ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness

Andre Diler[1] , **Mehdi Chaid**[1] , **Abderahmane Bouziane**[1]
[1]Département GIGL Polytechnique Montreal
andre.diler@polymtl.ca, mehdi.chaid@polymtl.ca, abderahmane.bouziane@polymtl.ca

Abstract

The history of computer vision led us to believe that like humans, Convolutional Neural Networks (CNNs) would recognize objects mainly by their shapes. Recent studies in the field however, have suggested that CNNs rely more on image textures rather than edges and shapes in order to perform object detection. A paper published in November 2018, by [Geirhos *et al.*, 2018], explored the idea of texture bias and presented novels solutions in order to shift the trend towards a stronger shape bias for CNNs, similar to how humans perceive things. The following report attempts to analyse the hypothesis proposed in the paper, as well as offer a scoped reproduction of the experiments conducted by the authors, in a smaller environment, in order to draw new conclusions and reinforce our understanding of the internals of CNNs.

1 Introduction

Modern Convolutional Neural Networks regularly reach very high performances on complex computer vision tasks such as image classification and segmentation. These performances are reaching human levels in term of accuracy, and come from a long history of studies on human and machine perception.

As such, and reinforced with other experiments on the matter, it is commonly believed that CNNs learn features from the shapes during the training phase and use these for detection.

1.1 Related Work

The first proof that shapes are fundamental in computer vision came from a very influential study on cognition, by [Hubel and Wiesel, 1959], who described how biological neurons could extract features from images, amongst which certain types of neurons were activated specifically by edges.

Another paper from [Marr, 1982] concluded that vision was hierachical. Low-level features, such as lines, were combined together to recognize more high level concepts, like wheels, windows, etc, and form objects.

The famous Neocognition paper [Fukushima, 1980] was the implementation that introduced the idea of hierachical vision. The multilayered neural network they proposed included multiple convolutional layers with wheighted receptive fields (filters). It was the first known deep neural network.

However, the first modern convnet was LeNet [Lecun *et al.*, 1998]. This CNN used backpropagation to automatically learn the filter values to extract meaningful features in images hierachically. Nowadays, all the recent convolutional neural networks are inspired from this network.

ConvNet were considered like a black box for a long time, resulting in considerable efforts made in recent years to analyze the inside of these networks.

The shape hypothesis that emerged from early experiments consists in ”High-level units appear to learn representations of shapes occurring in natural images” [Kriegeskorte, 2015]. This theory quickly became widespread in the community, and is understandable given the history of computer vision.

Furthermore, there are a lot of studies comparing human vision with computer vision. For example, [Kubilius *et al.*, 2016] stated that ”implicitly learn representations of shape that reflect human shape perception” while a paper from [Ritter *et al.*, 2017] concluded that ”state-of-the-art one-shot learning models trained on ImageNet exhibit a similar bias to that observed in humans: they prefer to categorize objects according to shape rather than color” .

However, several researchers raised doubts about the shape hypothesis. According to studies from [Gatys *et al.*, 2017] and [Brendel and Bethge, 2019], CNNs are able to classify texturized images even if their shape structure is destroyed. Moreover, the same paper from [Brendel and Bethge, 2019] also showed that CNNs with constrained receptive field sizes can reach competitive accuracies on ImageNet. It is also worth noticing that small receptive fields cannot capture the overal shape of an image.

These results have led the authors of the paper we've analysed, [Geirhos *et al.*, 2018], to emit a new hypothesis: the texture hypothesis, where ”in contrast to the common assumption, object textures are more important than global object shapes for CNN object recognition”.

Our objective in this report is to submit those hypothesis to a test through diverse experiments and validate the results obtained by the original authors in our own environment .

2 Methodology

2.1 Dataset

We used Imagenette [FastAi, 2019], a subset of the Imagenet dataset created by fastai to conduct our experiments. This ensured that our results would be comparable to the authors, while keeping an acceptable size for the dataset given the time and material constraints. Imagenette contains 13394 images of 10 easily classifiable classes from Imagenet, namely *tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute*. The classes are balanced with around 950 images in each one.

This dataset is more easily classifiable than Imagenet in less time, which fits our purpose. We used the '160px' version of the images, where the shortest size of the images are resized to that size, with their aspect ratio maintained.

2.2 Style Transfer

This procedure ensures that the images conserve their edges without the textures. To remove all the texture information of an image, we used style transfer like in the original paper.

The image for which you want to change the texture, but keep the edges is called the **content image**. The image were the the texture is extracted is called the **texture image**. The image created by the style transfer is called the **stylized image**.

We applied style transfer on Imagenette to replace the original texture of the object (dog's fur for example) with a random texture (a parachute's fabric and color) from sampled from the Describable Texture Dataset [Cimpoi *et al.*, 2014].

