Comparing the Image Completion Capabilities of Masked Autoencoder and Neural Fields

Gwyn Gras-Usry
Washington State University
Pullman, Washington
gwyneth.gras-usry@wsu.edu

Dr. Navid Kardan
University of Central Florida
Orlando, Florida
kardan@Knights.ucf.edu

Abstract

In this paper, we explore the reconstruction abilities of three different approaches, a vanilla neural field, a Neural Knitwork, and masked autoencoder (MAE) with an image completion task. For this task, inpainting, we mask a section of pixels and have our model reconstruct the non masked image. The vanilla neural field and Neural Knitwork both use neural fields, which take coordinates and output a representation of some signal. For our approaches, the neural field takes pixel coordinates and outputs a pixel color representation. Our vanilla neural field is a single neural field and the Neural Knitwork consists of two neural fields, a Patch MLP and MLP Reconstructor. Our third model, the MAE, is a denoising autoencoder that reconstructs the original signal given a masked input.We applied all three approaches to our image completion task, inpainting, where we used PSNR to compare the quality of the reconstructions of CIFAR10 test images. The MAE had the highest average PSNR, with the vanilla neural field and Neural Knitwork being around the same and lower. The two neural field based models could reconstruct some images with much higher or lower PSNRs than average due to the images having different optimal parameters. The MAE reconstructed all images with similar accuracy. We found the MAE was very robust against transformations, such as rotation. A MAE pre-trained only on CIFAR10 was also able to reconstruct different datasets, CIFAR100 and SVHN, with very similar accuracy.

1. Introduction

We use three different models, a vanilla neural field [2], Neural Knitwork [3], and MAE [1], to perform an inpainting task. This task involves masking a section of the input and then passing the masked image into a model for reconstruction. We used CIFAR10 images to compare the PSNR for these reconstructions. Our first approach, the vanilla neural field [2] uses a simple neural field with a Fourier feature mapped coordinate input to represent images. Our second approach, the Neural Knitwork [3], expanded upon this vanilla neural field architecture. The Knitwork uses two neural fields with the same design as the vanilla neural field, the first being the Patch MLP. The Patch MLP takes Fourier feature mapped pixel coordinates and outputs a patch representation. The second neural field, the MLP Reconstructor, takes the patch representation and reconstructs the original pixels from it. Both approaches train neural fields individually for each image. Our third approach, the MAE, on the other hand has to pre-train before it can be used for reconstruction. The MAE is a denoising autoencoder that is pre-trained on masked images that it reconstructs using an asymmetric encoder-decoder design. We also test the MAE's reconstruction abilities with out-of-distribution datasets and transformed images.

We found the MAE was overall the best at reconstructing multiple batches of test images with similar accuracy and had the highest average PSNR. However, the vanilla neural field and Neural Knitwork were able to get much higher PSNRs on some images. Due to the individual nature of the optimal parameters for each image, these two models could not get high PSNRs on every image as they used the same parameters for each image.

Another finding was that the MAE was naturally very robust against transformations and able to reconstruct datasets it was not pre-trained on very well. The MAE has a much greater ability to generalize than the neural fields, which must have prior knowledge of these transformations to perform them.

2. Related Work

2.1. Positional Encoding

Tancik, et al. [2] use Fourier feature mapping, also called positional encoding, to vastly improve a multilayer perceptron's (MLP)/neural field's ability to represent high frequency functions in images. This same Fourier feature

mapping is used in the Neural Knitwork. The MAE positionally embeds inputs for both the encoder and decoder, similar to positional encoding.

2.2. Neural Fields

Similar to a field in physics, neural fields in visual computing take coordinates and generate a representation in the form of a field, where each coordinate has corresponding reconstructed value [4]. Neural fields can also be referred to as coordinate-based MLPs. Neural Knitworks [3] uses neural fields for synthesis tasks, i.e. inpainting, super-resolution, and denoising. In this case, the neural fields take pixel coordinates and represent images by reconstructing the original signal, an RGB pixel value. The Knitwork architecture consists of a patch MLP, which takes positionally encoded [2] coordinates and outputs an overlapping 3x3 multiscale patch representation. Patch reconstruction error and cross-patch consistency are calculated for this representation. A discriminator is used to predict which patches are from the original distribution. Then the patch representation is passed to the MLP reconstructor and the original pixel color is reconstructed for each coordinate. We used our own recreation of this Neural Knitwork to perform inpainting on a single image. Our implementation only uses one patch scale, original size, and does not include the discriminator. [2] is another neural field we use on this inpainting task.

