Deep-learning based Hologram Regeneration

Youjing YU, Yuyi ZHANG

Department of Electrical and Electronic Engineering

The University of Hong Kong

25 July - 30 September 2020

Abstract

In recent years, there has been a surge in interest in the development of computer-generated holography(CGH)[1-3]. However, various methods have been proposed to replace the traditional use of spatial light modulators(SLM) due to their incompleteness and reduced image quality.

The rise of artificial intelligence and machine learning witnessed by the 21st century opens a whole new window of possibilities for calculating CGH[4-5]. As interns working under the Department of Electrical and Electronic Engineering, we have been tasked with evaluating and developing various deep-learning algorithms to test their image regeneration abilities. Our work is based mainly on the paper *Deep-learning-generated Holography* by Dr. Ryoichi Horisaki from Osaka University[6]. We generated holography images from the images using computer programs, pass the datasets in for training, and regenerate the original images using deep learning techniques.

Keywords: Deep learning, holography

Method

Datasets

In the hologram-generating phase and the image-regeneration phase, we have tried a total of 7 datasets and finally decided on 5 to explore more deeply. These five datasets are:

(1) the COCO datasets;

(2) computer-generated random patterns;

(3) the MNIST hand-written digit dataset;

(4) Arabic hand-written digit dataset;

(5) optically produced hologram.

COCO contains 40670 image pairs in total, which were generated from COCO 2017 test and validation images. Due to the GPU memory limit, our networks were trained with 2000 or 12500 COCO images. Other datasets each contain 2000 image pairs for training and another 300 image pairs for testing, except for the hologram dataset, which we only have 240 image pairs for training and 20 pairs for testing. All images in the above datasets are of size 256x256.

Hologram-regeneration

Following the paper by Dr. Horisaki, we wrote a MATLAB program to generate Fresnel diffraction patterns from the original images. The pixel counts of the original images and the target intensity patterns were both 256x256, which was 4x4 times greater than the datasets used by Dr. Horisaki(64x64). The optical wavelength was set to be 632 nm, pixel pitch 30 μm, and the object distance 13 cm.

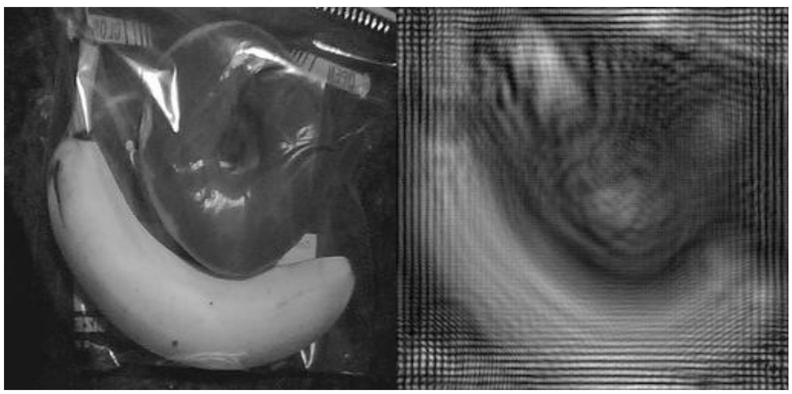


Figure 1. an example of an image pair in dataset COCO

Deep-learning algorithms

During the training phase, we have experimented with ten network structures, namely SqueezeNet, ResNet, Wide ResNet(WRN), WRN with fewer layers(WRN-fewer), DenseNet and their respective generative adversarial network(GAN) structures. Our ResNet structure was based on the structure proposed by Dr. Horisaki except for adding more layers since our image size was larger, and all other network structures could be found on our code published on GitLab.

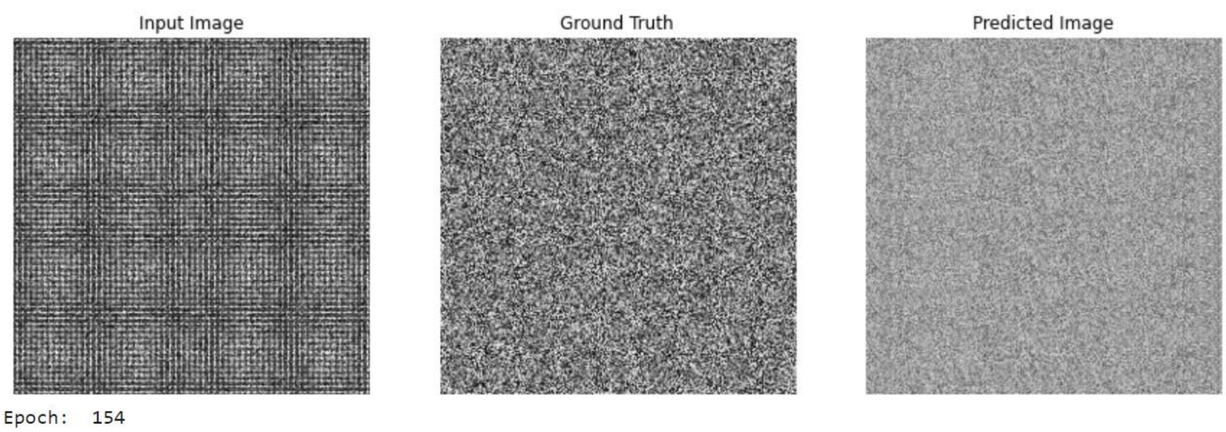
Results

We will present our result section by section according to each dataset because different datasets yield different outcomes to the same network structure.

1.Random Patterns

- **Without GAN**

Out of the six networks tested, as shown in Figure 2, ResNet(Fig 2a) and WRN yield the best results. Although WRN and WRN-fewer(Fig 2b) have about the same losses, WRN-fewer is considered better because it has fewer layers and takes a shorter time to train. DenseNet(Fig 2c) has slightly worse results since the generated image is not smooth, and sharp edges can be seen. However, SqueezeNet(Fig 2d) is not able to yield any result at all.



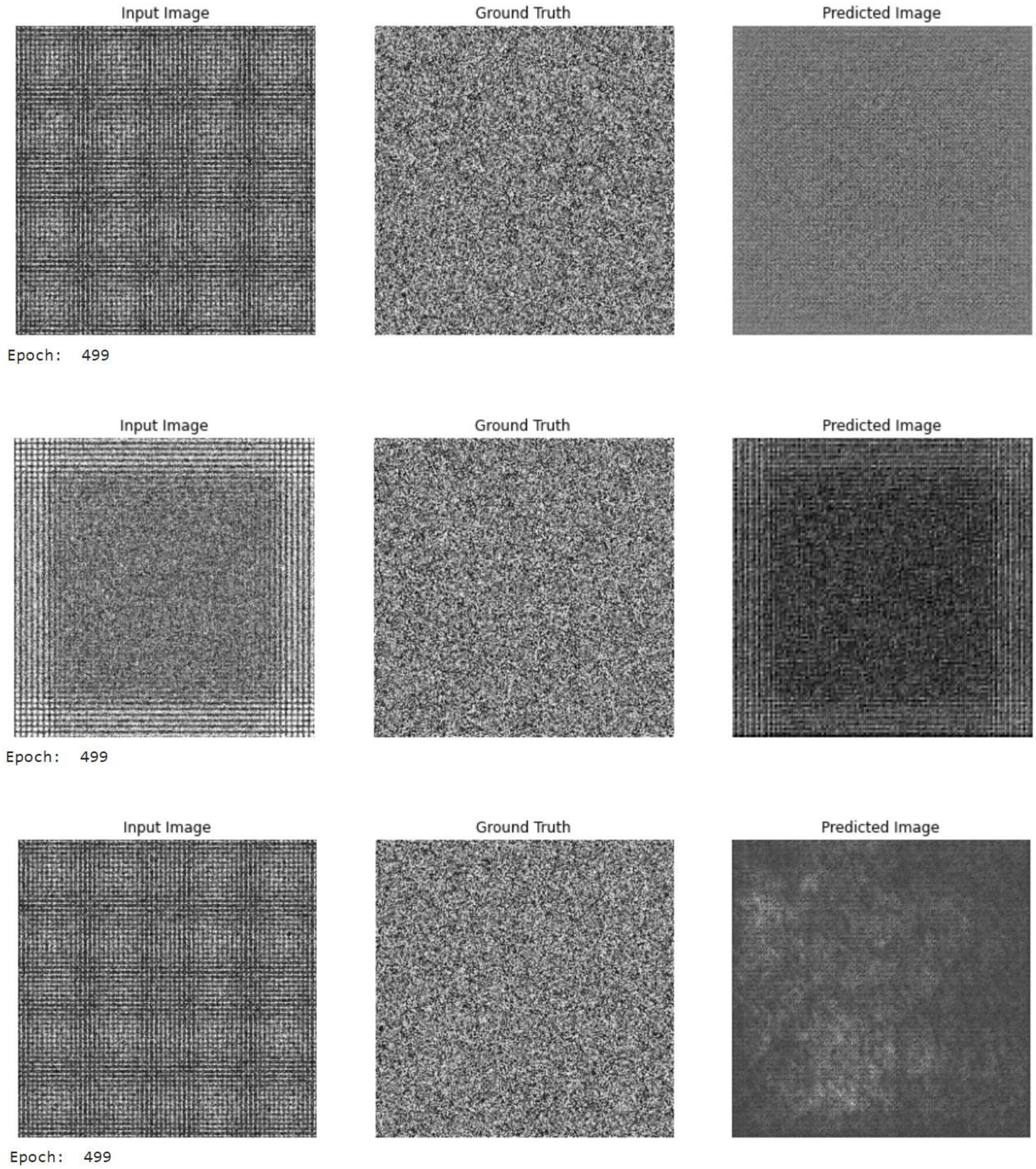
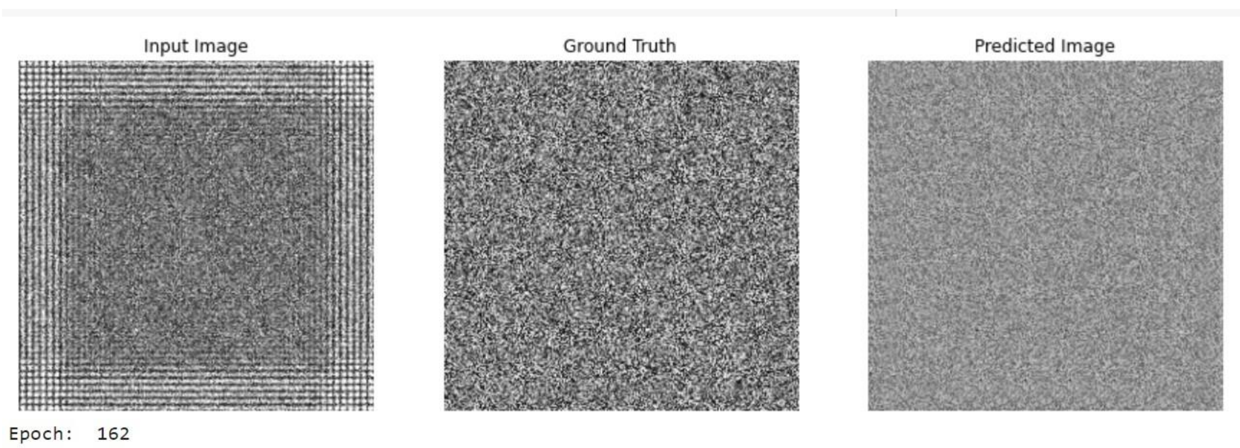
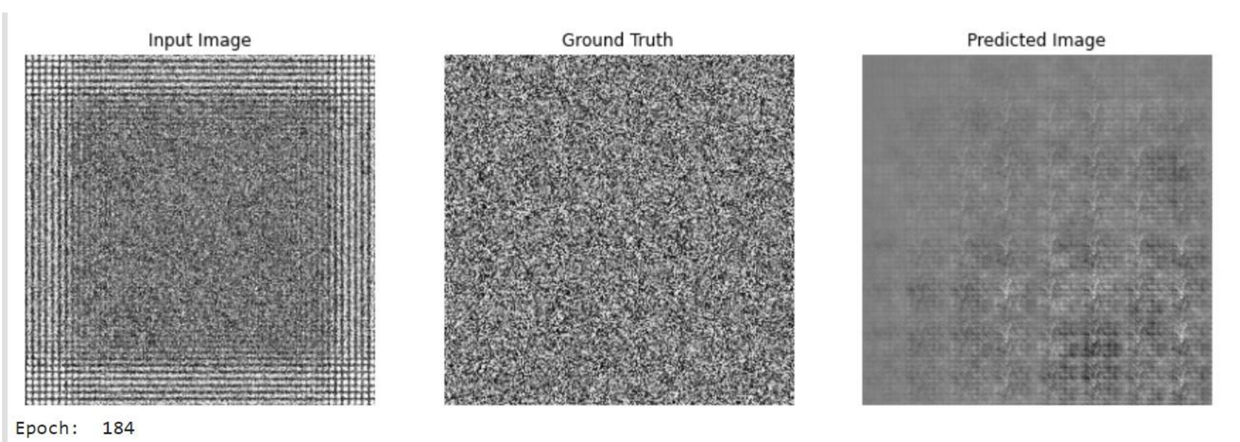


Figure 2. random patterns trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet

- **With GAN**

As shown in Figure 3, ResNet(Fig 3a) yields better results than without the GAN structure with GAN structures. However, WRN(Fig 3b) produces worse outcomes, and DenseNet(Fig 3c) and SqueezeNet(Fig 3d) yield no results at all. DenseNet is only shown until the 18th epoch because the whole network collapses and only produces a black image afterward.





