idea is also supported by some other explanations, such as in LeCun et al. (2015): Intermediate CNN layers recognise "parts of familiar objects, and subsequent layers [...] detect objects as combinations of these parts" (p. 436). For the purpose of this study. We termed this theory the shape hypothesis.

We can find many empirical studies that support this hypothesis. Perception techniques like Deconvolutional Networks(Zeiler Fergus,2014) usually outline object parts in advanced CNN features. Kubilius et al. (2016) goes one step further and proposes CNNs as computational models of human shape perception. After conducting numerous experiments comparing the ability of humans and CNNs regarding shape representations, he concluded that CNNs "implicitly learn representations of shape that reflect human shape perception" (p. 15). Ritter et al. (2017) discovered that CNNs develop a preference towards shapes just like toddlers and primarily focus on object shape rather than colour for object recognition. Moreover, CNNs are the most predictive models available for replicating human ventral stream object recognition (e.g. Cadieu et al., 2014; Yamins et al., 2014); and it is established knowledge that object shape is the most significant cue for humans for recongnizing (Landau et al., 1988), much more than other cues like color, size or texture. On the other hand, some studies deem object textures a more important indicator for CNN object recognition. These findings suggest that CNNs can classify texturised images with optimal efficiency, even if the shape structure is completely destroyed (Gatys et al., 2017; Brendel Bethge, 2019). Conversely, standard CNNs are bad at classifying images of objects where the shapes remain intact but the texture cues are unavailable (Ballester deAra´ujo, 2016). To shed some more light on this subject, two other findings implicate that local information such as textures may actually be all that is required to achieve ImageNet object recognition. Gatys et al. (2015) discovered that putting a linear classifier on top of a CNN's texture representation does not result in decline in its ability to classify compared to the original network performance. Recently, Brendel Bethge(2019) demonstrated that CNNs with somewhat constrained receptive fields are able to classify objects accurately on ImageNet, even though this restrains a model's ability to integrate object parts for recognizing shapes. Rather, it picks up small local patches and suggests that this data is enough for precisely completing its task. When we take all of these findings into consideration, it becomes apparent that local textures can communicate sufficient information to CNNs about object classes. In theory, it is possible that texture recognition alone is sufficient for an object recognition. The findings derived from these studies directly contradict

the more commonly accepted shape hypothesis. Therefore, we need to consider these theories as a second explanation, termed as the texture hypothesis. In order to resolve these two contrasting hypotheses, we aim to delve deep into this debate with a number of methodically designed quantitative analysis. For this purpose, we created some images with a texture-shape cue conflict. This enabled us to identify the biases among CNNs and human observers regarding texture and shape. These experiments revealed behavioral patterns in favour of the texture hypothesis: A rabbit with a tiger texture is a tiger to CNNs, and still a rabbit to humans. Our study surpasses defining existing biases and presents results for our two other objectives: shifting these biases by training CNNs on Stylized ImageNet, and exploring the advantages of shifting biases. We show that the texture bias in standard CNNs can be modified to pick up shape-based representations over texture-based representations. Our findings also implicate that networks that lean towards a shape bias are far more likely to be robust to a vast range of image distortions and nearly perfect(or surpass) human-like object classification.

II. Methods

The following datasets were used in this experimentation:

1. Original MNIST: The Original MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST

Fig. 1. Original MNIST Dataset Samples

2. Stylized MNIST: We created a stylized version of the MNIST database. For each training example, 5 stylized examples were created from 5 stylize images. The Stylized MNIST database has a training set of 300,000 examples. We created two variations of the Stylized MNIST one with all gray-scale images and one with colored images. We created this dataset (Stylized-MNIST or SIM), by using AdaIN style transfer removes each images' original texture and modifies the image based on the artistic style of the specified stylze image. AdaIN was used over other stylization techniques due of two reasons: to ensure that the training on the SIM and the testing on cue conflict stimuli are carried out through different stylization techniques; instead of iterative stylizing (eg. Gatys et al., 2016) so that the findings do not fall under a particular form of stylization. Secondly, AdaIN can created stylized images faster than other techniques. To create the entire stylization of Original MNIST, it would take a long time for an iterative approach found in other techniques.

Fig. 2. Stylize images

Fig. 3. Stylized MNIST samples

3. Original + Stylized MNIST: We combined the Original MNIST database and the colored Stylized MNIST database. The Original + Stylized MNIST database has a training set of 360,000 examples.

For this experimentation, the testing set is the 10,000 test set examples from the Original MNIST dataset.