2.3. Autoencoders

An autoencoder consists of an encoder that maps the input signal to a latent representation and a decoder, which reconstructs the original input from this representation. He, et al. [1] use a masked autoencoder (MAE) approach with an asymmetric design. The encoder is larger than the decoder, as it only has to work with masked images during pre-training. The decoder is lightweight and processes the full set of tokens, which includes mask tokens and the latent representation. This asymmetric design saves time when training because the larger encoder only sees around 25% of the input.

3. Methods

3.1. Image Completion Task: Inpainting

When inpainting, we mask a section of the input image and pass it to our model for reconstruction. We used the vanilla neural field from Fourier feature mapping [2], Neural Knitwork [3], and MAE [1] to reconstruct the same batches of CIFAR10 test images with the same inpainted region and compare their PSNR. We mask a rectangular section of pixels, starting at (10, 10) and ending at (14, 20).

$$PSNR = 10 \cdot log_{10}(\frac{MAX_I^2}{MSE})$$

MAX_1 is the maximum pixel value, which is typically 255 or 1. Here we use 1 for all models. MAX_1 is divided by mean squared error for masked patches.

3.2. Fourier Feature Mapping

Fourier feature mapping enables our vanilla neural field [2] to better represent higher frequency content (fine image detail) in images. In figure 5, you can see the visible improvement of the high frequency representation for one 512x512 image. The image with no mapping is blurry compared to the mapped images. Basic mapping is still not as detailed as Gaussian Fourier feature mapping, which produced a very accurate image representation. We use Gaussian Fourier feature mapping for our models. The following three mappings are compared in figure 5.

$$No\ mapping:\ \gamma(v) = v$$

$$Basic\ mapping:\ \gamma(v) = [cos(2\pi v), sin(2\pi v)]^T$$

$$Gaussian\ Fourier\ feature\ mapping:$$

$$\gamma(v) = [cos(2\pi Bv), sin(2\pi Bv)]^T\ where\ each$$

$$entry\ in\ B \in R^{mxd}\ is\ sampled\ from\ \mathcal{N}(0, \sigma^2)$$

3.3. Vanilla Neural Field

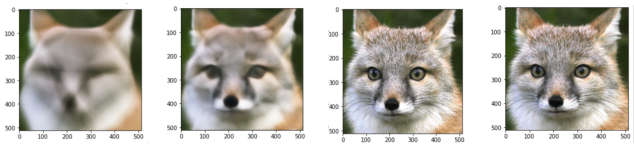
We use the neural field found in [2] as our vanilla neural field model. This neural field takes Fourier feature mapped pixel coordinates and outputs a representation of RGB pixels.

3.4. Neural Knitwork

Our Neural Knitwork implementation comes from [3]. The Neural Knitwork consists of a Patch MLP and MLP Reconstructor. The Patch MLP takes Fourier feature mapped coordinates, using the same mapping as our vanilla neural field, and returns a 3x3 patch representation. We then calculate patch reconstruction loss for the patch representation. This representation is passed to the MLP Reconstructor, which reconstructs the original image pixels. We calculate reconstructed pixel loss for the representation.

$$\mathcal{L}_{Recon} = \sum_{x}^{N} \frac{(\phi(x) - \hat{\phi}(x))^2 * m(x)}{|\phi(x)|}$$

Patch reconstruction error comes from taking the difference between predicted patches $\hat{\phi}$(x) and ground truth ϕ(x). A mask m(x) is applied to this loss for inpainting. We compute masked Mean Squared Error (MSE) for patches at N coordinates [3].