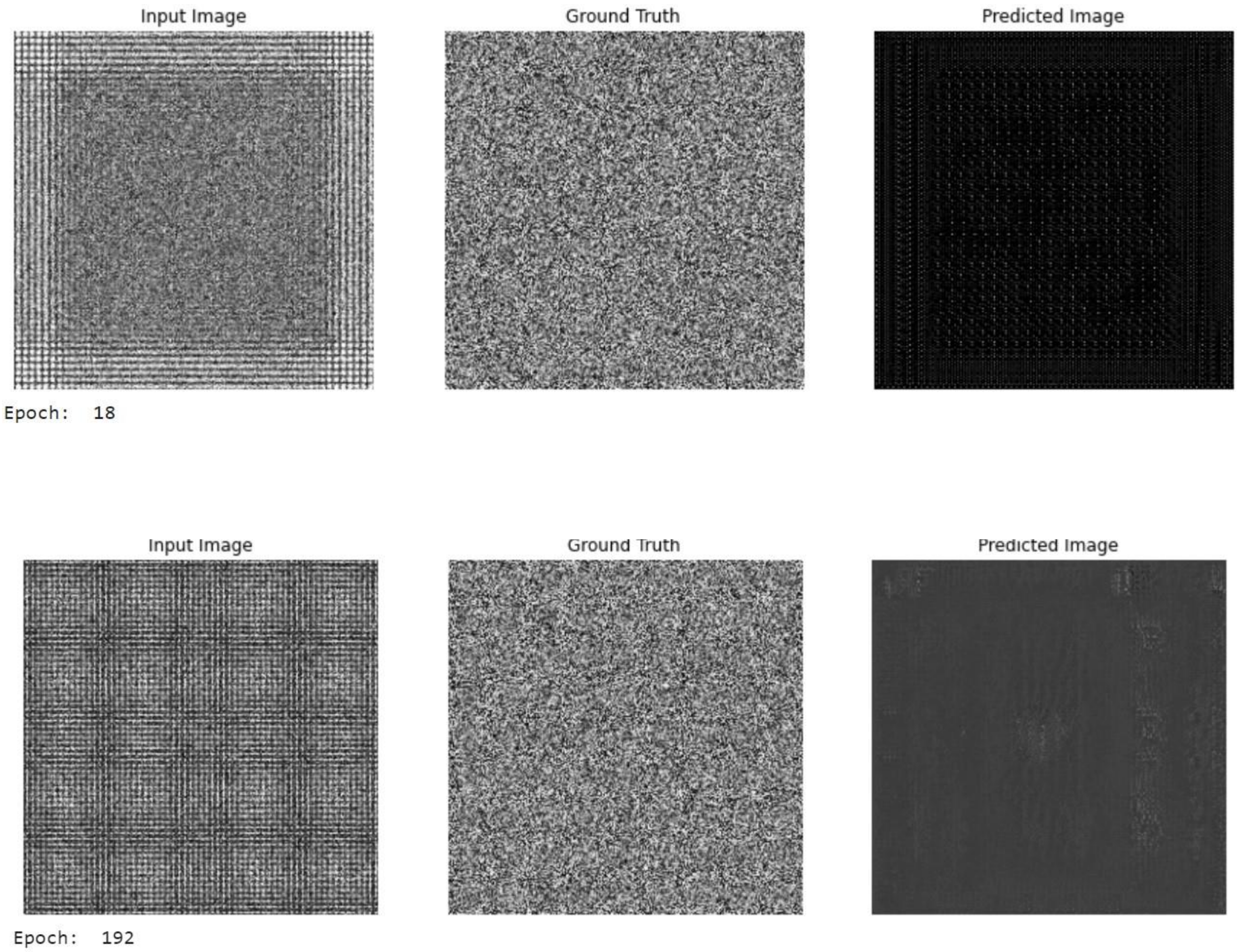
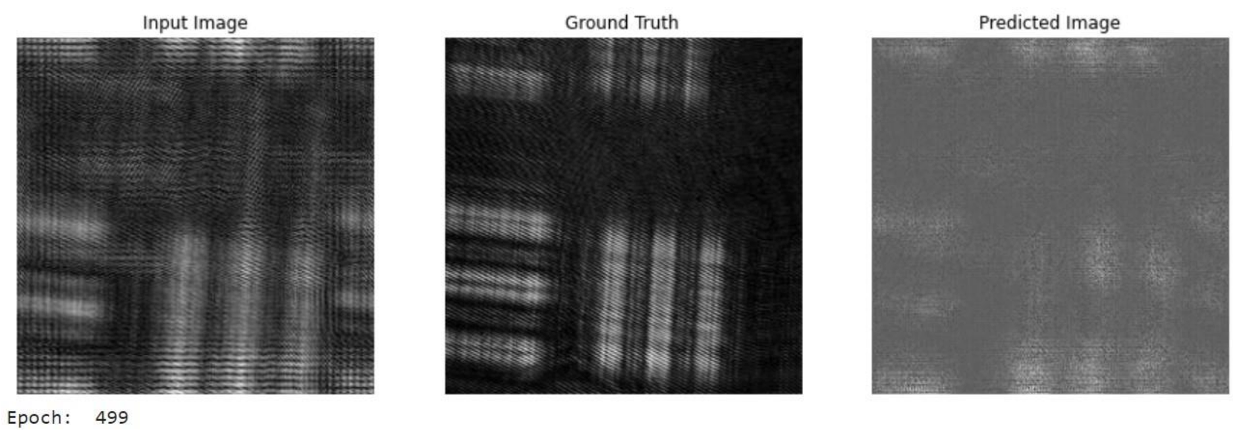


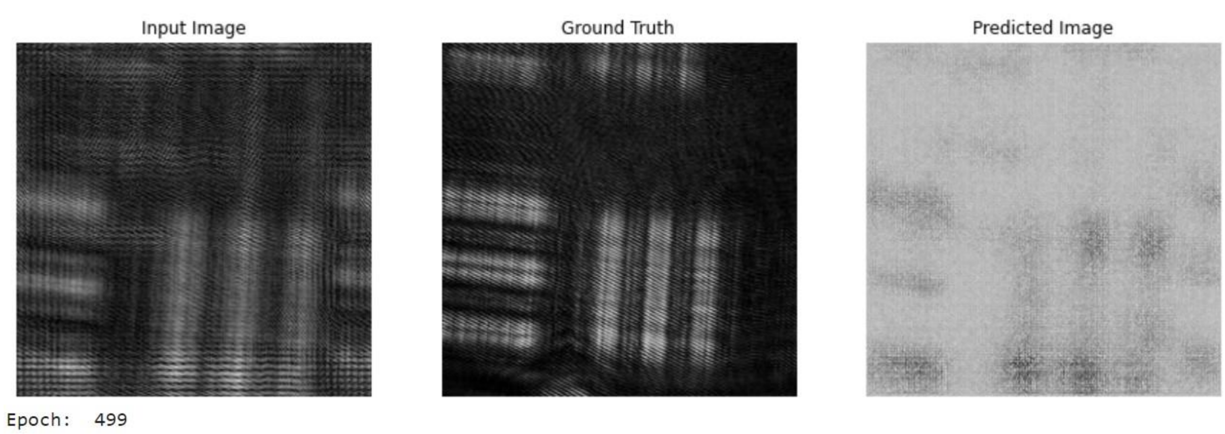
Figure 3. random patterns trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet, together with the GAN structure

2. Hologram

- **Without GAN**

Out of the six networks tested, as shown in Figure 4, ResNet(Fig 4a), WRN(Fig 4b) and DenseNet(Fig 4c) all yield good results, with DenseNet performing slightly better than the other two. However, SqueezeNet is not able to yield any result at all. We believe that this is because the hologram dataset is straightforward. The input patterns are mostly alike, which may explain why ResNet, WRN and DenseNet work well, but not SqueezeNet since SqueezeNet needs a larger dataset to function.