Our first aproach was to use a random textures to stylize Imagenette. However, the edges of the images were sometimes destroyed, and were not identifiable by humans.

To avoid poor quality images, we handpicked 11 textures that generated good stylized images. Each time an image was to be stylized, we picked a texture image randomly from our pool. This ensured that the texture of a stylized image had no correlation with its original class.

We used the implementation of AdaIN-style [Huang and Belongie, 2017] made by [BethgeLab, 2019] for the style transfer. The architecture of AdaIN-style can be seen on Fig. 1.

AdaIN-style is the first real-time style transfer algorithm than can transfer arbitrary new styles. That means that we can stylize an image with any texture image we want. This pretrained architecture uses a VGG-19 network to encode the content and style images. An AdaIN layer is then used to perform style transfer in the feature space. A Decoder then decodes the AdaIN output to image space.

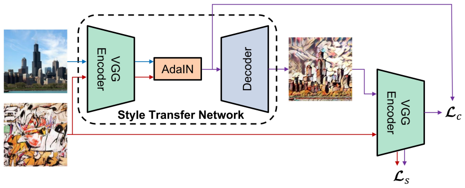


Figure 1: AdaIn style transfer network.

2.3 Metrics

We decided to only use the multi-class Accuracy, as our class are balanced, and not too numerous.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is the True Positives, TN is True Negatives, FP is the sum of the False Positives and FN is the sum of False Negatives.

Top-k accuracy is a useful metric when there is a lot of classes to predict. Our dataset only contains 10 classes, so we deemed it wasn't necessary to introduce this metric.

2.4 ResNet

The authors of the original article used ResNet-50 and other variants of the model in their experiments. Due to the great computational cost of training such behemoths, we've settled with a more modest ResNet-18 to fit our time constraints and smaller dataset.

ResNet is based on the idea that stacking layers in a CNN should not degrade the network performance, because any unnecessary layer could in theory be simplified to an identity function that wouldn't change anything.

However, this is not the case in regular CNNs because of the vanishing gradients, where the gradient becomes smaller and smaller as multiplications are stacked during backpropagation. Moreover, a network with a large number of layers is also more prone to overfit, if trained with sufficient data because it has too much parameters.

Furthermore, learning the identity function is a hard task for a neural network. Most of the time, neurons will, approximate the identity function at best, which would lead to an increasing error as layers are added to the network.

The authors of ResNet, [Kaiming *et al.*, 2015], proposed a ground-breaking solution to the problem, by introducing 'skip-connections' that would map a layer to another beyond its next-in-line neighbor.

This dual-connection facilitates the learning of the identity function as the next layer could be zero-ed out by the new function, leaving only the current layer as output to the skip-connection, as illustrated in Fig. 2. This is due to the newly formed $F(x) + x$ function where $F(x)$ refers to the connection to be skipped.

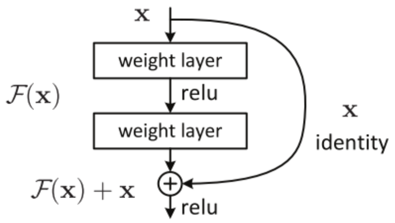


Figure 2: Residual block: identity shortcut connection.

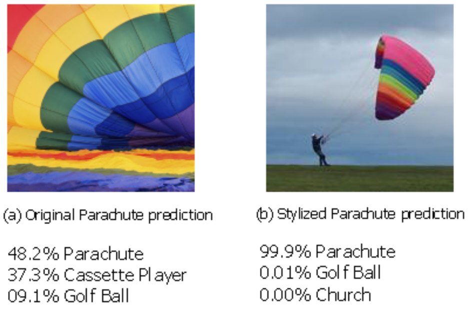
3 Experiments

For all of our experiments, SIN and IN represent Stylized-Imagenette-160 and Imagenette-160 datasets respectively. For example, the nomenclature IN →IN signifies that the model was trained on IN and tested on IN. The two datasets have the same image identifiers for their testing set, only the stylization introduced a change.

3.1 Texture bias hypothesis

The first experiment conducted by the authors was to validate the texture bias through image classification on handpicked stylized subsets. Their hypothesis was that the re-texturized image would be classified as the texture representation, rather than the underlying content using the shapes, which would be contrary to popular belief.

We've reproduced their experiment using the picture of an English Springer as our content image (a), and a stylized version of the Springer (b) using AdaIN style transfer (Huang & Belongie, 2017) described in section 2.2 to introduce a texture-shape cue conflicts. The texture used to generate the stylized image is a colorful parachute, which can be seen in Fig. 3. The model trained on imagenette-160 succesfully identified both the texture and the test image.