(a) No mapping (b) Basic mapping (c) Gaussian Fourier feature mapping (d) Ground truth

Figure 1: Vanilla neural field reconstructions with different types of Fourier feature mapping after being trained for 2000 iterations.

$$\mathcal{L}_{Pixel} = \sum_{x}^{N} |\hat{\rho}(\hat{\phi}(x)) - c(x)|$$

Reconstructed pixel loss is computed as an l_1 loss between pixel color output $\hat{\rho}(\hat{\phi}(x))$ and ground truth pixel color c(x) [3].

3.5. Masked Autoencoder

Our MAE model and approach comes from [1]. First, the MAE is pre-trained on CIFAR10 training images. We mask a certain percent of random pixel patches when pre-training and the MAE reconstructs the original image. Pre-training is unsupervised. During pre-training, the encoder takes the masked images and creates a latent representation. This representation is then passed to the decoder along with the mask tokens and the original image is reconstructed from these inputs. Fine-tuning is then used to evaluate the model by performing image recognition on test images to get an accuracy, which is supervised. The decoder is discarded for fine-tuning because there are no masked patches to reconstruct. Only the encoder is used in fine-tuning.

3.5.1 Pre-training and Fine-tuning

During pre-training, we mask random pixel patches and then reconstruct the original image using both the MAE encoder and decoder. We used a 75% mask ratio, as this was found to be the best masking ratio in [1] for increasing fine-tuning and linear probing accuracy. Our pre-training loss was only computed for masked patches.
We used fine-tuning to evaluate our model on image recognition with the CIFAR10 test set. We passed un-masked images into the encoder and got the recognition accuracy for test images.

4. Experiments

4.1. Datasets

For our image completion task - inpainting, we masked two 64 image batches of CIFAR10 test images for all models. We used 50,000 CIFAR10 training images to pre-train our MAE. When evaluating the reconstructions from the MAE, we compared the test sets of 10,000 CIFAR10 images, 10,000 CIFAR100 images, and 26,032 SVHN images. All test and training images were 32x32 pixels.

4.2. Vanilla Neural Field Architecture

5 linear layers followed by ReLU activation functions, then a linear output layer followed by sigmoid activation function. This neural field takes positionally encoded pixel coordinates as input and outputs a pixel color representation.

Parameter	Value
Optimizer	Adam
Learning Rate	0.01, 0.001
Batch Size	64
Mapping	Gaussian Fourier feature (Gauss 1.0)

Table 1: Vanilla Neural Field Parameters

4.3. Neural Knitwork Architecture

Patch MLP:
5 linear layers followed by ReLU activation functions, then a linear output layer followed by sigmoid activation function. Takes positionally encoded pixel coordinates as input and outputs 3x3 patches.

MLP Reconstructor:
5 linear layers followed by ReLU activation functions, then a linear output layer followed by sigmoid activation function. Takes 3x3 patch input with 27 (3x3x3) values per coordinate and outputs RGB pixels.

Parameter	Value
Optimizer	Adam
Learning rate	0.01, 0.001
Batch size	64
Mapping	Gaussian Fourier feature (Gauss 1.0)

Table 2: Neural Knitwork Parameters. Parameters are the same for Patch MLP and MLP Reconstructor

4.4. MAE ViT Architecture

We used the MAE ViT models from Masked Autoencoders [1]. Our base, large, and huge models