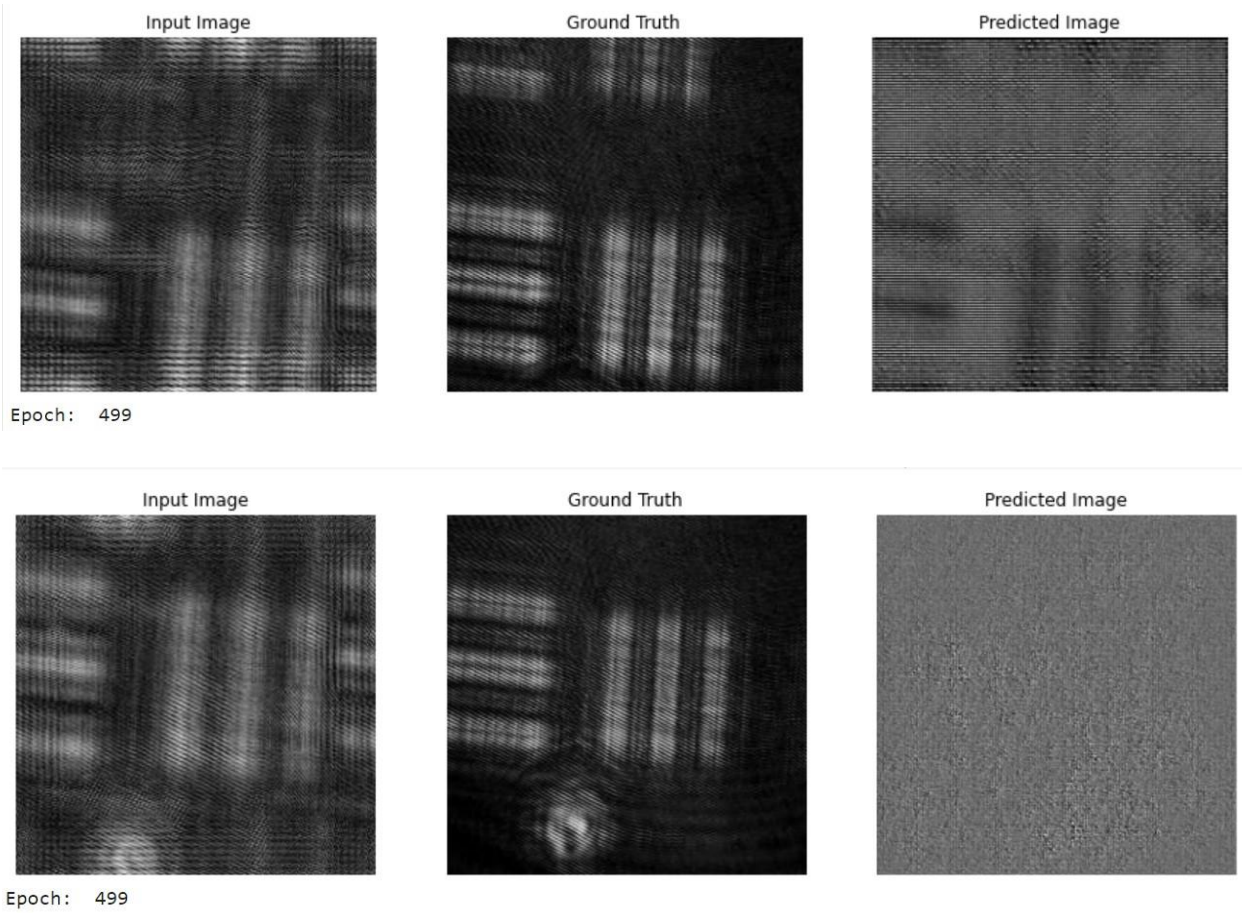
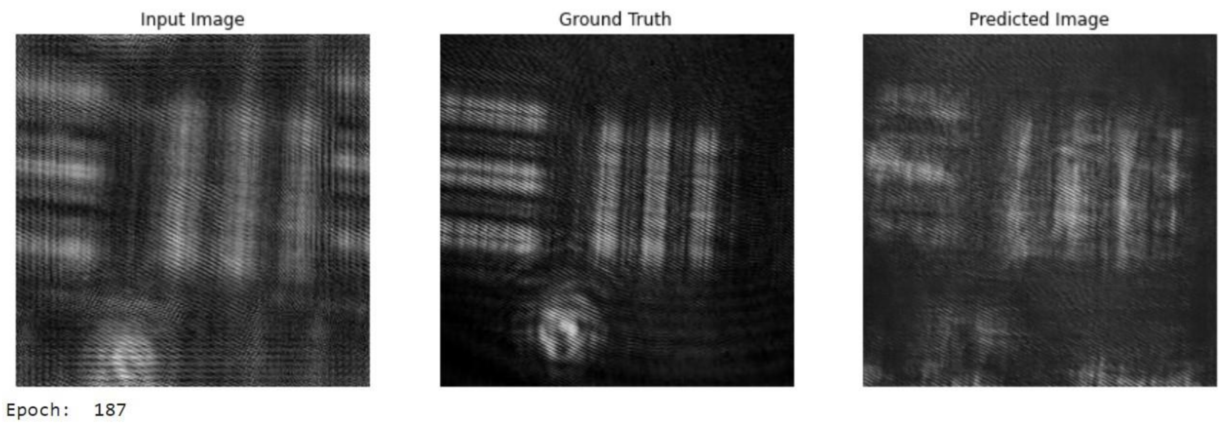
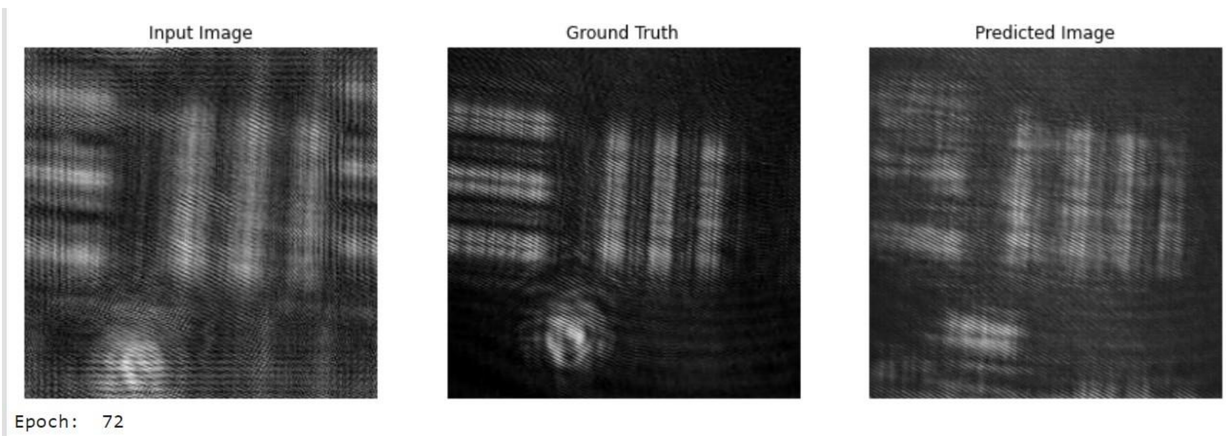


Figure 4. holograms trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet

- **With GAN**

With GAN structures, as shown in Figure 5, ResNet(Fig 5a) and WRN(Fig 5b) yield considerably better results than without the GAN structure. However, DenseNet(Fig 5c) fails to yield anything at all. Most surprisingly, SqueezeNet(Fig 5d) delivers passably good results, though not as good as ResNet and WRN.





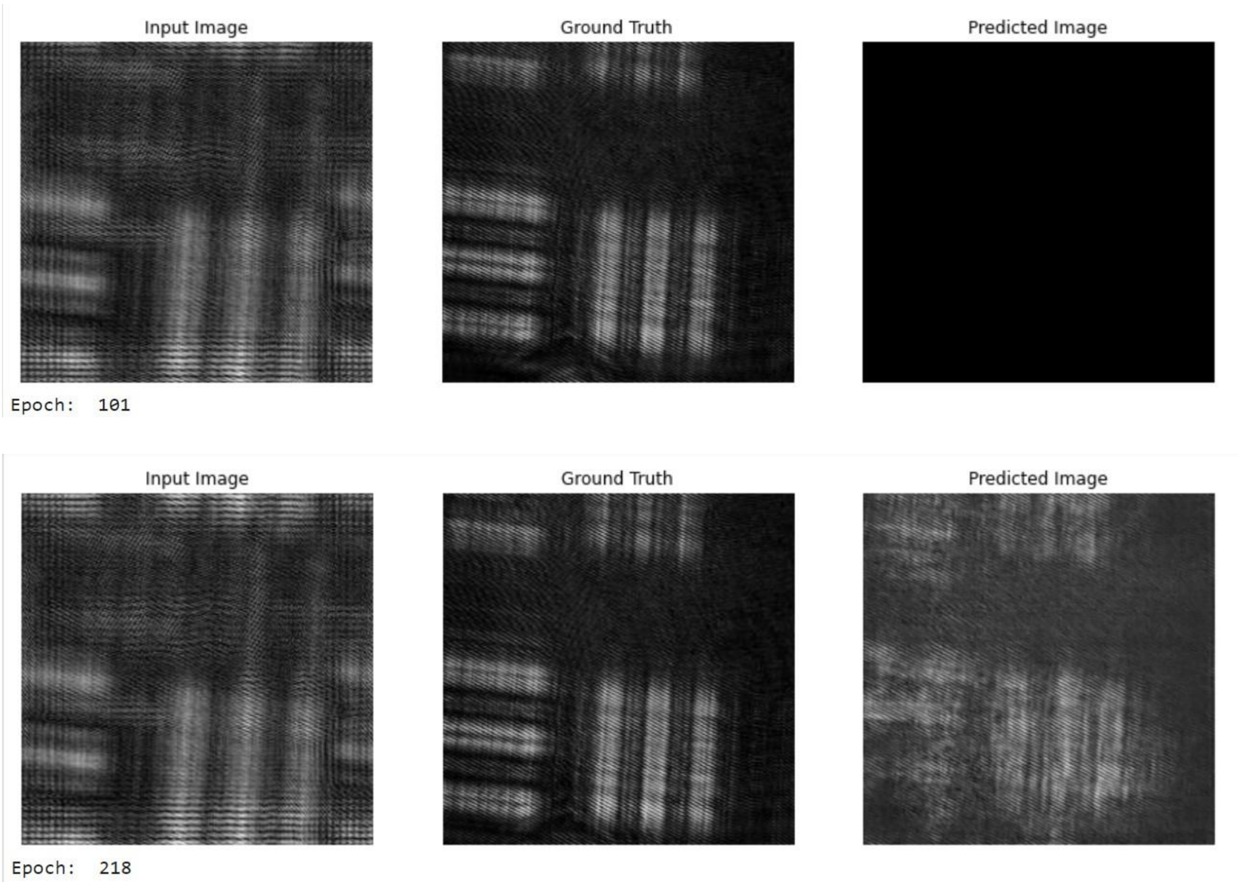


Figure 5. holograms trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet, together with the GAN structure

3. Coco

- **Without GAN**

Out of the six networks tested, as shown in Figure 6, only ResNet(Fig 6a) yields satisfactory results. WRN(Fig 6b), DenseNet(Fig 6c), and SqueezeNet(Fig 6d) do not give anything at all. We believe that this is because the COCO images are much too challenging to learn so that even ResNet only gives shapes of the objects, not the precise picture.

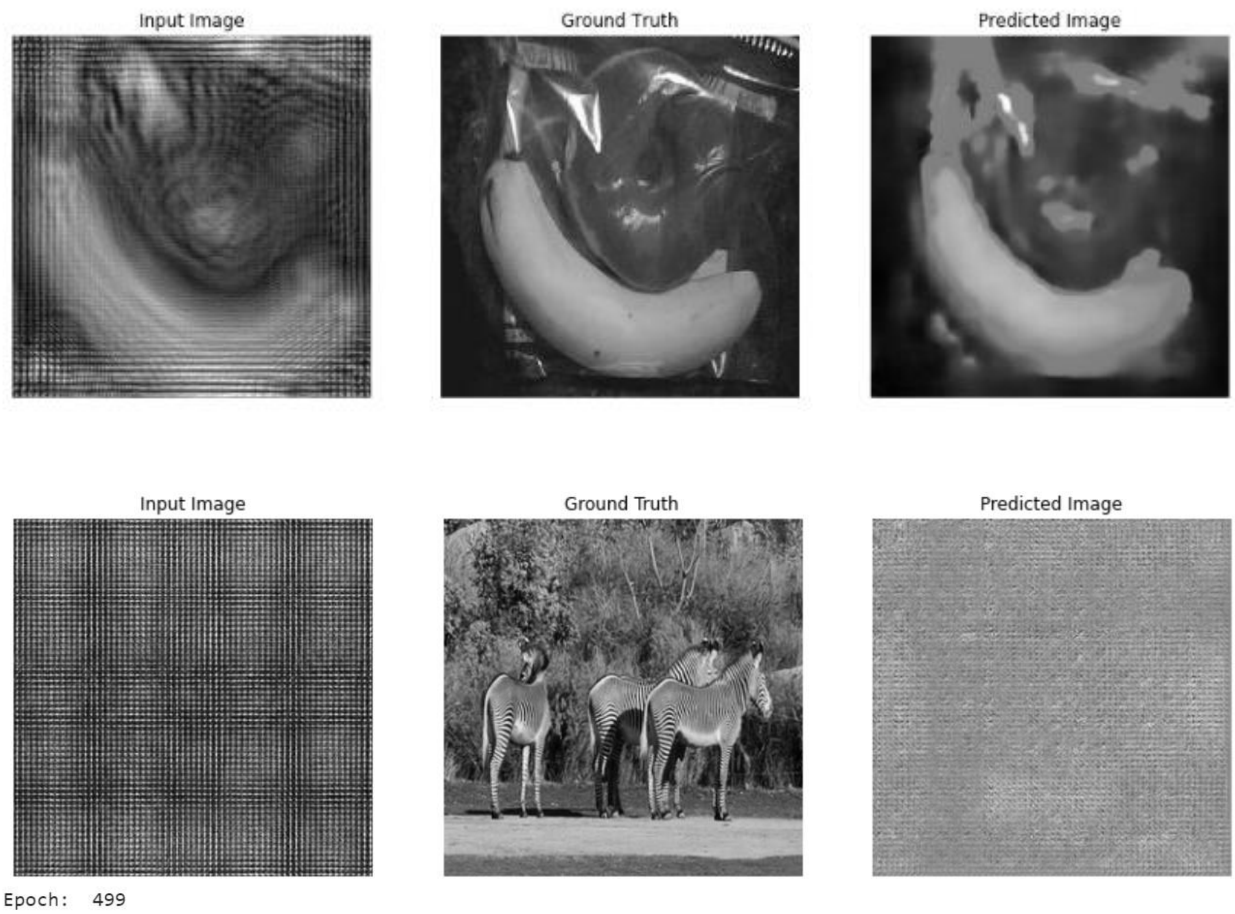
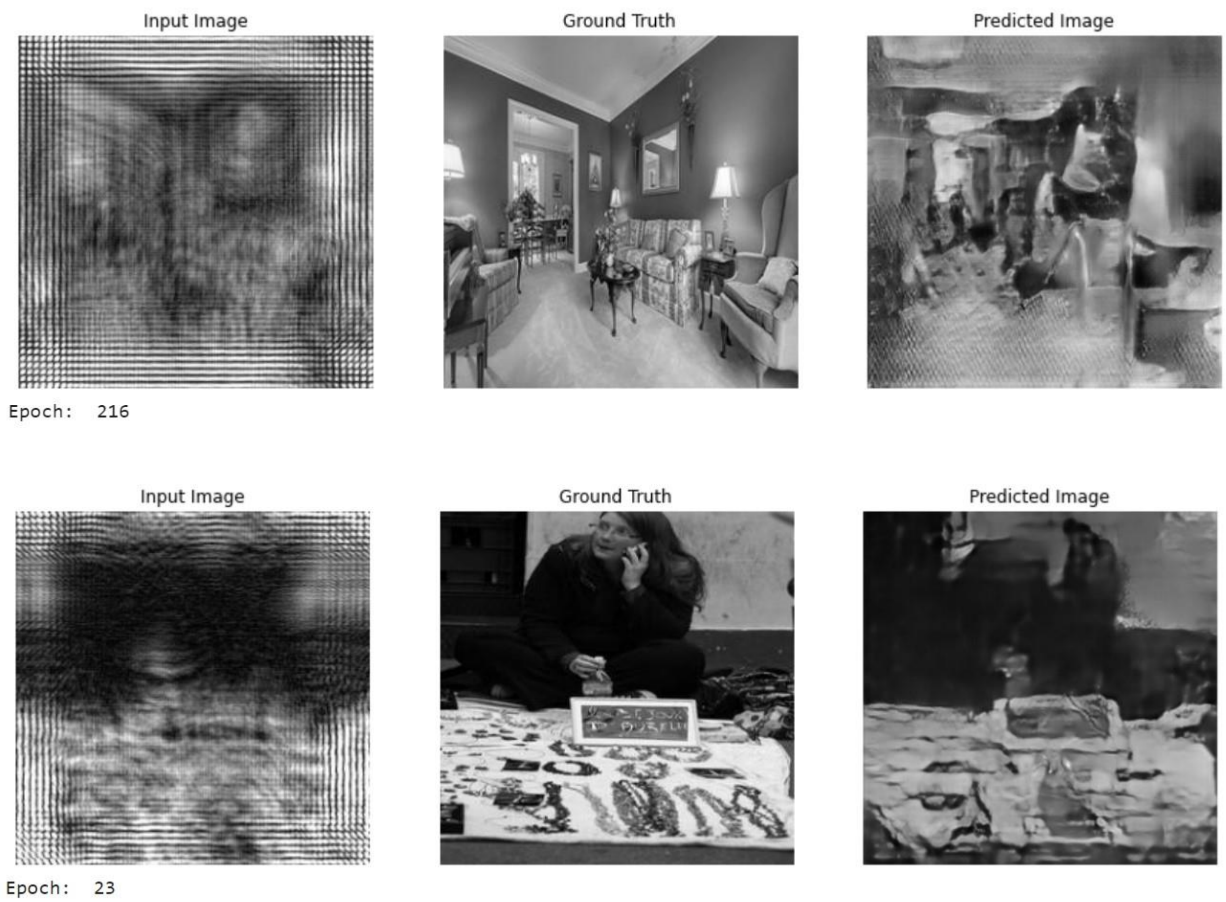


