Deep image priors for transformers and their variants

Sriram Balasubramanian

1 Introduction

Deep neural networks (DNNs) have been extremely successful at a wide variety of tasks like computer vision, natural language processing, time series forecasting, etc. However, the choice of neural network architecture for these different domains is different, and is motivated by the nature of the function we are trying to approximate. Many architectures exploit the fact that their domain is structured in some ways and has certain invariances or symmetries within them. For example CNNs exploit the fact that image datasets generally have translational invariance, that is, the mapping from the images to labels does not change when the image is shifted. RNNs, which are used to process sequential data have a locality bias, that is, an assumption that the relevant information needed to predict the target is found locally. They also assume that all information need to predict the target at time t can be found in the input before time t. Ulyanov et al. [2017] attempted to quantify this kind of inductive bias by using a randomly initialized neural network to perform certain image reconstruction tasks. I will try to apply their methodology to transformer based architectures and discuss their inductive biases.

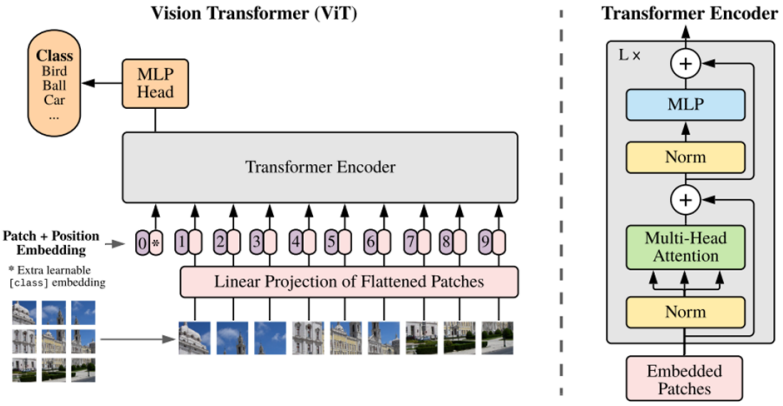
2 Related Work

Ulyanov et al. [2017] showed that it was possible to perform many image reconstruction tasks on corrupted images like image denoising, image inpainting, image restoration and also other kinds of tasks like image super resolution by exploiting the inductive biases present in architectures like deep CNNs like ResNets [He et al., 2015] and U-Nets [Ronneberger et al., 2015]. They train a randomly initialized generative neural network to map a random noise vector to the corrupted image. Due to the inductive biases of the network and regularization properties of SGD, the network outputs a "clean" version of the corrupted image before overfitting on the corruptions. They propose that randomly initialized neural networks can be an image prior, similar to hand crafted priors like total variation norm. We can use this method to quantify the "image priors" in these deep neural nets and compare them against one another.

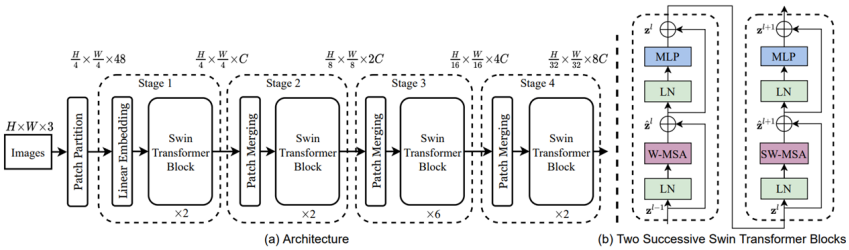
Several attempts have been made to theoretically analyze the deep image prior phenomenon. Cheng et al. [2019] analyze the He initialized U-Net CNNs as a low-pass filter, using a Gaussian process interpretation. Heckel and Soltanolkotabi [2019] also show that CNN decoders produce low-pass filters when trained using gradient descent. Tachella et al. [2020] introduce a formal link between such networks through their neural tangent kernel (NTK) and well-known non-local filtering techniques.

The Transformer architecture introduced by Dosovitskiy et al. [2020] was very similar to the transformers used in NLP and is described in Figure 1a. The image is divided into smaller patches, and each patch is mapped to an embedding by a shared transform. These embeddings are then fed into the transformer, which consists of a multi-head attention block and an MLP on top of it. We stack several of these transformer blocks to obtain the encoder. To decode it back into an image, we use an MLP. Since the architecture is identical to that of traditional NLP transformers, we don't expect any strong image specific inductive biases.