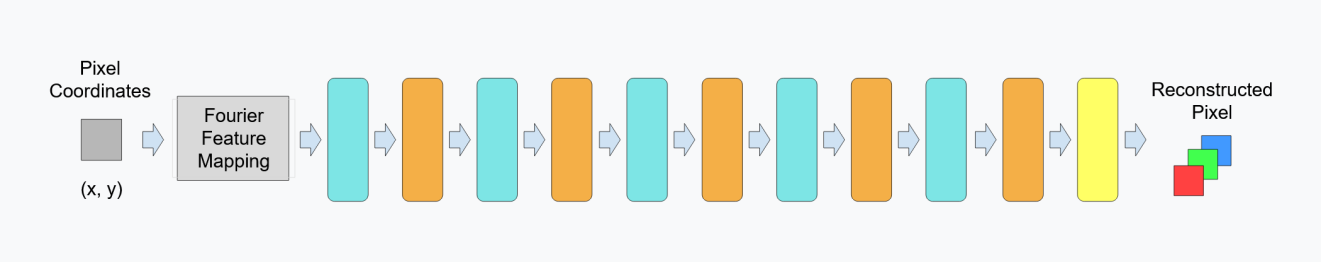


Figure 2: Vanilla Neural Field architecture, from [2]. Fourier feature mapping is done on the coordinates of the input image. These encoded coordinates are then passed to a neural field, which reconstructs the original image into a pixel representation. This neural field, or coordinate-based MLP, is shown as a series of 10 blocks. The blue blocks represent the 5 linear layers, followed by ReLU activation functions represented by orange blocks. The final yellow block represents the Sigmoid activation function on the output layer.

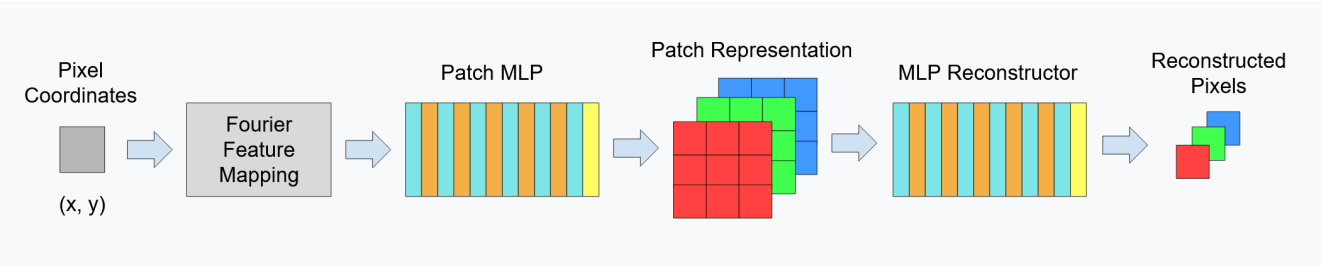


Figure 3: Neural Knitwork architecture, from [3]. Fourier feature mapping is done on the coordinates of the input image. These encoded coordinates are then passed to the Patch MLP, which outputs a patch representation with 3x3 overlapping patches. A single RGB 3x3 patch is shown for the Patch Representation. The MLP Reconstructor takes these patches and returns the reconstructed image in the form of a pixel representation. The Patch MLP and MLP Reconstructor use the same architecture as the vanilla neural field.

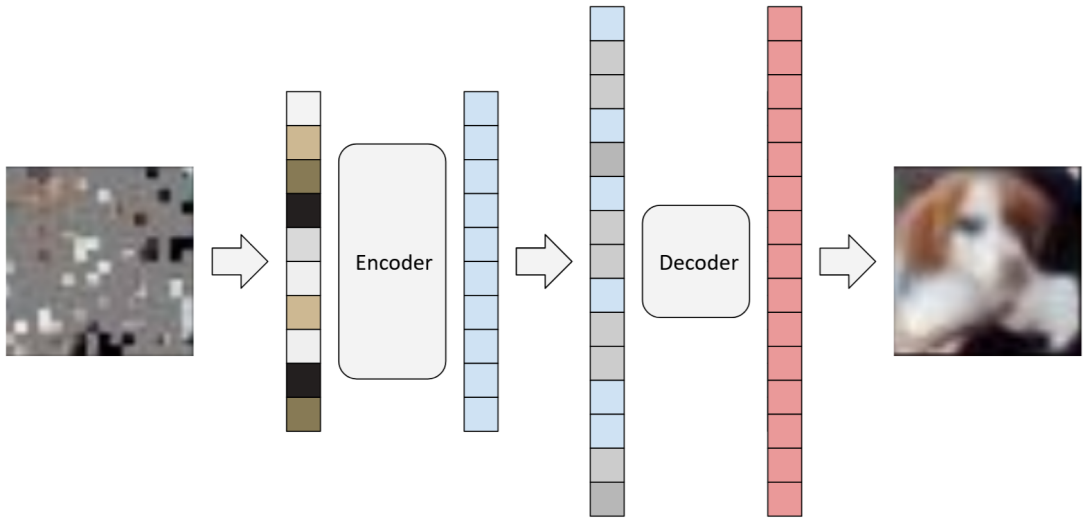


Figure 4: MAE architecture, from [1]. The input image is masked and passed to the encoder. The encoder takes the unmasked patches and maps them to a latent representation. This latent representation, along with mask tokens, is then passed to the lightweight decoder which reconstructs the input image.