Figure 6. COCO pictures trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet

- **With GAN**

Out of the six networks tested, as shown in Figure 7, all networks yield much better results than the structures without GAN. ResNet(Fig 7a) and WRN(Fig 7b) both perform very well, and DenseNet and SqueezeNet are a little behind in terms of accuracy. However, we notice that the deep learning models can only give us shapes of the objects present in the image, and yielding well-defined pictures is beyond their abilities.



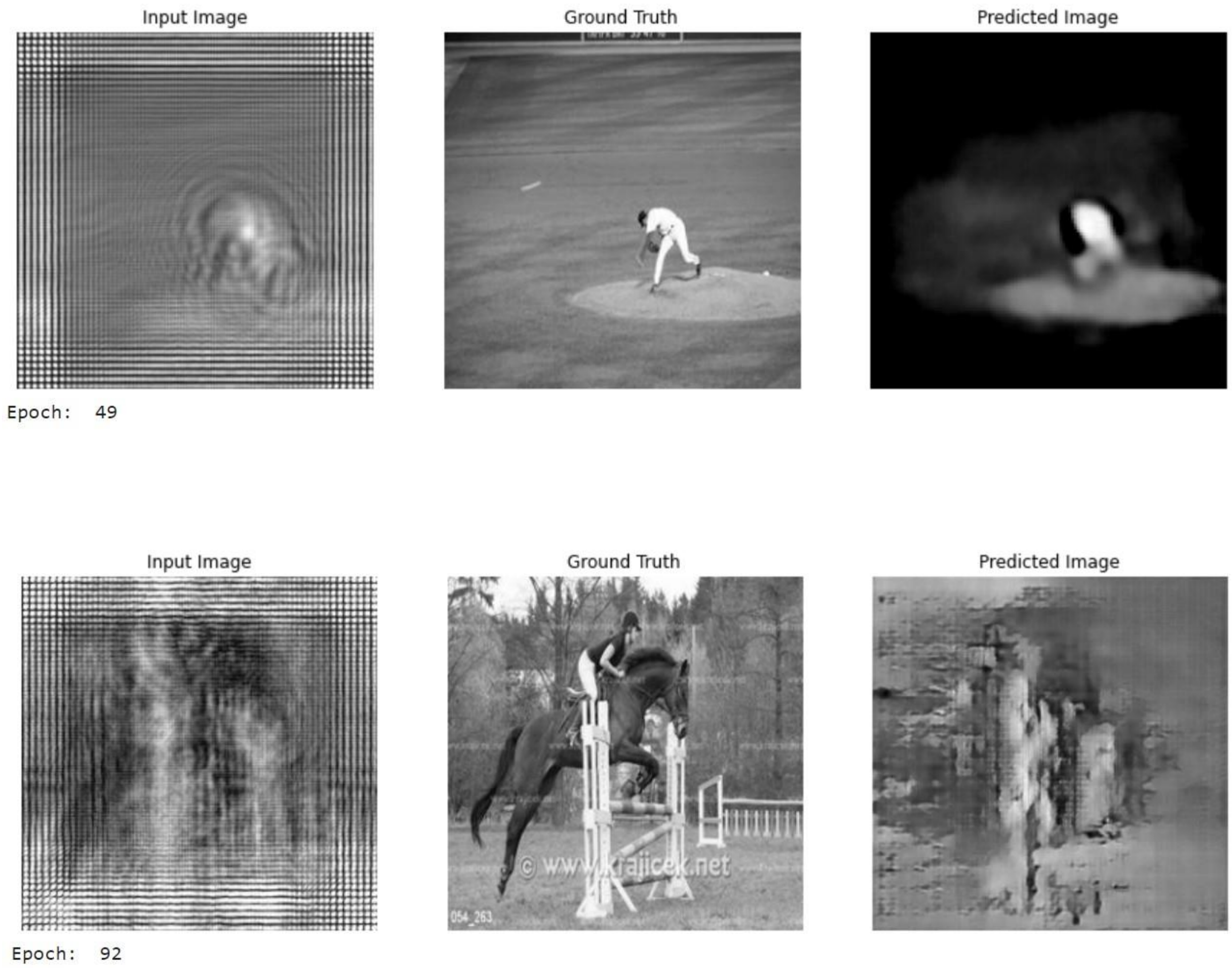
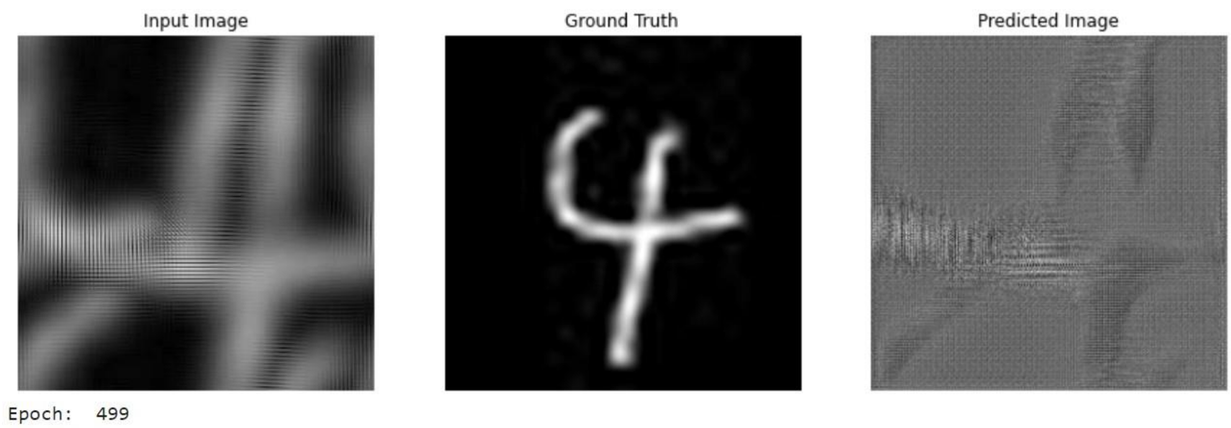


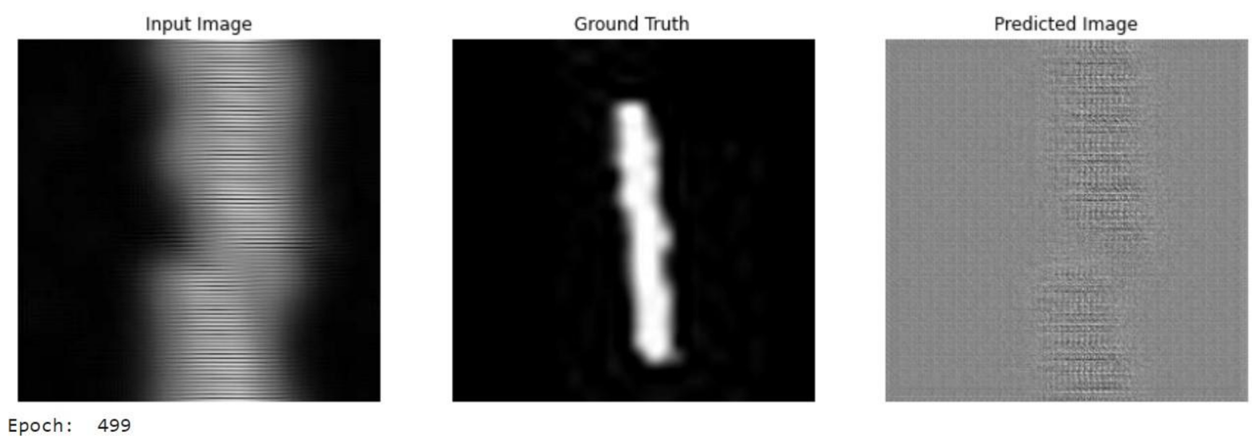
Figure 7. COCO pictures trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet, together with the GAN structure

4. MNIST

- **Without GAN**

As shown in Figure 8, none of the networks show passable results out of the six networks tested. Except for the WRN(Fig 8b), which shows some form of contour, all the other three networks, namely ResNet(Fig 8a), DenseNet(Fig 8c) and SqueezeNet(Fig 8d), do not yield any picture at all.