(a) Original Parachute prediction

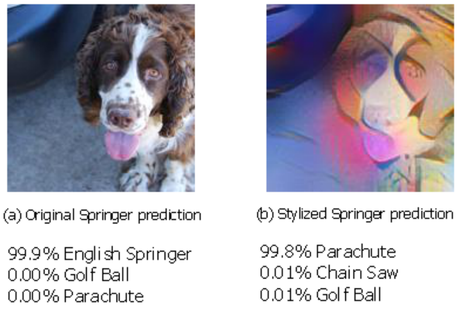
48.2% Parachute
37.3% Cassette Player
09.1% Golf Ball

(b) Stylized Parachute prediction

99.9% Parachute
0.01% Golf Ball
0.00% Church

Figure 3: Classification results of the colorful parachute texture and test image after training on imagenette-160.

The classification results after training the resnet-18 on the imagenette-160 dataset can be seen on Fig. 4. We've obtained probabities in line with what was suggested by the authors, where the stylized image was recognized as a parachute, due to the texture bias, rather than an English Springer.



(a) Original Springer prediction

99.9% English Springer
0.00% Golf Ball
0.00% Parachute

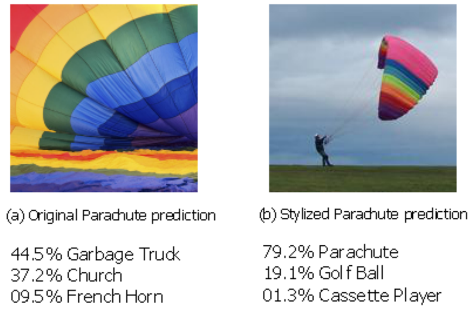
(b) Stylized Springer prediction

99.8% Parachute
0.01% Chain Saw
0.01% Golf Ball

Figure 4: Classification results for the English Springer after training the model on imagenette-160.

Coincidentally, training the model on the stylized dataset, which was believed to reduce the texture bias in favor of a shape bias, no longer produce correct predictions for the texture image of the parachute, as can be seen in Fig. 5. The texture image is wrongfully predicted as a garbage truck, due to the lack of shape clue in the image.

The content image on the right however is correctly predicted, thanks to its form. It is also to be noted that the confidence in this prediction is lower than for the imagenette-160 trained model, as priving the CNN from its texture bias reduce the performances of the model.



(a) Original Parachute prediction

44.5% Garbage Truck
37.2% Church
09.5% French Horn

(b) Stylized Parachute prediction

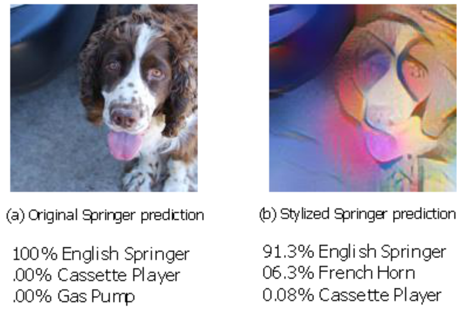
79.2% Parachute
19.1% Golf Ball
01.3% Cassette Player

Figure 5: Classification results of the colorful parachute texture and test image after training on stylized-imagenette-160.

Retraining the model on the stylized imagenette-160 dataset with the same parameters (resnet-18, 20 epochs, 2e-2 learning rate), led to classification probabities available in Fig. 6.

As expected, and although the results are a bit less accurate than on the original model, we get the correct class prediction for the stylized image, suggesting that our CNN recognized the object from its shape this time around, having no texture clue to learn from.

The confidence level for the prediction of the original image however is higher for this model, suggesting that it was easier to infer the class of this particuliar image from its shape rather than its texture. This shows that, although stylized-imagenette-160 does not perform on average better than imagenette-160, there are still instances where it can outperforms the later, when enough shape clue is present.



(a) Original Springer prediction

100% English Springer
.00% Cassette Player
.00% Gas Pump

(b) Stylized Springer prediction

91.3% English Springer
06.3% French Horn
0.08% Cassette Player

Figure 6: Classification results for the English Springer after training the model on stylized imagenette-160.

The results of this experiment are also available under the dog-parachute-experiment notebook, in the github repository.

3.2 Training Procedure

Global Parameters

Number of epochs: For our experiments, we did a fixed number of epochs (15) because it did not overfit, nor underfit the data too much (Fig. 7).

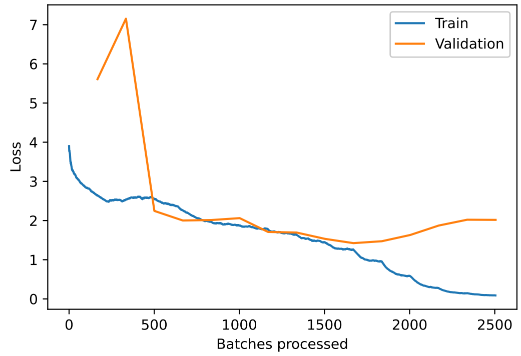


Figure 7: Loss of training on IN →IN

Learning rate: We used FastAI's "lr finder" (Fig. 8) to tune our learning rate. We fixed the maximum learning rate at 1e-2 by looking at the learning rate space search graph.