have the same architecture as theirs except for our patch sizes of 2x2 and 4x4 instead of 16x16 and 14x14. We use a MAE pre-trained on CIFAR10 training images for all experiments.
MAE ViT Base model:
Encoder - embedding dimension: 768, depth: 12, attention heads: 12
Decoder - embedding dimension: 512, depth: 8, attention heads: 16
MAE ViT Large model:
Encoder - embedding dimension: 1024, depth: 24, attention heads: 16
Decoder - embedding dimension: 512, depth: 8, attention heads: 16
MAE ViT Huge model:
Encoder - embedding dimension: 1280, depth: 32, attention heads: 16
Decoder - embedding dimension: 512, depth: 8, attention heads: 16

Parameter	Value
Optimizer	AdamW
Base learning rate	1.5e-4
Batch size	64
Weight decay	0.005
Optimizer momentum	β_1, β_2 = 0.9, 0.95
Learning rate schedule	cosine decay
Warmup epochs	40
Augmentation	RandomCrop, RandomHorizontalFlip

Table 3: Pre-Training Parameters

Parameter	Value
Optimizer	AdamW
Base learning rate	1e-3
Batch size	64
Weight decay	0.005
Optimizer momentum	β_1, β_2 = 0.9, 0.999
Layer-wise lr decay	0.75
Learning rate schedule	cosine decay
Warmup epochs	5
Training epochs	400(B)
Label smoothing	0.1
Mixup	0.8
Cutup	1.0
Drop path	0.1 (B/L), 0.2 (H)

Table 4: Fine-Tuning Parameters

4.5. Inpainting Results

For the image completion task: inpainting, we masked a rectangle from pixel coordinate (10, 10) to (14, 20), about 3% of the image. We calculate the average PSNR of reconstructed images for the first two batches of CIFAR10 test images for these experiments. PSNR is only calculated for the inpainted region. Our MAE, a base MAE ViT model with 2x2 patches, got an average PSNR of 13.9509. Our vanilla neural field got an average PSNR of 9.8905 with a learning rate of 0.001 after training on each image for 5000 iterations. With a learning rate of 0.01, we got an average PSNR of 9.7732. For our Neural Knitwork's Patch MLP, we got an average PSNR of 9.7837 with a learning rate of 0.001 after training on each image for 5000 iterations. For the same training run with the same parameters, our MLP Reconstructor had an average PSNR of 9.7413. With a learning rate of 0.01, our average PSNR for the Patch MLP was 9.5539 and for the MLP Reconstructor was 9.5569. It seems the MAE is able to reconstruct the CIFAR10 images with a similar PSNR and visual accuracy for each image, as seen in figure 11. It has the highest PSNR on average. However, the vanilla neural field and Neural Knitwork can both achieve much higher than average PSNRs on some images and much lower on others, see figure 9. Since each image has its own neural field train on it individually, the same parameters that work very well for one image may not work great for another. This shows up in our wide range of PSNRs as we used the same parameters for every single image. It seems the vanilla neural field and Neural Knitwork have more accurate reconstructions than the MAE for some images but the MAE, on average, performs better.

4.5.1 Learning Rate

Since both the vanilla neural field and Neural Knitwork train a single neural field for each image, the optimal learning rate varies image by image. Learning rates 0.01 and 0.001 tend to produce the highest average PSNRs. However, some images in the batch will still get a very low PSNR with these learning rates. The 0.001 learning rate got a slightly better average PSNR than 0.01 for both models.