III. CapsuleNet models and training details

We used the standard Capsule Net architecture with Dynamic Routing [1]. Our implementation was carried out using PyTorch framework. We Trained all three datasets for 24,000 iterations. The datasets were trained for the same number of iterations, as the size of the datasets are different, therefore training for same no. of epochs would result in training for uneven no. of iterations. We logged the train and test metrics every 500 iterations. For our experimentation we trained with Adam gradient descent using a momentum term of 0.9 and a learning rate of 0.01. We used a batch size of 200. The network took images with 3 input channels and had 16 output

TABLE I
NAME OF DATASET ALONG WITH THEIR COLOR CODES

Color Code	Dataset
	CapsuleNet MNIST batch_size=256 lr=0.001 num_routing=3 lr_decay=0.96/runs
	CapsuleNet MNIST-stylized batch_size=256 lr=0.001 num_routing=3 lr_decay=0.96/runs
	CapsuleNet MNIST-stylized_rgb batch_size=256 lr=0.001 num_routing=3 lr_decay=0.96/runs
	CapsuleNet MNIST-original&stylized batch_size =256 lr=0.001 num_routing=3 lr_decay=0.96/runs

dimensions. The 8 Primary Capsules had 256 input channels and 32 output channels. CapsuleNet also reconstructs the input image from the data encoded in the 16 output dimensions. CapsuleNet with dynamic routing contains two loss metrics - margin loss (for calculating loss due to misclassifications) and reconstruction loss (for calculating loss in difference between reconstructed images and original image). Total loss is combined margin loss and reconstruction loss.

Here are the graphs for Train set Accuracy, Total Loss, Margin Loss and Reconstruction Loss:

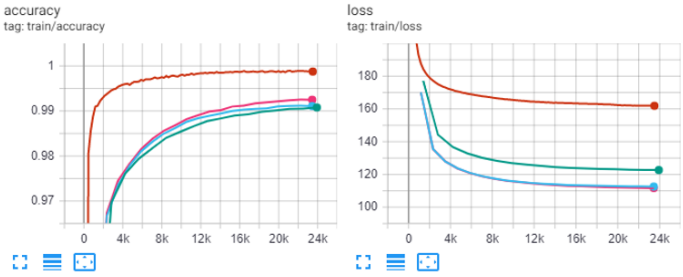

Fig. 4. Train set Accuracy and Total Loss

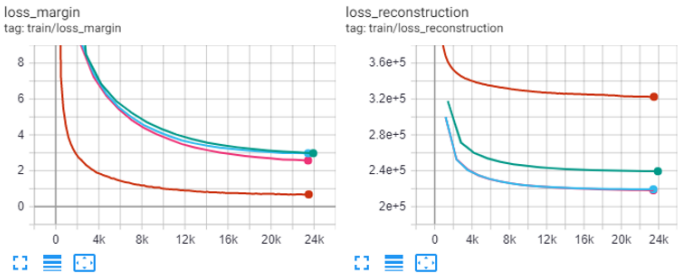

Fig. 5. Train set Margin Loss and Reconstruction Loss

In the training set, the CapsuleNet trained on Original MNIST reaches an accuracy of about 0.998, the CapsuleNet trained on gray-scale Stylized MNIST reaches an accuracy of about 0.99, the CapsuleNet trained on colored Stylized MNIST reaches an accuracy of about 0.992, the CapsuleNet trained on Original+Stylized MNIST reaches an accuracy of about 0.99.

Here are the graphs for Test set Accuracy, Total Loss, Margin Loss and Reconstruction Loss:

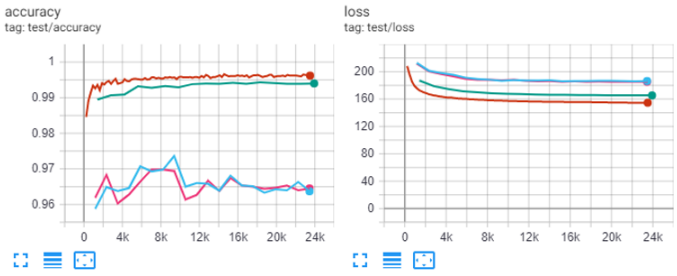

Fig. 6. Test set Accuracy and Total Loss

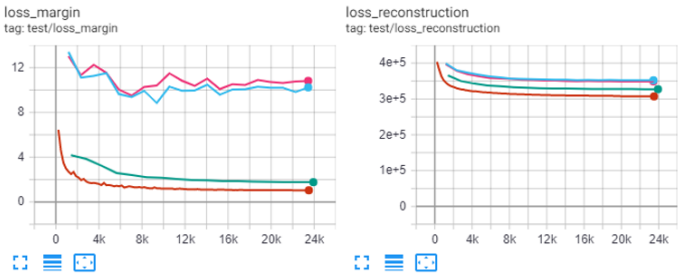

Fig. 7. Test set Margin Loss and Reconstruction Loss

In the testing set, the CapsuleNet trained on Original MNIST reaches an accuracy of about 0.996, the CapsuleNet trained on gray-scale Stylized MNIST reaches an accuracy of about 0.97, the CapsuleNet trained on colored Stylized MNIST reaches an accuracy of about 0.97, the CapsuleNet trained on Original+Stylized MNIST reaches an accuracy of about 0.994.