Building on this architecture, Swin Transformer was introduced by Liu et al. [2021], which is a variant of transformer which has hierarchical patching at each layer. A diagram of Swin Transformer is presented in Figure 1b . It begins by dividing the image into very small patches of size 4 and then passing it through a transformer block. The 4 of these patches are then merged and fed to another



(a) Vision Transformer



(b) Swin Transformer

transformer block, and this repeats a few times. This is supposed to model the hierarchical nature of images, using large scale and small scale characteristics of the image. The initial transformer layers would extract the low level features of the image, and the next layers extract higher level features.

We proceed to investigate the image prior properties of the two transformer-based architecture in this paper.

3 Problem Statement

We consider an image x and corrupted image $\tilde{x} = x + c$, where c is a random corruption vector. We train a neural net $f(z, \theta)$ where θ are the model parameters and z is a random code vector which can be considered fixed for our purposes. We also have an optimizer O which is a variant of gradient descent which iteratively updates θ such that the parameters of the neural net at step t is θ_t.

We can then analyze the difference between mean squared error of the net output w.r.t the clean and noisy image by using the below equation:

$$E_c[\|\tilde{x} - f(z,\theta)\|^2] = E_c[\|c\|^2] + E_c[\|x - f(z,\theta)\|^2] + 2E_c[c^T(x - f(z,\theta))]$$

or $\Delta MSE = E_c[\|c\|^2] + 2E_c[c^T(x - f(z,\theta))]$. Also note that θ evolves according to gradient descent as follows:

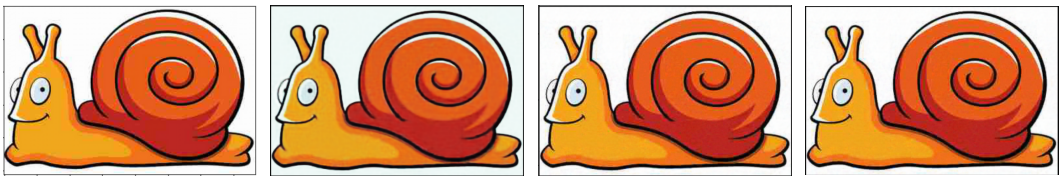
$$\theta_t = \theta_{t-1} - \eta(x + c - f(z,\theta_{t-1}))^T \frac{\delta f}{\delta \theta}|_{\theta=\theta_{t-1}}$$

Therefore, θ_t, and thereby $f(z,\theta_t)$ depends on c. This tells us that at step t, if $x - f(z,\theta_t)$ is uncorrelated with c, the term $E_c[c^T(x - f(z,\theta))]$ would be 0, implying that the difference between the MSEs would be high. On the other hand, if $f(z,\theta_t) \approx x + c$, then the term would be negative and thus the difference between the two MSEs would be low or even negative.

We can immediately observe that this depends crucially on the nature of the function f and the specific optimizer O which governs the evolution of θ. Therefore, the strength of the image prior is attributable to the tuple (f, O, θ_0). There is a debate on which of these is most significant to the

(a) Clean Image (original) (b) Corrupted with noise (c) Denoised by SkipNet (d) Denoised by Transformer (e) Denoised by Swin Transformer

Figure 2: Denoising of an image of a F16 airplane



(a) JPEG image with artifacts (b) DeJPEGed by SkipNet (deep CNN) (c) DeJPEGed by Transformer (d) DeJPEGed by Swin Transformer

Figure 3: DeJPEGing of an image of a snail

image prior properties of the neural network. Heckel and Soltanolkotabi [2019], Cheng et al. [2019] argue that the most significant factors are the initialization and the convolutional layers of the neural network. In contrast, Tachella et al. [2020] argue that the crucial factor is the optimizer, and perform experiments which show that gradient descent does not work as well as Adam. In this paper, we primarily study the effect of architecture on image priors.

We primarily use difference between PSNR or MSE between model output and clean image to quantify the quality of reconstruction. Let us assume that there is a maximum threshold on the MSE between the output of the neural network that is tolerable, say m, then we can try to measure the image priors of the neural network - optimizer pair as corresponding to $\max_{t,\|\tilde{x}-f(z,\theta_t)\|^2>m} \Delta MSE_t$. One could also use other metrics like difference between the peak signal-to-noise ratio (PSNR), where $PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$ where MAX_I is the maximum possible pixel value of the image (here fixed at 255).

4 Methods

We follow the methods in Deep Image Prior to measure the extent of inductive biases in various architectures. We consider two tasks: image denoising and image inpainting.