4.6. MAE Transformed Image Reconstructions

We found the MAE was able to reconstruct out-of-distribution data very well. The model we used was a 2x2 patch base MAE ViT that we pre-trained for 400 epochs on CIFAR10. This model had not seen any other dataset before evaluation. The images we fed this model had a 75% mask ratio. First we compared CIFAR10 vs CIFAR100 images and found the MAE reconstructed CIFAR100 images just as well as. Their reconstruction error distributions heavily overlapped, as seen in figure

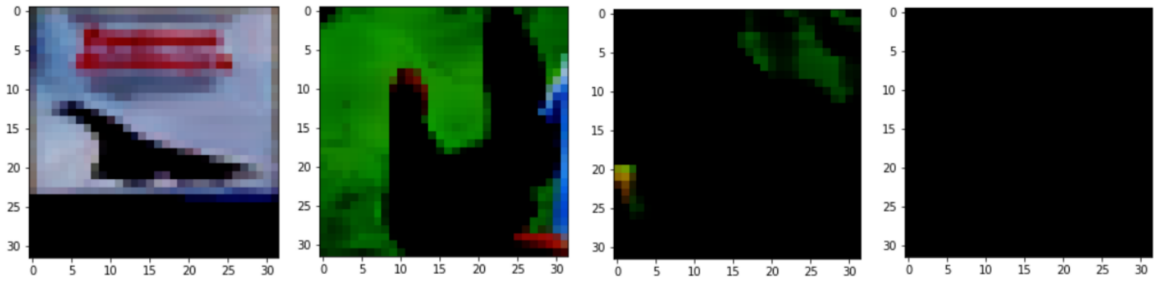


Figure 5: Vanilla Neural Field reconstructions with learning rate of 0.001. PSNRs left to right: 23.0794, 14.6108, 5.7339, and 2.9864.

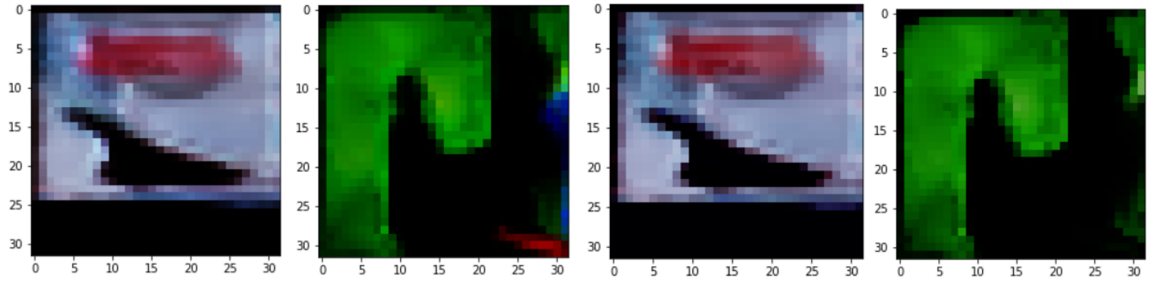


Figure 6: Neural Knitwork reconstructions. Left: Patch MLP reconstructions with PSNRs of 17.6551 and 14.6832. Right: MLP Reconstructor reconstructions with PSNRs of 17.8374 and 14.7784.

Figure 7: One batch of masked CIFAR10 images with rectangular mask from pixel (10, 10) to (14, 20)

Figure 8: One batch of CIFAR10 images reconstructed by MAE from inpainted image.