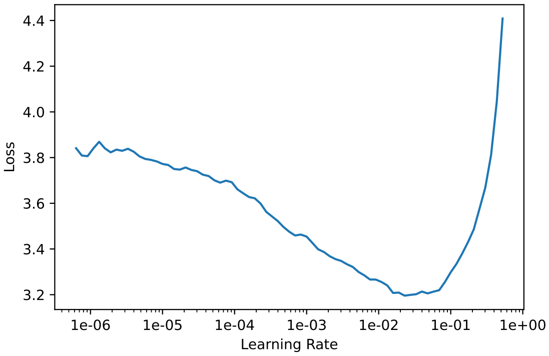


Figure 8: Learning rate finder IN →IN

Validation set: We do a random 80/20 split of the train set to generate the validation set.

Experiment-specific parameters

We used an Early Stopping callback for the FineTuning experiment only, because this model was more prone to overfitting. As we can see in Fig. 9 , the maximum learning rate has to be much lower than previous experiments We defined the max learning rate at 1e-4.

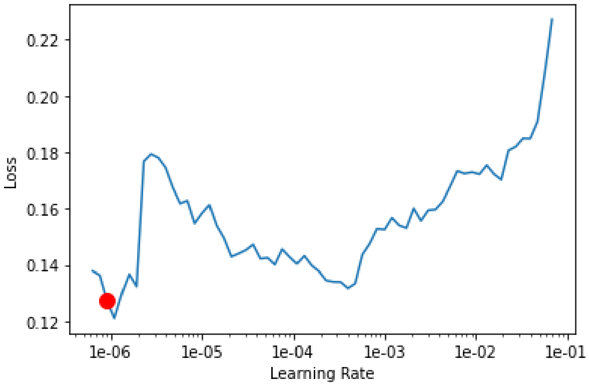


Figure 9: Learning rate finder of SIN + IN + finetune(IN)→IN

3.3 Robustness of shape-based representations

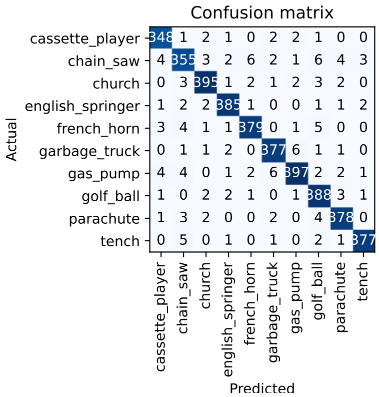
We reproduced the 4 experiments of the original paper to prove the robustness of a shape-biased CNN (the CNN trained on SIN), and show the fragility of the texture biased CNN (the CNN trained on IN).

Architecture	IN→IN	SIN→SIN	IN→SIN	SIN→IN
ResNet-18	**0.963**	0.908	0.313	0.624

Table 1: Accuracies of ResNet18 model on different tests

IN →IN

A model trained on IN and tested on IN is the usual training scheme used in the literature. It yields competitive results, as we already know (Fig. 10).



	cassette_player	chain_saw	church	english_springer	french_horn	garbage_truck	gas_pump	golf_ball	parachute	tench
cassette_player	348	1	2	1	0	2	2	1	0	0
chain_saw	4	355	3	2	6	2	1	6	4	3
church	0	3	395	1	2	1	2	3	2	0
english_springer	1	2	2	385	1	0	0	1	1	2
french_horn	3	4	1	1	379	0	1	5	0	0
garbage_truck	0	1	1	2	0	377	6	1	1	0
gas_pump	4	4	0	1	2	6	397	2	2	1
golf_ball	1	0	2	2	1	0	1	388	3	1
parachute	1	3	2	0	0	2	0	4	378	0
tench	0	5	0	1	0	1	0	2	1	377

Figure 10: Confusion Matrix of IN →SIN

IN →SIN

The original paper trained a model on IN and tested it on SIN which yielded poor results (Fig. 11) as the model is texture biased, and the SIN dataset is specifically designed to promote shape biased models.

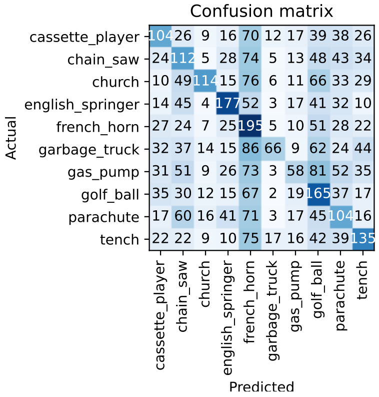


Figure 11: Confusion Matrix of IN →SIN

SIN →SIN

This training scheme yields fairly good results (Fig. 12). This shows that the model is capable to learn meaningful features on the stylized dataset.