In our experimentation, we have also tested how well the networks can predict on reconstruct composite images. The composite images contain two samples overlayed on one another. Geofrey et. al. shifted the images in the horizontal and vertical axis, here we have not done so. // Here are a few examples of the test for Original MNIST (1st row contains the composite image, 2nd and 3rd row contains the reconstruction of the top 2 predictions): Here are a few examples of the test

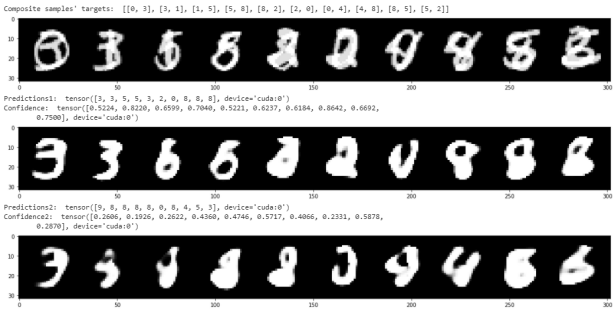

Fig. 8. Composite image prediction and reconstruction of Original MNIST

for Stylized MNIST (1st row contains the composite image, 2nd and 3rd row contains the reconstruction of the top 2 predictions): Here are a few examples of the test for Original

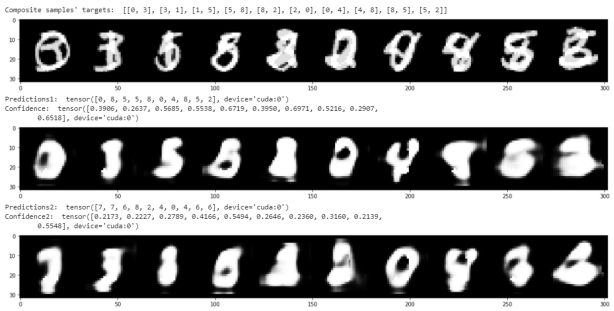

Fig. 9. Composite image prediction and reconstruction of MNIST stylized RGB

+ Stylized MNIST (1st row contains the composite image, 2nd and 3rd row contains the reconstruction of the top 2 predictions): We have also tested how the 16 dimensions affect

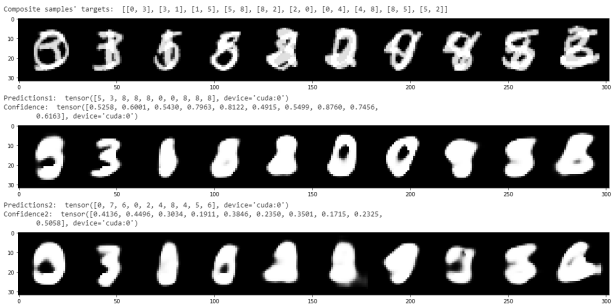

Fig. 10. Composite image prediction and reconstruction of MNIST Original Stylized

the CapsuleNet. We have observed that each of the dimensions control certain features of the image, for example: thickness of the strokes, skewness, scaling etc. Modifying the dimensions manually brings change to the reconstruction image. We have relatively changed each dimension by [-1, -0.5, -0.25, -0.15, -0.1, 0, 0.1, 0.15, 0.25, 0.5, 1].

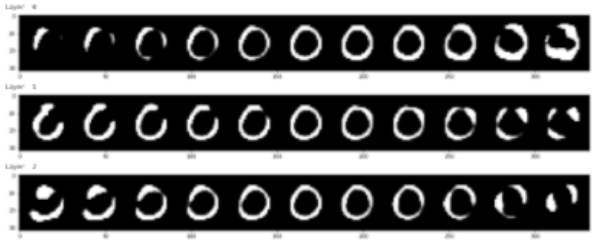

Fig. 11. Reconstructed image for changing a dimension of Original MNIST

RESULTS

Original + Stylized MNIST has accuracy better than Stylized MNIST datasets. The accuracy of Original + Stylized MNIST comes close to the accuracy of Original MNIST, the train loss is lower in lower in Original + Stylized MNIST, test loss is almost as low as that of the Original MNIST dataset. At 500 and 1000 iterations of Gray-scale Stylized

Fig. 12. Reconstructed image for changing a dimension of Stylized MNIST

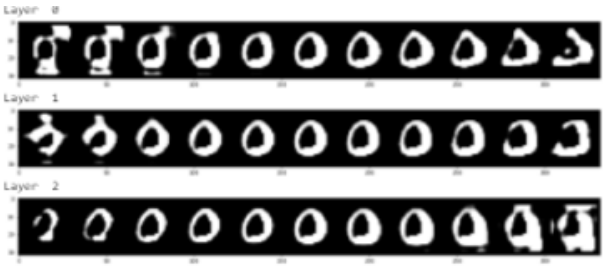

Fig. 13. Reconstructed image for changing a dimension of Original + Stylized MNIST