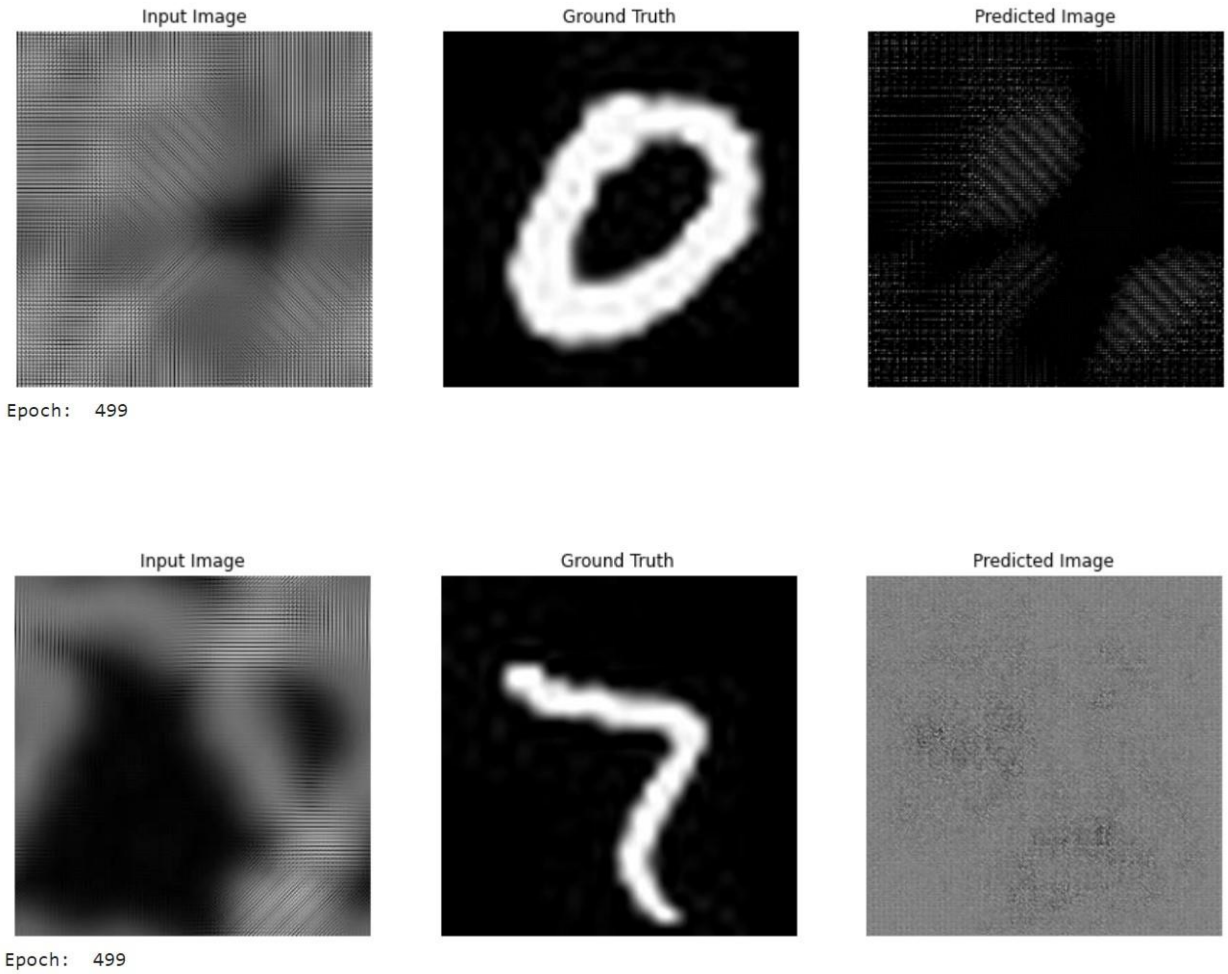
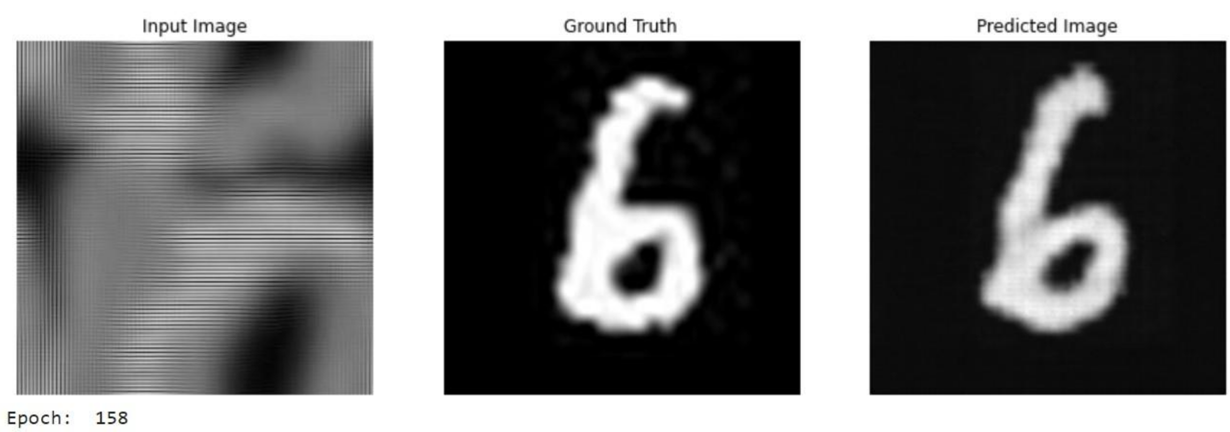


Figure 8. COCO pictures trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet

- **With GAN**

As shown in Figure 9, all networks generally perform well with the GAN structure. In particular, ResNet(Fig 9a) and DenseNet(Fig 9c) yield the best results, while WRN-fewer yields good results while being considerably faster. We attribute this success mainly to the fact that the MNIST dataset is relatively simple and easy compared to the coco dataset.



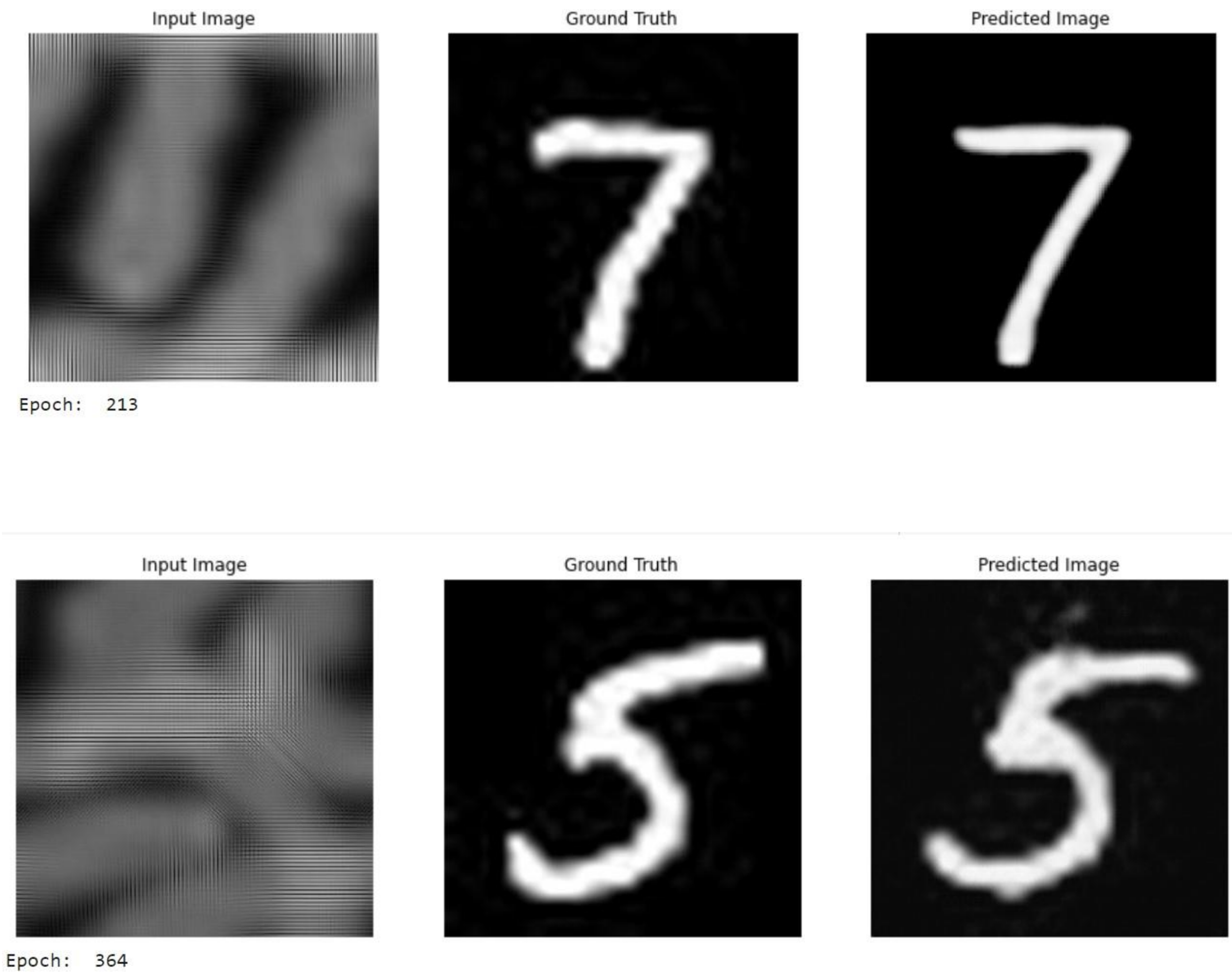


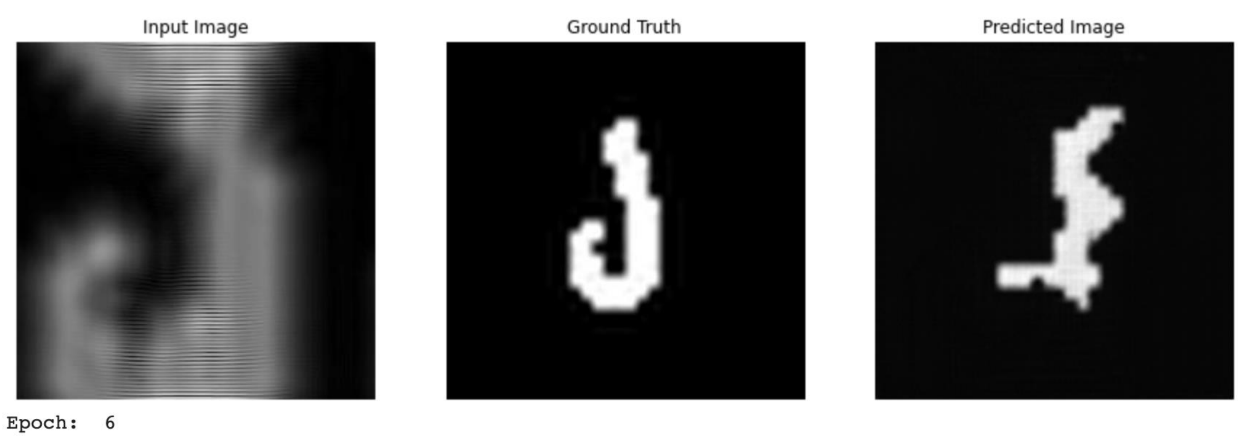
Figure 9. MNIST dataset trained with (a) ResNet, (b)WRN-fewer, (c) DenseNet, (d) SqueezeNet, together with the GAN structure

5. ARABIC

- **With GAN**

As shown by the previous examples, especially the MNIST dataset, networks with GAN structures generally perform better than those without GAN. Hence, for the Arabic character dataset, we only focus on the GAN structure networks.

However, during the training of the Arabic character dataset, a most peculiar pattern has occurred. The networks show satisfactory results, usually after only less than ten epochs, before deteriorating quickly. After a few hundred epochs of training, the networks will collapse, and nothing yields. Here we illustrate using the ResNet and the WRN results. The image regenerated by ResNet is right after six epochs (Fig 10a), but it collapses after 500 epochs(Fig 10b). When trained by WRN, the image after three epochs is good(Fig 10c), but it collapses after 500 epochs(Fig 10d). The model will start to collapse after only 50 or so epochs. The model collapse never happens on other datasets, so we are unsure why the model will collapse so fast, even with data augmentation to prevent overfitting.