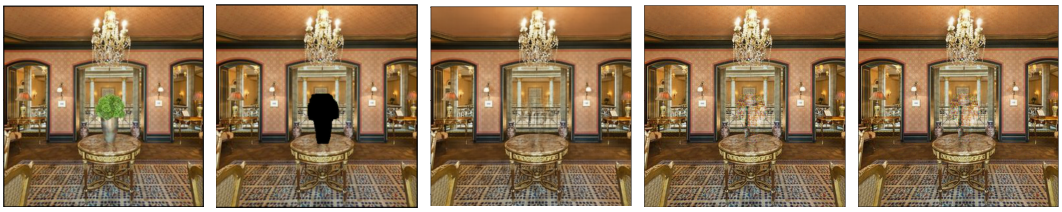
The image denoising task consists of taking an image with Gaussian noise with zero mean and 0.1 standard deviation added to it as input and producing the original clean image as output. The image inpainting task consists of taking an image with certain parts of the image masked out as input and producing the original clean image as output.

The models considered are the Vision Transformer [Dosovitskiy et al., 2020] and the Swin Transformer [Liu et al., 2021] as encoder with a lightweight MLP decoder on top of it . We also consider some other convolutional decoders later.

We measure the quality of image reconstruction using PSNR and mean squared error. We vary hyperparameters like patch size, number of layers, number of (attention) heads and choose the best for each architecture. We also produce plots of the PSNR of the output image vs the corrupted image and PSNR of output image vs the original image as a function of the number of iterations and analyse the differences between them.

(a) Clean Image (original) (b) Image with black text added (c) Inpainted by Skip-Net (d) Inpainted by Transformer (e) Inpainted by Swin Transformer

Figure 4: Inpainting of text over an image of a woman



(a) Clean Image (original) (b) Image with vase masked out (c) Inpainted by Skip-Net (d) Inpainted by Transformer (e) Inpainted by Swin Transformer

Figure 5: Inpainting of vase over an image of a hall

5 Results

We see the results on the denoising task in Figures 2 and 3 and inpainting task in Figure 4 and 5. We can see that Transformers do a poor job of denoising the image as compared to SkipNets. The "denoised" image output by the Transformer is still very noisy as compared to SkipNet's denoised image. Similarly, SkipNet's inpainting is almost flawless, with hardly any artifacts visible while the Transformer inpainting is very poor, with multiple visible artifacts. We can also see faint grid lines on the image which are remnants of the patch boundaries of the transformer. Surprisingly, Swin transformers do not perform any better than transformers at these tasks, even though they have a hierarchichal architecture .

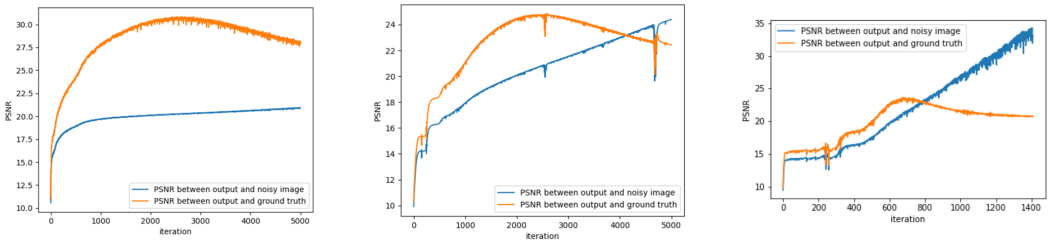
In Figure 9, we can see the PSNR vs iteration plots for different architectures. The blue curve is the PSNR between the model output and the noisy image, while the orange curve is the PSNR between the model output and the uncorrupted image. We can see that the gap between the two plots is very high for SkipNets, as compared to transformers. In fact, the images corresponding to the peak of the PSNR curve for the transformer are low quality and are not very clear and the image only becomes clear at the end, while the images at the peak of the skip-net curve are nicely denoised. Swin Transformers seem to peak much quickly at around 1000 iterations and the gap between the two curves at this peak is even smaller than Transformers.

In Table 1, we see a similar trend across the models. SkipNets perform much better than Transformers or Swin Transformers, which is in accordance to the perceptual quality of the inpainted image in Figure 4 and 5.

In Figures 6 and 7, we see the outputs of the model after 0, 2000 and 4000 iterations of training of SkipNets and Transformer, with each iteration corresponding to a gradient update. In 6, we see the initial and intermediate output of the skip net consist of smooth contiguous regions of color with little noise. This corresponds to the sparse gradient property of images. In 7, we can see that the initial intermediate output has some grid-like repeating patterns, which corresponds to the division of the input into patches by the transformer. The transformer output is initially highly noisy in contrast to the SkipNet output, and the output is never denoised throughout the training.