MNIST and colored Stylized MNIST, the test accuracy was about 4% higher than the train set (approx. 96% compared to 92%), after a few thousand iterations, the test accuracy was relatively lower than train accuracy. Notice that for Gray-scale Stylized MNIST and colored Stylized MNIST datasets, after a few iterations the margin loss is unstable but reconstruction loss slowly keeps decreasing, this can be further visualized by observing the reconstructed images. We also observe that training loss of Gray-scale Stylized MNIST and colored Stylized MNIST datasets are lower, however test loss of Original MNIST dataset is lower.

Original vs. Reconstructions after 23000 iterations (100 epochs of Original MNIST dataset)

The per class accuracy of Original MNIST, Stylized MNIST and Original + Stylized MNIST are give below:

We have also observer that modifying the dimension of the last layer changes feature of the reconstructed image.

DISCUSSION

This experimentation was carried out with the MNIST dataset, which is a dataset of black and white 28x28 images. AdaIN Stylization has more impact in colored photos of high resolution. We aim to carry out the same experimentation with a dataset of high resolution images.

CONCLUSIONS

In conclusion, the evidence contradicts the common assumption that machine recognition relies on object shapes. According to our findings and as demonstrated in this study, present day machine recognition in fact favors object textures rather than object shapes. Using our Stylized datasets, we observed that they achieved accuracy similar to the original datasets. This provides a deeper understanding of how neural

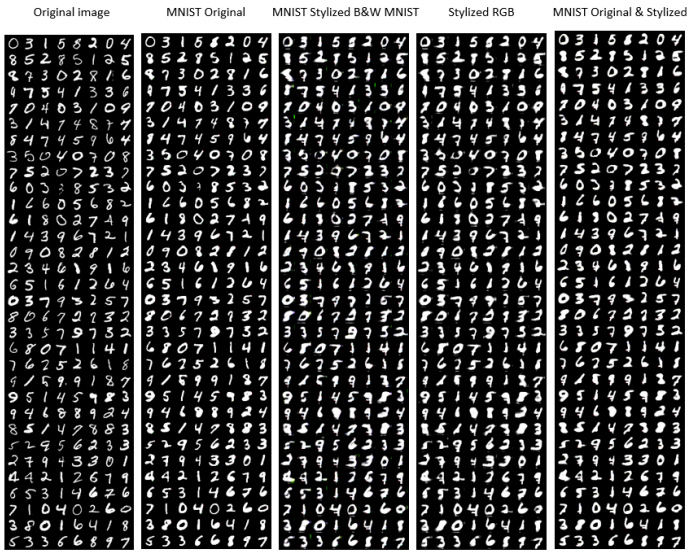

Fig. 14. Original vs. Reconstructions after 23000 iterations

TABLE II
TABLE OF PER CLASS ACCURACY

Per class Accuracy	**MNIST stylized greyscale**	**MNIST stylized RGB**	**MNIST Original & stylized**
Accuracy of 0 (0):	96.63% (947/980)	97.45% (955/980)	99.90% (979/980)
Accuracy of 1 (1):	99.74% (1132/1135)	99.74% (1132/1135)	99.56% (1130/1135)
Accuracy of 2 (2):	98.16% (1013/1032)	98.06% (1012/1032)	99.61% (1028/1032)
Accuracy of 3 (3):	98.81% (998/1010)	98.02% (990/1010)	99.31% (1003/1010)
Accuracy of 4 (4):	99.19% (947/982)	99.29% (975/982)	99.59% (978/982)
Accuracy of 5 (5):	99.22% (885/892)	99.55% (888/892)	99.33% (886/892)
Accuracy of 6 (6):	94.68% (907/958)	94.57% (906/958)	99.06% (949/958)
Accuracy of 7 (7):	94.94% (976/1028)	96.21% (989/1028)	99.03% (1018/1028)
Accuracy of 8 (8):	89.84% (875/974)	90.14% (878/974)	99.49% (969/974)
Accuracy of 9 (9):	92.1% (930/1009)	91.18% (920/1009)	99.11% (1000/1009)

network representations actually work and the nuances surrounding it.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet classification with deep convolutional neural networks." Commun. ACM 60, 6 (June 2017), 84–90.

[2] LeCun, Y., Bengio, Y. Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

[3] Geirhos, Robert Rubisch, Patricia Michaelis, Claudio Bethge, Matthias Wichmann, Felix Brendel, Wieland. (2018). "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." ICLR 2019 Conference

[4] Christina M Funke, Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "Synthesising dynamic textures using convolutional neural networks." arXiv preprint arXiv:1702.07006, 2017.