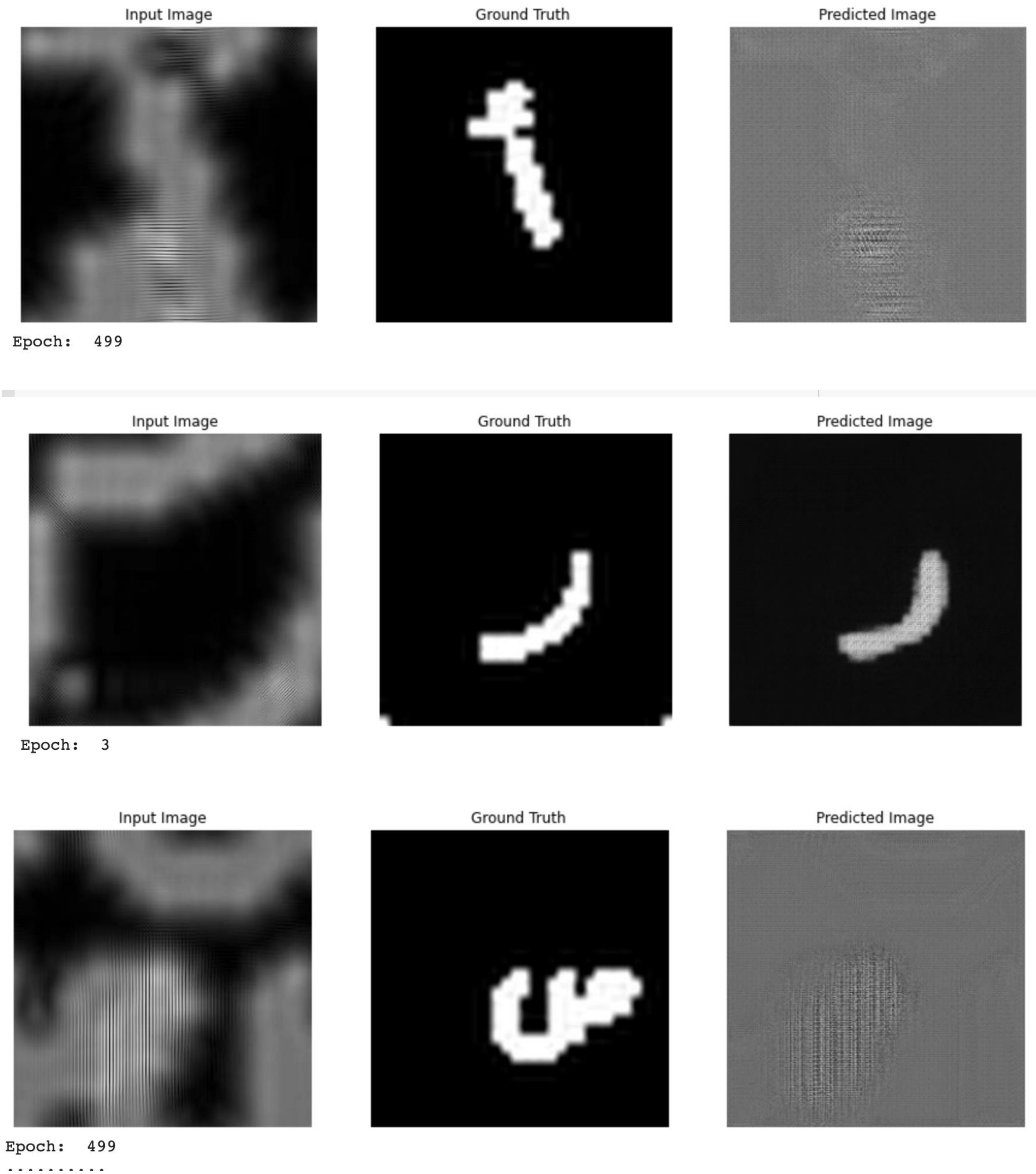
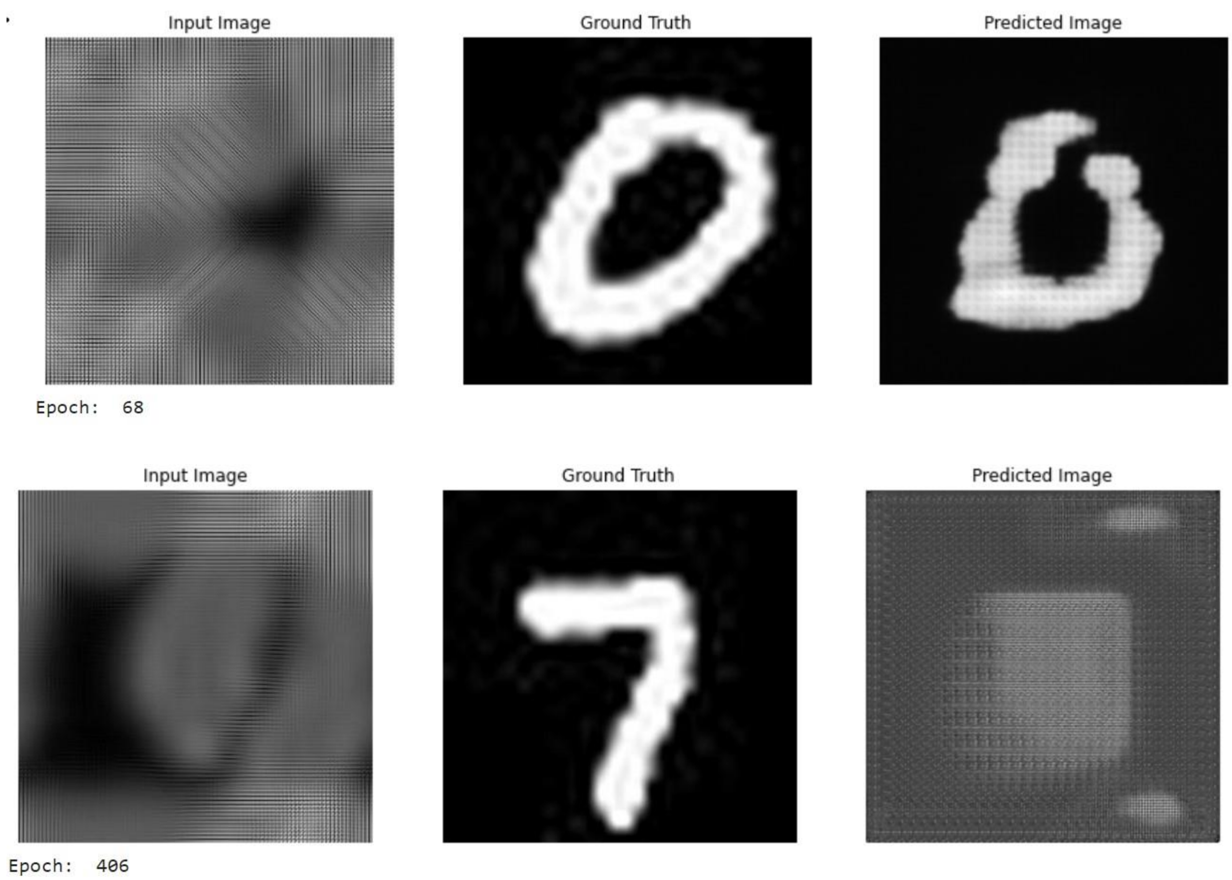


Figure 10. Arabic dataset trained with (a) ResNet after six epochs, (b)Resnet after 500 epochs, (c) WRN after three epochs, (d) WRN after 500 epochs

6. Cross-training

- **Arabic-MNIST**

In the final stage of our research, we tried using models constructed based on one dataset to test another dataset. We first used models trained on the Arabic dataset to test it on the MNIST dataset because both the Arabic and MNIST dataset are relatively simple and easy. However, during training, we noticed the same pattern as on the Arabic dataset. As shown in Figure 11, the ResNet result is relatively good after 68 epochs(Fig 11a) but deteriorates very quickly after 406 epochs. The same happened with WRN, where the result is good after five epochs(Fig 11c), but it collapses after 93 epochs(Fig 11d).



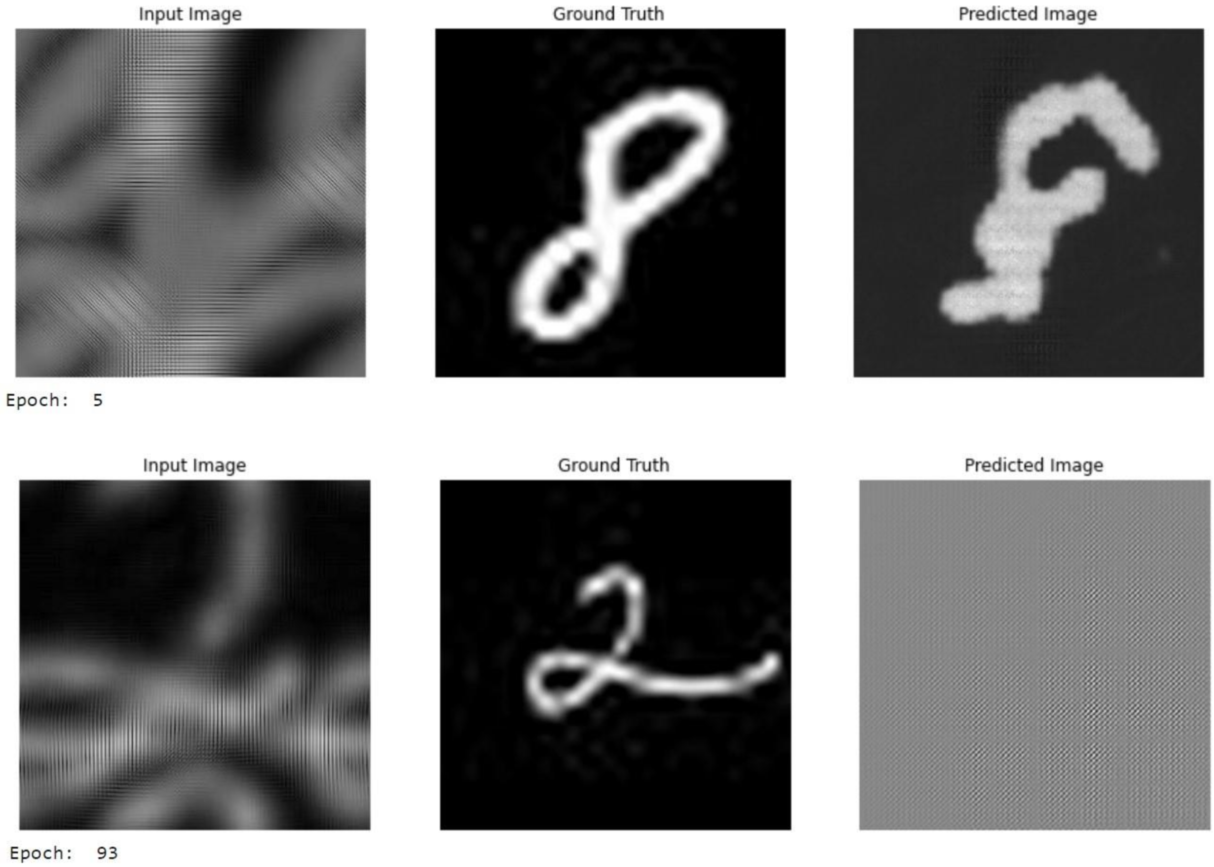
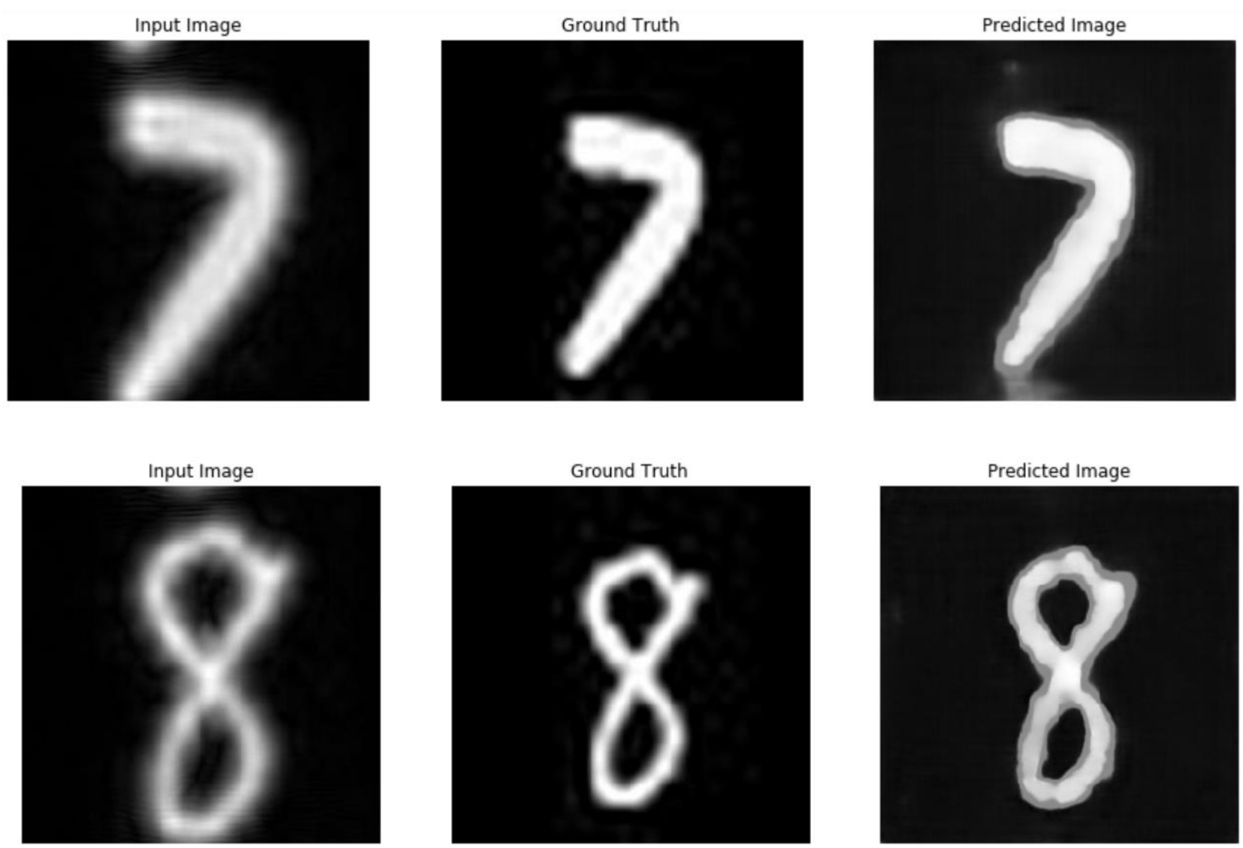