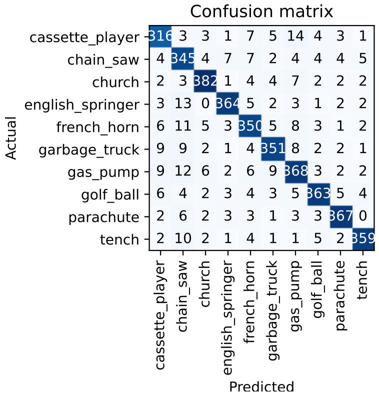


Figure 12: Confusion Matrix of SIN →SIN

SIN →IN

This training scheme yields superior results (Fig. 13) than IN →SIN (Fig. 11), however there are much better than IN →SIN. This demonstrates that a shape-biased model generalizes better than texture biased models.

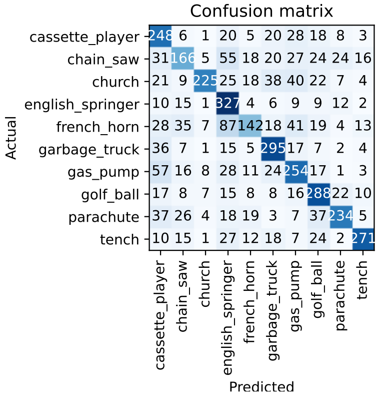


Figure 13: Confusion Matrix of SIN →IN

3.4 Shape-ResNet

Now that we demonstrated the excellent generalization capabilites of shape biased models, our goal is to surpass the performance of the IN →IN model. **All the following models are evaluated on the IN test set.**

Architecture	IN	SIN+SIN	SIN + IN + finetune(IN)
ResNet-18	0.963	0.973	**0.974**

Table 2: Accuracies of ResNet18 model on IN test set

SIN + IN →IN

Mixing the 2 datasets has the effect of a data augmentation. The model will be better at generalizing (Fig. 14). The performance improvement as a result of data augmentation is well known. However, this data augmentation is dedicated to increase shape bias, and does not make the model overfit.

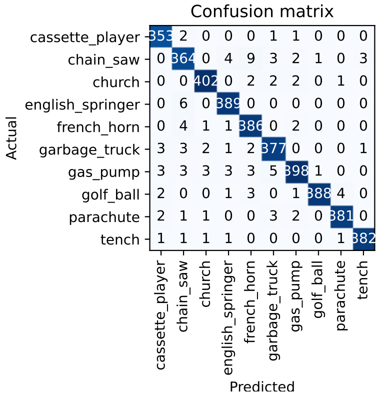


Figure 14: Confusion Matrix of SIN + IN →IN

In the Fig. 15, we observe that the validation loss is much higher than the training loss. The model overfits a lot on the training data.

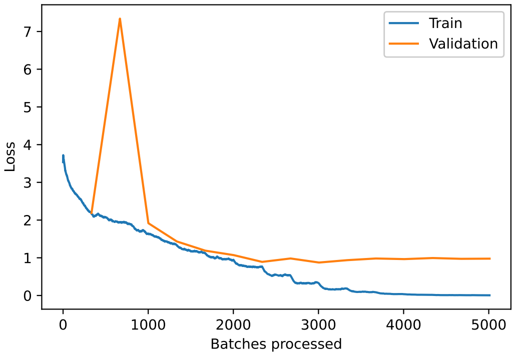


Figure 15: n
Loss of training on the SIN + IN →IN experiment

SIN + IN + finetune(IN) →IN

This method yields the best results (Fig. 16) The finetuning only occurs on the last layers of the ResNet (the classification layers) and not the feature extracting layers.

Hence, the model does not overfit on textures, but is better to better classify images.

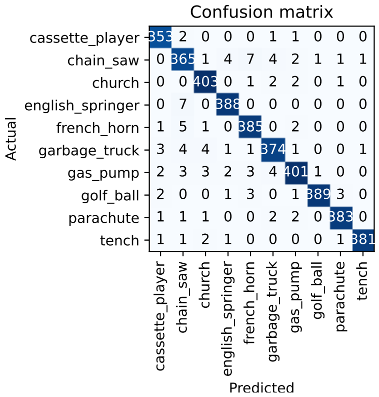


Figure 16: Confusion Matrix of SIN + IN + finetune(IN) →IN

We can see on the Fig. 17 that the train loss is fluctuating a lot even with a small learning rate. However, the validation loss is stable and slowly decreasing. The validation loss is slightly lower than the training loss, which is a sign of underfitting. This is what we want: the feature extraction is more shape biased than texture biased, and the model is not overfitting on the texture like before. We also observe that both losses are both really low compared to the previous experiments. We can conclude that finetuning had a positive effect on the generalization power of the model.