6 Discussion

6.1 The inductive biases of convolutional layers



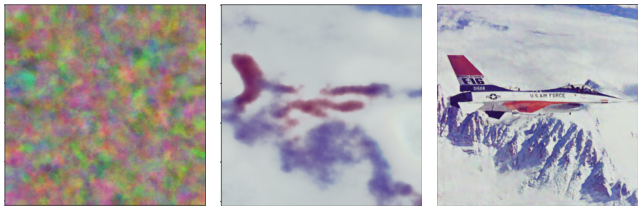
(a) SkipNet (b) Transformer (c) Swin Transformer

Figure 9: PSNR vs iteration (denoising task on F16 image)

Image	SkipNet	Transformer	Swin Transformer	Transformer + 3 conv layers	3 conv layers + Transformer
Woman	0.000244	0.000390	0.000395	0.000289	0.000423
Vase	0.00149	0.001650	0.001643	0.001814	0.001823

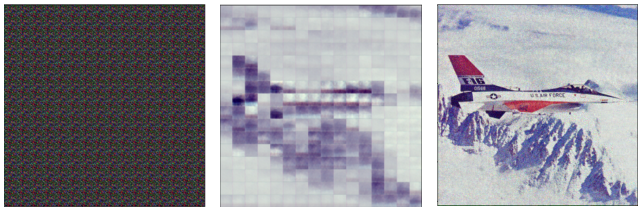
Table 1: MSE for inpainting task on image of a woman and a vase

We now discuss some reasons behind this phenomenon. To a first approximation, an image can be modeled as smooth contiguous regions separated by sharp edges. Since the added noise to the image is random, convolutional layers with kernel size more than 1 and stride 1 would struggle to fit the noise, and thus fit to the smooth parts of the image and learn a smoothing function, which is low pass and will smooth out the noise. ReLU activations, when multiplied by a suitable scale factor can reproduce the sharp edges found in the image. Thus, when combined and stacked together, these neural nets can easily generate natural images when trained. However transformers do not have any layers which are inherently biased towards smoothing like convolutional layers. Thus, they do not have strong image priors. But Swin Transformers are supposed to have stronger inductive biases as compared to vanilla transformers due to their hierarchical structure, so we can ask a natural question as to why we do not observe any significant difference in the metrics and plots that we computed.



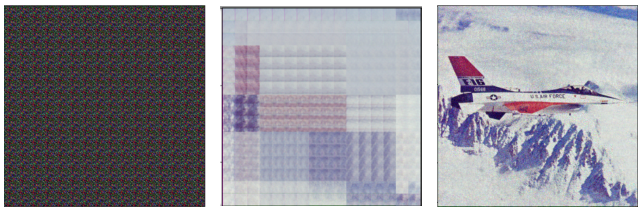
(a) Initial (b) Intermediate (c) Final

Figure 6: Output of SkipNet after 0, 2000, and 4000 iterations



(a) Initial (b) Intermediate (c) Final

Figure 7: Output of Transformer after 0, 2000, and 4000 iterations

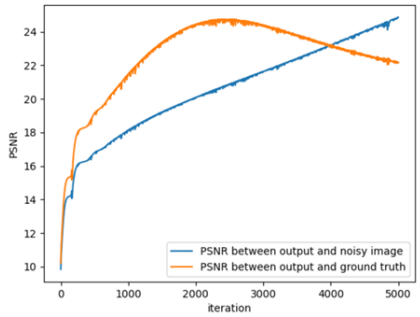


(a) Initial (b) Intermediate (c) Final

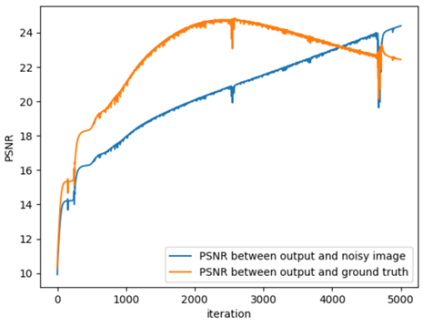
Figure 8: Output of Swin Transformer after 0, 500, and 1000 iterations

6.2 Generative vs predictive inductive biases

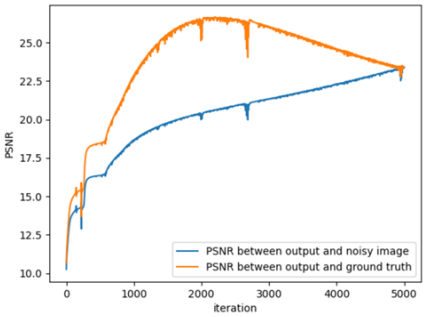
To answer this, we must first distinguish between two kinds of inductive biases present in models, generative biases and predictive biases. Generative bias is the bias towards generating a family of data distributions, while predictive bias is bias towards learning a particular family of input-output mapping. For example, for CNNs, shift invariance is an example of predictive