Figure 11. MNIST dataset trained with Arabic characters (a) ResNet after 68 epochs (b)Resnet after 406 epochs (c) WRN after five epochs (d) WRN after 93 epochs

- **Coco-MNIST**

We then trained the coco dataset to test it on the MNIST dataset. Since the COCO images are very complicated, but the MNIST images are relatively simple, we believe that this will give us better results. Indeed, the pictures regenerated from MNIST datasets are very good. As shown in Figure 9, the image regenerated from the diffraction pattern using ResNet is perfect (Fig 12a,b). The ResNet-GAN structure gives passably good results(Fig 12c). However, those generated using WRN are not that good(Fig 12d), which might be because there are too many parameters in WRN that over-fitting has occurred.



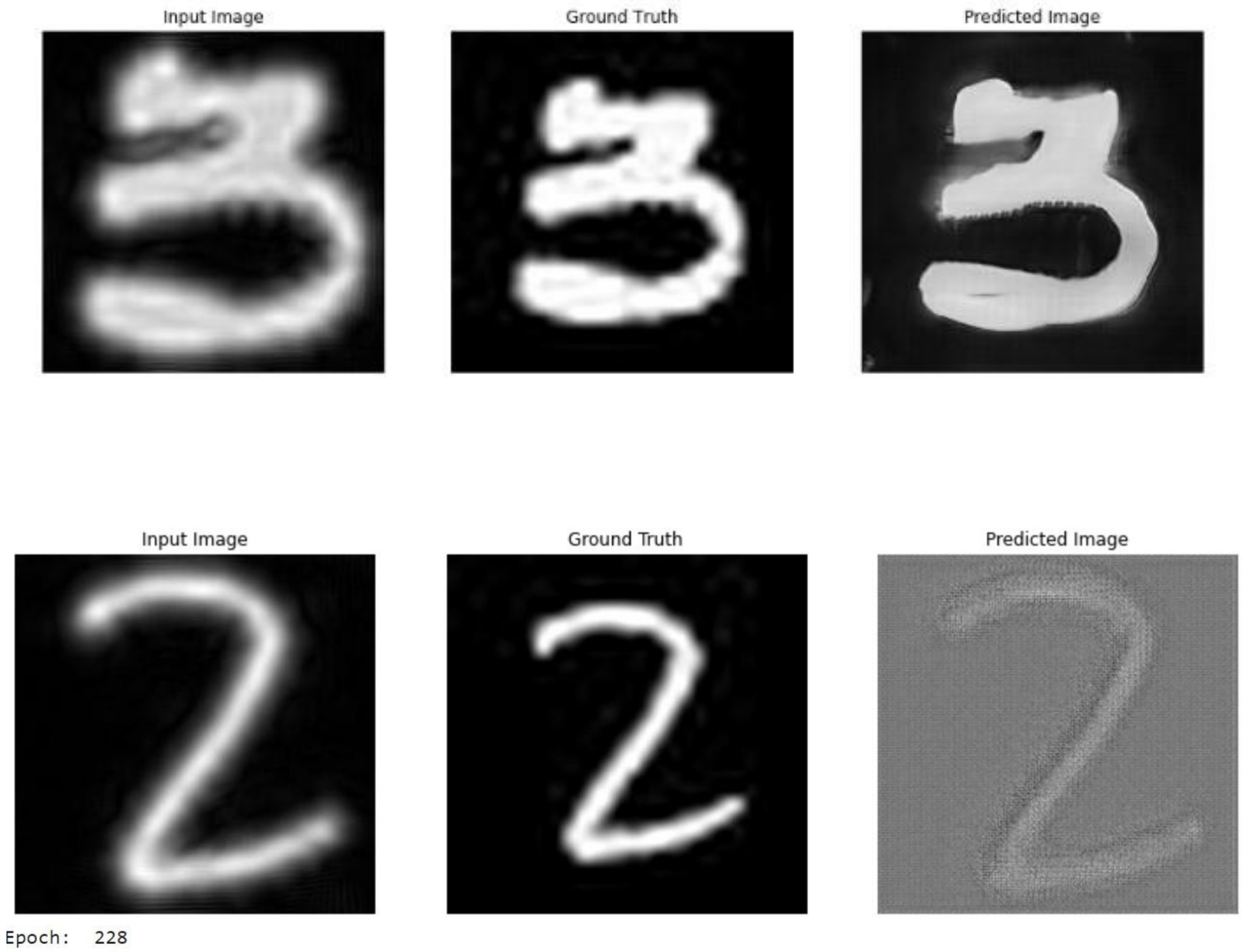


Figure 12. MNIST dataset trained with COCO images (a) ResNet after 94 epochs (b)ResNet after 970 epochs(c) ResNet-GAN after 45 epochs (d) WRN after 228 epochs

- **Coco-Arabic**

We then trained the coco dataset to test it on the Arabic dataset, which is slightly more complicated than the MNIST dataset. However, the result is far from ideal. There are no images generated at the end of the test, with either ResNet(Fig 13a) or WRN(Fig 13b).

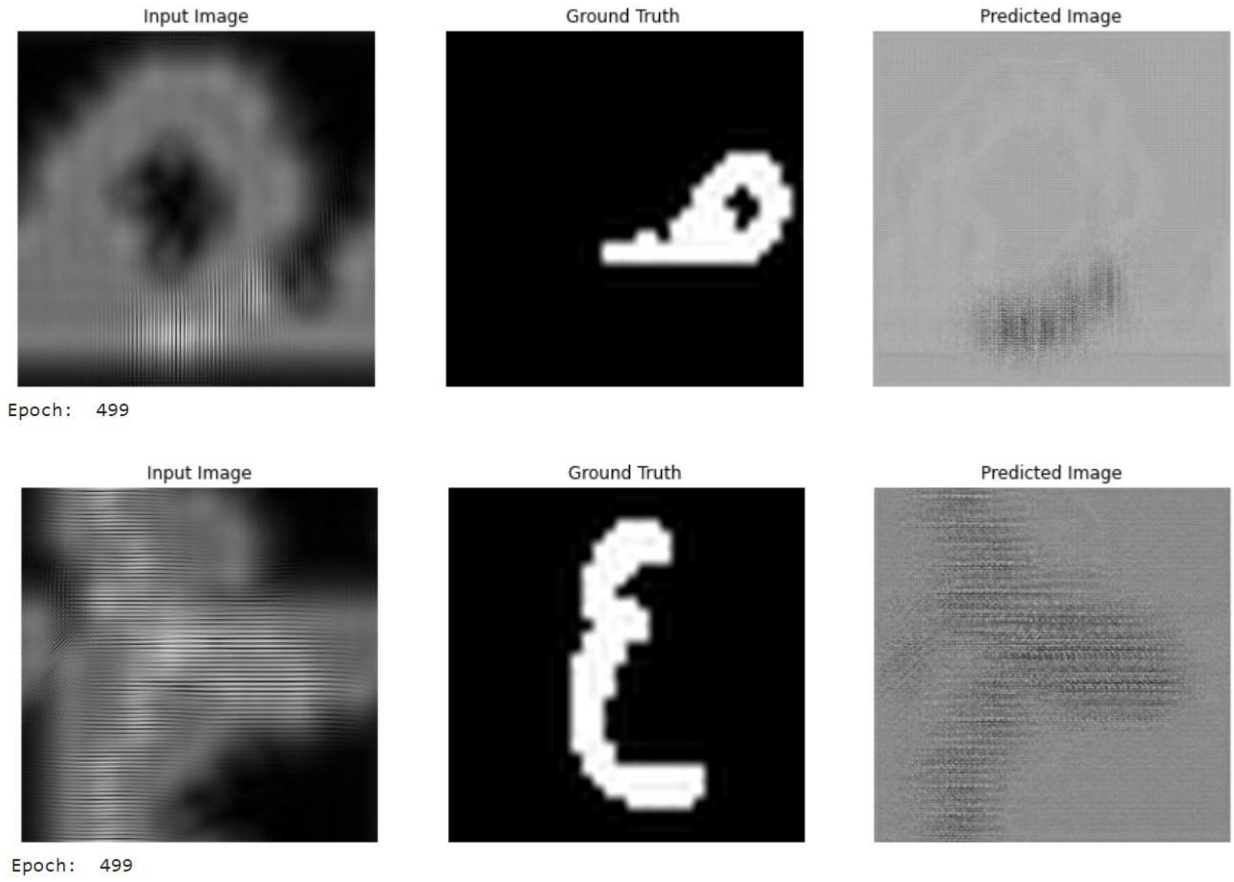


Figure 13. Arabic dataset trained with COCO images (a) ResNet (b) WRN

- **Coco-holo**

Lastly, we trained the coco dataset to test it on the hologram dataset. However, the result is not as satisfactory either, though slightly better than when testing with the Arabic dataset. The ResNet (Fig 14a) gives good results. The ResNet-GAN structure (Fig 14b) gives somewhat worse results, though passable as well. The WRN(Fig 14c) performs worse, as only a contour can be seen.