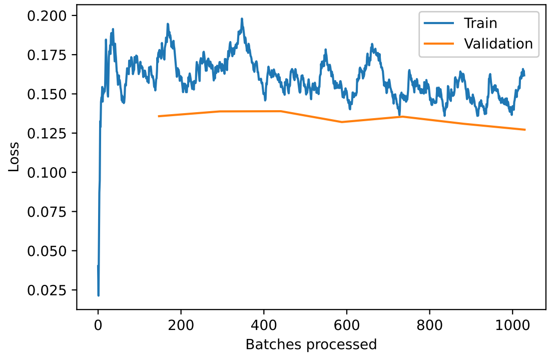


Figure 17: Loss of training on the finetuning experiment

3.5 Noise resistance

One of the major finding in the original paper was that the model trained with a shape bias had developed a better resistance to distortions than the regular model. A model trained to recognize textures should behave differently from one trained to recognize shapes on a noised data set. The key idea is that not all types of distortions are the same and this model should in theory only be resistant to some of them. We decided to focus on 3 main types of distortions. A uniform noise, a high pass filter and a low pass filter.

The result of the original paper show us that the SIN model has a better resistance to uniform noise, high pass filtered image and a worse resistance to low pass filtered images.

This makes sense since a uniform noise has a much worse effect on texture than overall shape. Also, a high pass filter acts as a contour amplifier thus making the shape in the image more prominent. In the same line of thought, a low pass filter has the opposite effect, blurring the edges of an image and making it's shape much less discernible.