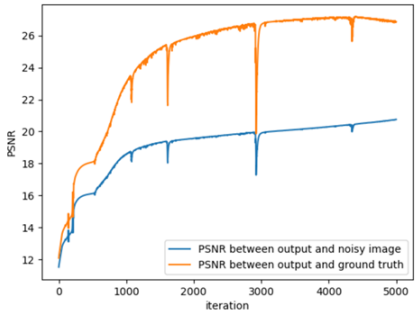
(a) 3 convolutional layers before a transformer



(b) Plain transformer with no convolutions



(c) 1 convolutional layers before a transformer



(d) 1 convolutional layers before a transformer

Figure 10: PSNR vs iteration for different configurations of convolutional layers and transformer (denoising of F16 image)

bias, and not generative bias, since it is related to how the target (that is, the labels) is invariant to changes in the image. However, as we just described, convolutional layers also have generative biases, but these are very distinct from predictive biases. In general, generative inductive bias is not equivalent to or even correlated with predictive inductive bias.

As we saw, the experiments in the deep image prior paper only test for image reconstruction tasks, which are good for measuring generative biases but not predictive biases. The hierarchical nature of the Swin transformer induces favorable predictive inductive biases but not generative inductive biases. This could be one reason why they are used in many predictive tasks like image classification, segmentation ,etc but not generative tasks where CNNs are still state of the art.

In order to test this, we can perform the following experiment. We can equip the transformer with a stack of convolutional layers, before the input of the transformer or after the output of the transformer. When equipped before the transformer, these convolutional layers induce favorable predictive inductive biases because these convolutional layers can be trained to identify input image features (like edges, corners, etc), but hardly any generative biases because they will be suppressed by the biases of the transformer which transformers the output of the convolutional layers. When equipped after the transformer, they will hardly contribute to predictive biases because the image structure in the input signal has been completely changed by the transformer, but they will induce good generative inductive biases. In some sense, this is similar to attaching a ConvNet decoder to the transformer, which inherits the favorable generative inductive biases of the decoder.

According to the theory we outlined, a model with convolutional layers after the transformer output should be much better at image reconstruction as compared to a model with convolutional layers equipped before the transformer input.

6.3 Experiments with transformers with convolutional layers

We show the PSNR plots vs iteration in Figure 10. As we can observe, even adding 3 convolutional layers before the transformer does not really help, since the generative bias due to the convolutions is negligible. However, even adding one convolutional layer after the transformer really helps the gap between the PSNRs . Even after 4000 iterations, the gap between the PSNRs is still positive for the transformer with 1 convolutional layer, with higher max PSNR. Adding 3 conv layers after the transformer only helps more, as the gap is still high and positive even after 5000 iterations.

We can also see the effect of adding convolutional layers to transformers with regards to the inpainting task in Table 1. For the woman photograph, we see the expected trend but for the vase inpainting we actually see the opposite trend. This means that the theory outlined above has some limitations as it is not perfectly able to explain the results.

7 Conclusion

The results show that convolutions in CNNs have much stronger generative inductive biases towards smoothness as compared to attention mechanisms of Transformer or hierarchical patching of Swin Transformer. This doesn't really mean that Transformers are inferior to CNNs, since generative inductive bias may not be linked to predictive inductive bias. Indeed, Transformers have achieved state-of-the-art performance at many predictive tasks which is why they have become popular in computer vision, but they are still not preferred for image generation because of poor generative inductive biases.

References

Zezhou Cheng, Matheus Gadelha, Subhransu Maji, and Daniel Sheldon. A bayesian perspective on the deep image prior. *CoRR*, abs/1904.07457, 2019. URL http://arxiv.org/abs/1904.07457.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Reinhard Heckel and Mahdi Soltanolkotabi. Denoising and regularization via exploiting the structural bias of convolutional generators. *CoRR*, abs/1910.14634, 2019. URL http://arxiv.org/abs/1910.14634.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021. URL https://arxiv.org/abs/2103.14030.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Julián Tachella, Junqi Tang, and Mike Davies. The neural tangent link between cnn denoisers and non-local filters, 2020. URL https://arxiv.org/abs/2006.02379.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. *CoRR*, abs/1711.10925, 2017. URL http://arxiv.org/abs/1711.10925.