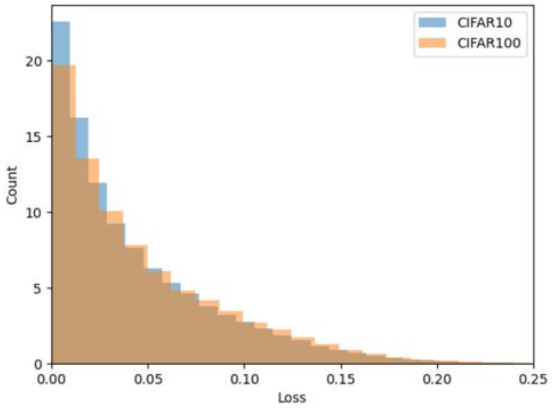


Figure 9: CIFAR10 vs CIFAR100 Reconstruction Error

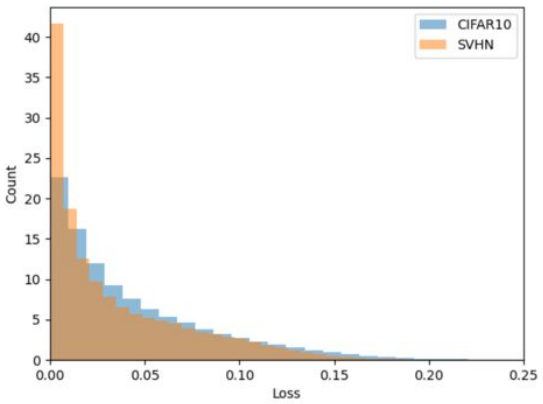


Figure 10: CIFAR10 vs SVHN Reconstruction Error

13. Since CIFAR10 and CIFAR100 contain similar types of images (animals, cars, etc.), we wanted to reconstruct images from a very different looking dataset. For this we used the SVHN dataset, which contains images of numbers 0-9. We found the MAE was able to reconstruct SVHN images just as well as CIFAR10 and CIFAR100, with the same overlapping error distributions (see figure 14 and 15). The MAE was able to reconstruct all three datasets with similar levels of accuracy.

We also tried rotating the test images for CIFAR10 and SVHN at 45° and 90° angles and then reconstructing them. For both angles, the results were the same, overlapping error distributions and all reconstructed images being of similar accuracy (see figure 16). Despite the rotations, the MAE was able to reconstruct all images very well. Next, we tried a mask ratio of 95%, but again the results were the same, just with much lower quality reconstructions due to the very high mask ratio. We also tried a lower mask ratio, 25%, and got the same results. The MAE appears to be very good at generalizing and is robust against transformations.

Figure 11: Reconstructions of images with 75% masking from CIFAR10 (left) and SVHN (right)

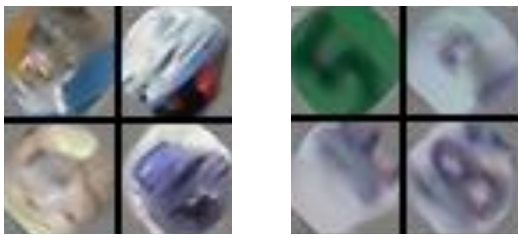


Figure 12: Reconstructions of images rotated 45° with 75% masking from CIFAR10 (left) and SVHN (right)

5. Conclusion

Based on the average PSNR for 2 batches of CIFAR10 test images, the MAE had the best overall reconstructions. The vanilla neural field had the second best PSNR and Neural Knitwork had the third, although they were very close. The vanilla neural field and neural Knitwork both could reconstruct some images better than others, as seen in figure 9, where the PSNR varies by around 10 points. Images where the neural field parameters were optimal had much higher PSNRs than the average PSNR for MAE, but they could also have much lower values if parameters were not optimal.

The MAE was found to be capable of reconstructing OOD images without specifically pre-training on them with the same accuracy as the dataset it had been pre-trained on. It also displayed high robustness against transformations, specifically rotation, when it had never seen these types of transformations in pre-training.

References

[1] Saining X. Yanghao L. Piotr D. Ross G Kaiming H., Xinlei C. Masked autoencoders are scalable vision learners. facebook ai research, 2021.

[2] Ben M. Sara F. Nithin R. Utkarsh S. Ravi R. Jonathan B. Ren N Matthew T., Pratul S. Fourier features let networks

learn high frequency functions in low dimensional domains, 2020.

[3] Robert A. Craig M. Ivan A. Carmine C. Christos T. Mikolaj C., Javier C. Neural knitworks: Patched neural implicit representation networks, 2021.

[4] Takikawa T. Saito S. Litany O. Yan S. Khan N. Tombari F. Tompkin J. sitzmann V. Sridhar S. Xie, Y. Neural fields in visual computing and beyond. computer graphics forum, 2022.