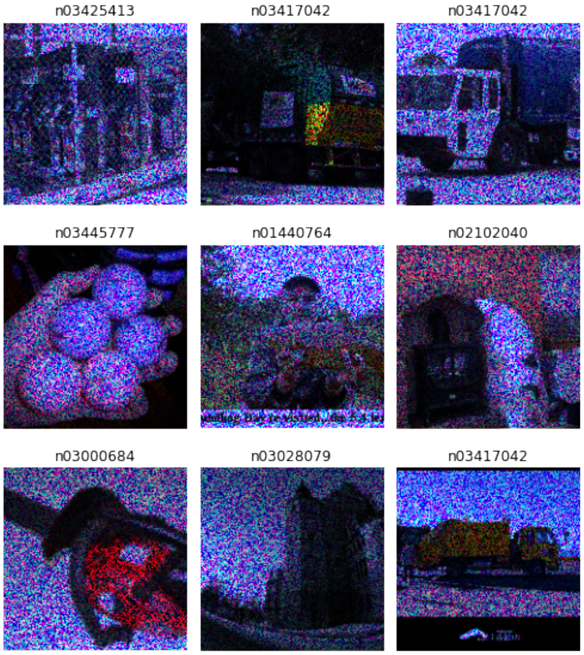


Figure 18: Uniform noise

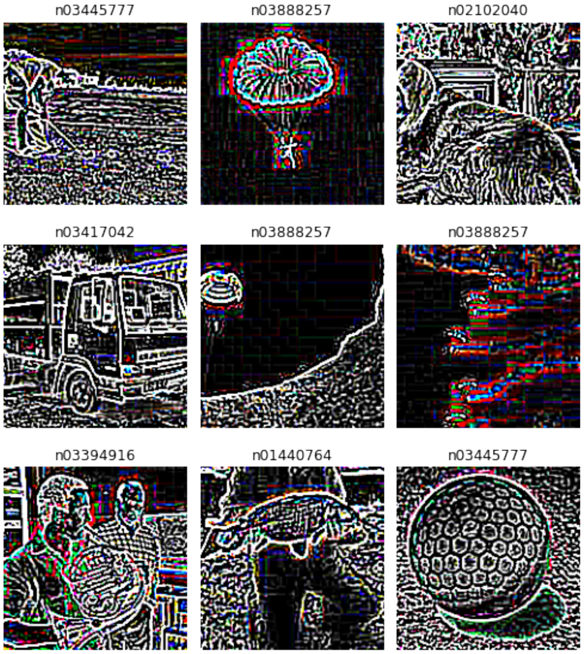


Figure 19: High pass

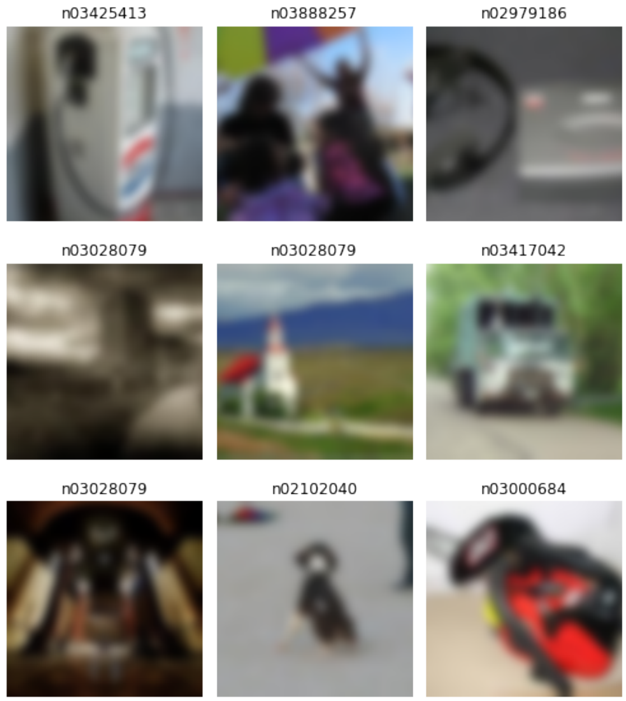


Figure 20: Low pass

Experiment

We decided to reproduce this experiment with our own data. All of the distorted data sets are created form the regular Imagenette data set. We applied the distortions at different scale gradually increasing the effect on the images.

We then compared the results of our three trained models against those different noise levels. The models were the regular resnet trained on imagenette (IN), a resnet trained on a stylised version of imagenette (SIN) and one trained on a combination of this data (SIN+IN). Each model was given the same test set on which a gradually increasing noise was applied and an accuracy was computed.

The different levels of noise were chosen arbitrarily to create different enough distortions. The original values and algorithms used in the paper were not available.

Results

In all of the test we conducted, the SIN model always start off worse. This is to be expected because it was trained on stylized images and the test set for the distorted images is based on regular images. The IN and SIN+IN model start at about the same accuracy.

The figures clearly show that the shape biased models have a better resistance to noise the regular model.

We were surprised by the results for two reasons. First, the SIN model started off at a worse accuracy, so we were expecting to compare the resistance to distortions as a relative drop from the original accuracy but it managed to have a better absolute accuracy than the regular model.

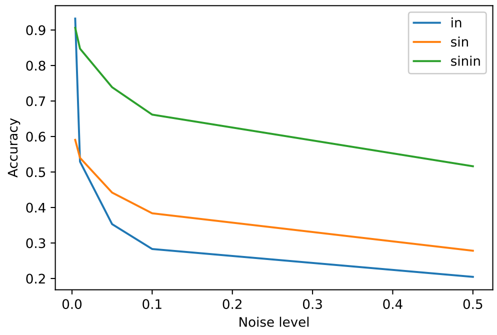


Figure 21: Uniform noise

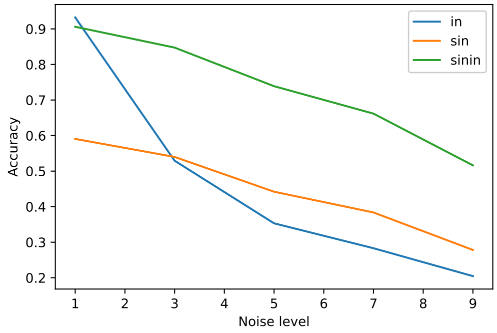


Figure 22: High pass

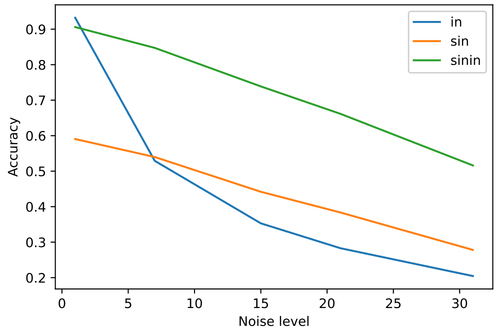


Figure 23: Low pass

Secondly, we were expecting the SIN model to under perform in the low pass area (Fig. 23), but it managed to be better than the IN model. This goes against the results of the original paper and against common sense. It is possible the levels of blur applied were too strong for the IN model to detect any texture.

3.6 Github

This report, as well as all the source code used to produce those results are available under our git repository: JeyzerMC/CNN-Texture-Biais.

4 Approach analysis

The original scope of our project was to try to reproduce the findings in the reference paper. We approached the subject with goal of of validating a simple thesis. Models can be induced with a shape bias during their training by changing the input data and this shape bias can have numerous advantages.

We would like to think that we achieved this goal, but there were some shortcomings in our methods that may have impacted our results and we would like to clarify their causes and possible effects.

4.1 Experiments

The first liberty we took was to only replicate a subset of the experiments of the paper. We can see the excellent quality of the work, because their experiments were easily reproducible, and most of the steps were clearly explained. Some of the experiment in the original paper were purely exploratory and we felt did not contribute our main thesis. As an example, we did not choose to test if human vision had an inherent shape bias as it has been a widely idea and outside our scope.

Also, a lot of the experiments comparing the models against transformed test sets used transformations that we felt were a bit redundant. Some of these transformations like a silhouette detector and a gray scale version of the image were not as important as the different disturbances we introduced. We don't feel like omitting to reproduce these experiments weakened our thesis.

4.2 Dataset

Most of the experiments depend on comparing two identical models trained on two versions of the same data set. One version left intact and another that saw stylised using the AdaIn model.

This is were we took most of our liberties because of our time and resources constraints. The data set we chose to use for was Imagenette, a subset of the Imagenet data set. This data set is limited by many factors. First, the number of images is considerably inferior to imagenet. Secondly, the images have lower resolution. 160 by 160 pixels in the Imagenette variant we used. Thirdly, the data set only contains 10 classes as opposed to Imagenet who has 200.

We experimented with other data sets to solve these issues, but ended up rejecting them for various reasons. The cifar10 data set's images were too low resolutions and we found the AdaIn model could not stylise them. The original Imagenet dataset was too voluminous and the training time was too long to enable us to experiment. Ultimately, we decided to remain with Imagenette and its limitations.

The number of images reduced out training time, but also reduced the size of the test set, thus giving us less confidence in our metrics. The lower resolution also helped our experimentation velocity by decreasing the training time, but it may have had an effect on the resistance to noise experiment. It is entirely possible that noise applied to low resolution images would have a different effect.

The fact that we only have 10 classes was helpful when we were visualising our confusion matrices but rendered one of our metric, top k accuracy, completely, unusable. The top k accuracy was used in the original paper as the top 5 accuracy on a dataset with 200 classes. It is meaningless to have a top 5 accuracy in a 10 classes dataset and reducing there is no k value for which this metric would start to be meaningful.

4.3 Models

The last divergence we took from the paper was our choice of models. The authors compared the effect of shape bias on 4 model architectures. Granted all of these models were slight variations of the ResNet-50, the variation of models add wight to their experiments.

In our case, we focused out training on ResNet-18. This model has a lot less parameters and is faster to train. Taking into consideration the fact that our images are of lower resolution we decided this architecture was powerful enough.

4.4 Results

Overall, we were able to reproduce their results and even surpass them. This can be justified by the fact that the classes in Imagenette were handpicked to be easily classified, and that there were less classes than Imagenet. Our results fluacted a bit between different training. For example, fine tuning does not seem to be necessary in our study. However, when the performance of SIN + IN is sometimes lower than 0.9. In this scenario, fine tuning always surpassed IN →IN.

To correct this, we could have calculated our accuracy on the average of five runs, like the authors of Imagenette [FastAi, 2019] suggested.

4.5 Ideas for future studies

Testing this robustness in related areas where shape bias is more important than image classification, like for example in image segmentation, could prove to be quite interesting.

5 Conclusion

We were able to replicate most of the experiences of the original paper and reach the same results. Our experiments showed that training a network with a combination of regular and stylized images increased its classification performance and robustness to disturbances. This show a clear advantage for models that have a strong shape bias.

This is encouraging as it show that we are getting closer to understanding what is learned by out models. It's also interesting to know that we can introduce biases into our model by transforming the data we feed into it.

References

[BethgeLab, 2019] BethgeLab. Stylize datasets. https://github.com/bethgelab/stylize-datasets, 2019.

[Brendel and Bethge, 2019] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

[Cimpoi *et al.*, 2014] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[FastAi, 2019] FastAi. Imagenette. https://github.com/fastai/imagenette, 2019.

[Fukushima, 1980] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. pages 193–202, 1980.

[Gatys *et al.*, 2017] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture and art with deep neural networks. pages 178–186, 2017.

[Geirhos *et al.*, 2018] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness, 2018.

[Huang and Belongie, 2017] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.

[Hubel and Wiesel, 1959] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. page 574–591, 1959.

[Kaiming *et al.*, 2015] He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[Kriegeskorte, 2015] Nikolaus Kriegeskorte. Deep neural networks: a new framework for modelling biological vision and brain information processing. *bioRxiv*, page 429, 2015.

[Kubilius *et al.*, 2016] Jonas Kubilius, Stefania Bracci, and Hans de Beeck. Deep neural networks as a computational model for human shape sensitivity. 2016.

[Lecun *et al.*, 1998] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. pages 2278–2324, 1998.

[Marr, 1982] David Marr. Vision: A computational investigation into the human representation and processing of visual information. *Henry Holt and Co., Inc.*, 1982.

[Ritter *et al.*, 2017] Samuel Ritter, David G. T. Barrett, Adam Santoro, and Matt M. Botvinick. Cognitive psychology for deep neural networks: A shape bias case study, 2017.