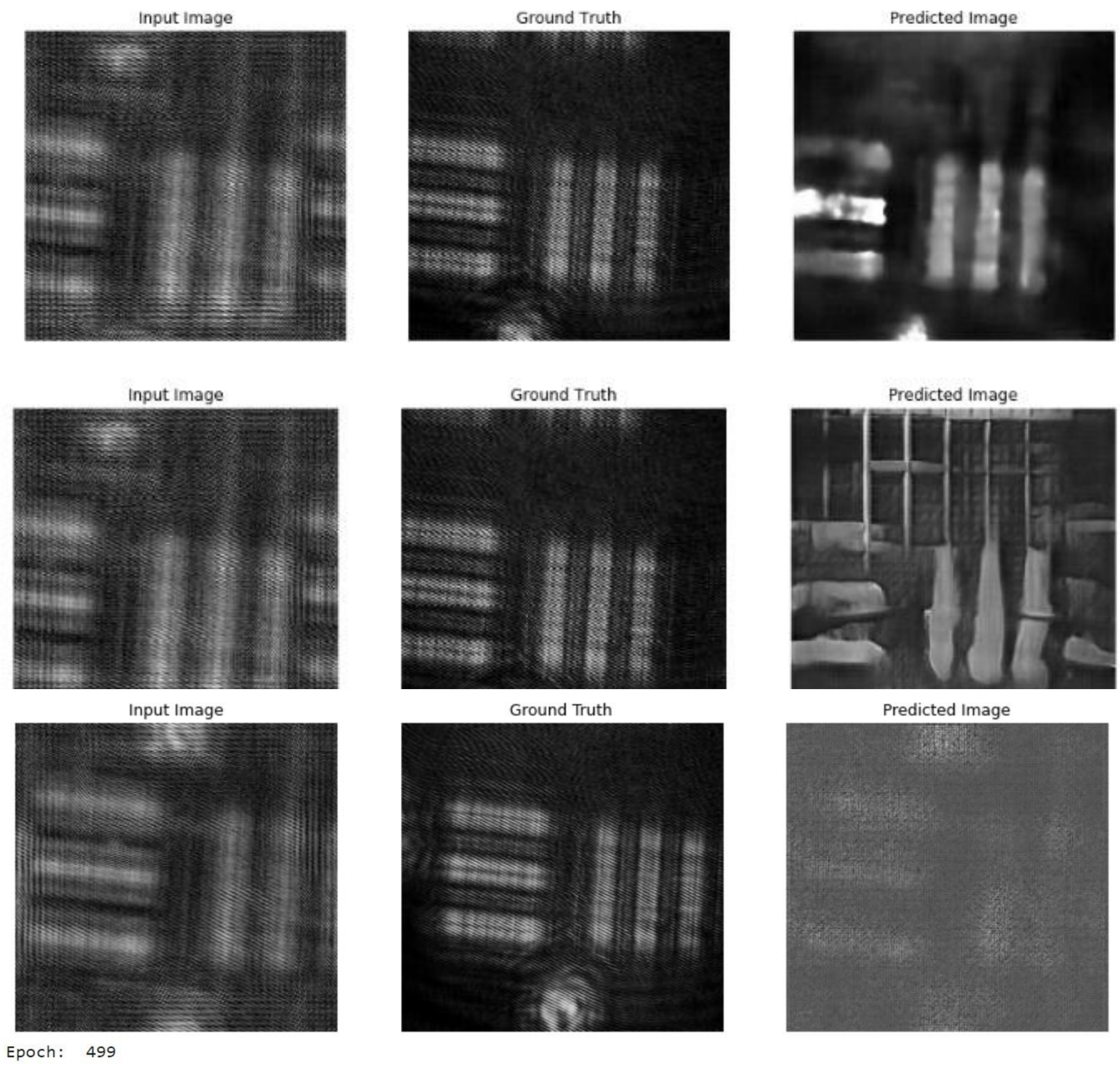


Figure 14. Holograms trained with coco images (a) ResNet (b) ResNet-GAN (c) WRN

Discussion

- **The use of GAN**

Our experiment with five datasets and five networks shows that those network structures with GAN structures yield much better results than those without GAN. We believe that this is because GAN is generally recognized as being particularly useful for image-to-image translations by generating data with the same characteristics as those in the training dataset to improve their learning. GAN improvement is especially evident in the MNIST dataset, where image quality is significantly improved when using the GAN structure. Thus, though it takes twice as long to train with GAN structure, we argue that GAN should still be applied for the image to image translation.

- **Cross-training**

For several cross-training pairs that we tested, the one that yields the best result is when the model is trained with the COCO dataset and tested with the MNIST dataset. As mentioned above, we attribute this to the reason that the training dataset is sufficiently complex for the model to learn the characteristics of intensity patterns and the fresnel diffracted images. Thus when tested on simple images, the result is good.

- **Effect of image complexity on network performance**

We noticed that neural networks trained on datasets of complex images (COCO) usually yield better results when tested on simple datasets (MNIST) during experiments. In contrast, networks trained on simple datasets fail to give satisfactory results on complex datasets.

On choosing datasets for training hologram generating networks, we decided to use complex ones more. We also avoided using a single type of picture, which prevents the neural network from learning features of a particular object pattern rather than potential correlations between image and its hologram.

Dr. Horisaki used randomly generated patterns to train the network in his paper[6]. We also produced random patterns but eventually gave it up because it is not intuitive to see network performance. Dr. Horisaki's random image training dataset finally gives excellent results on testing pictures[6]. This phenomenon arouses our thoughts on the quantification of image complexity. Are the random patterns considered to be more complicated than images in COCO datasets? How about adding random noises to COCO? Will it increase or decrease the complexity? Does the pixel number affect complexity? Is it always good to train neural networks on complex data rather than easy ones, or the best choice is a combination of both?
We do not have answers due to the time limitation. Next might be cross-testing on different datasets and make a graph to show the relationship between image complexity and network performance.

- **Limitations of GAN**

Networks with GAN structures generally yield satisfactory results faster than those with GAN, say after 5~15 epochs. However, as the training process continues, mode collapse happens in GAN prediction. Completely unrelated patterns start to appear in prediction results. Sometimes prediction seems to be made entirely of patterns from other pictures. A situation similar to mode collapse occurs.

Mode collapse might be aroused by the generative nature of GAN. In past examples of GAN, edges of handbags could be transformed into a real handbag picture, and a sketch of facades could be transformed into vivid facades images. Noticeably, GAN changed a simple image to a much more complex one by adding details learned from the training dataset. However, generating holograms does not mean copying any object details from other pictures. A probable method to resolve this is to increase lambda, which is the weight of mean absolute difference in generator loss. Additionally, WGAN[13] and its modified version WGAN-GP[14] also seem to achieve better results and avoid mode collapse. That could be the next few things to try.

It also happens that when training GAN after some epochs (usually over 20), the network performance suddenly turns terrible and predicts nothing, which is very different from CNN, where the network becomes better as training time increases.

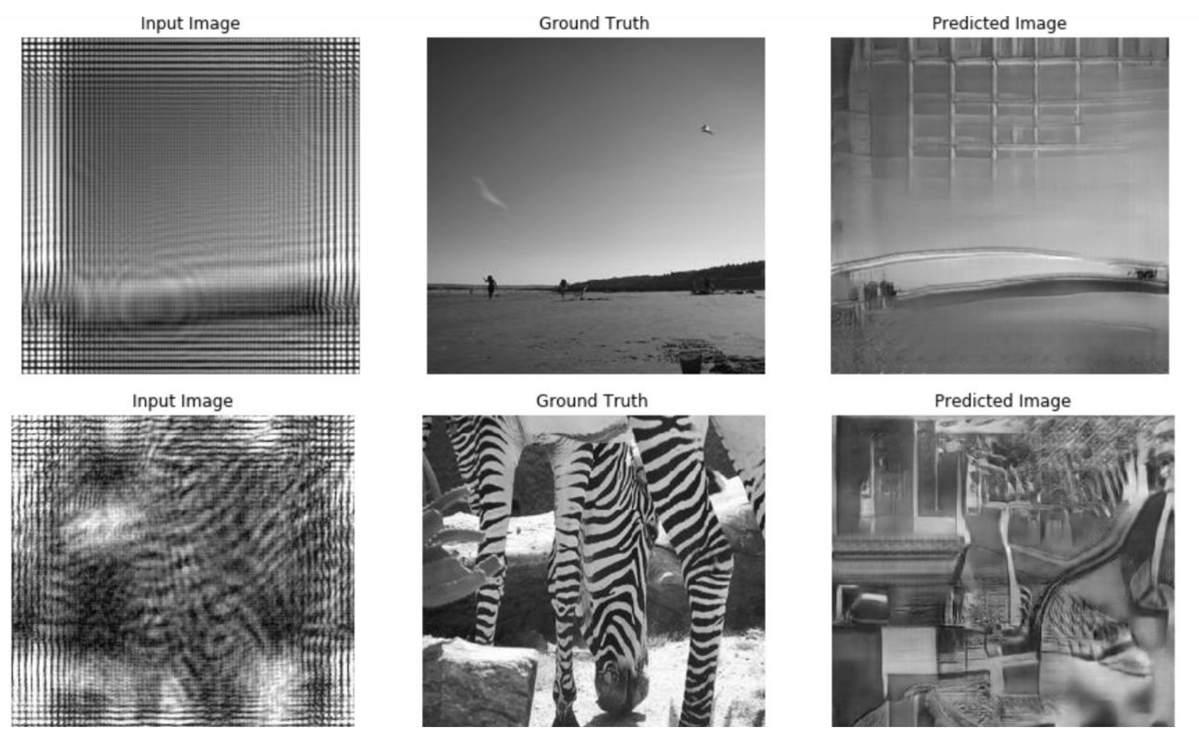


Figure 15. Mode collapse examples, CNN-GAN trained with COCO

Conclusion

In this report, we presented the results that we tested five datasets with ten deep learning models. Based on the paper by Dr. Horisaki, we generated fresnel diffracted images from original grey images and passed them in as the training set. The trained models are then expected to regenerate images from the fresnel diffraction images.

We found that GAN structures work better than those networks without GAN structures. In addition, different networks show different results on different datasets, except MNIST datasets, where all models perform well. However, we argue that this is because the MNIST dataset is relatively easy and straightforward. WRN works the best for random patterns and coco, while WRN and ResNet perform about the same for all other datasets. We propose that the reason is that since WRN is wider but shallower than ResNet, it is more suitable for datasets that are more complex because they can store more parameters. However, since datasets such as Arabic characters, holograms and MNIST are relatively easy, using WRN is discouraged because it takes twice as much time to train WRN while the result is about the same. As for DenseNet, it only gives passable results when training with the simple MNIST dataset. The same applies to SqueezeNet, where it does not yield satisfactory results except for the MNIST dataset and coco dataset.

For future work, we propose taking the network structure one step further. Since we have found and decided on the best network structure for different datasets, the next step is improving one specific network structure for each dataset. Due to time constraints, we did not focus on optimizing parameters. The parameters in all networks are the same for comparison purposes. It is also suggested to improve the structure itself to improve accuracy. Besides, we can take cross-training one step further, since in this work, we only cross-tested three network structures.

References

[1]Slinger, C., Cameron, C. and Stanley, M., 2005. Computer-generated holography as a generic display technology. *Computer*, 38(8), pp.46-53.

[2]Masuda, N., Ito, T., Tanaka, T., Shiraki, A. and Sugie, T., 2006. Computer generated holography using a graphics processing unit. *Optics Express*, 14(2), p.603.

[3]Oscurato, S., Salvatore, M., Borbone, F., Maddalena, P. and Ambrosio, A., 2019. Computer-generated holograms for complex surface reliefs on azopolymer films. *Scientific Reports*, 9(1).

[4]Wang, K., Li, Y., Kemao, Q., Di, J. and Zhao, J., 2019. One-step robust deep learning phase unwrapping. *Optics Express*, 27(10), p.15100.

[5]Ren, Z., Xu, Z. and Lam, E., 2019. End-to-end deep learning framework for digital holographic reconstruction. *Advanced Photonics*, 1(01), p.1.

[6]Horisaki, R., Takagi, R. and Tanida, J., 2018. Deep-learning-generated holography. *Applied Optics*, 57(14), p.3859.

Other references

[7]Sinha, A., Lee, J., Li, S. and Barbastathis, G., 2017. Lensless computational imaging through deep learning. *Optica*, 4(9), p.1117.

[8]Wang, K., Dou, J., Kemao, Q., Di, J. and Zhao, J., 2019. Y-Net: a one-to-two deep learning framework for digital holographic reconstruction. *Optics Letters*, 44(19), p.4765.

[9]Padmanaban, N., Peng, Y. and Wetzstein, G., 2019. Holographic near-eye displays based on overlap-add stereograms. *ACM Transactions on Graphics*, 38(6), pp.1-13.

[10]Rivenson, Y., Zhang, Y., Günaydın, H., Teng, D. and Ozcan, A., 2017. Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light: Science & Applications*, 7(2), pp.17141-17141.

[11]Goi, H., Komuro, K. and Nomura, T., 2020. Deep-learning-based binary hologram. Applied Optics, 59(23), p.7103.

[12]Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W. and Keutzer, K., 2020. Squeezenet: Alexnet-Level Accuracy With 50X Fewer Parameters And <0.5MB Model Size. [online] arXiv.org. Available at: <https://arxiv.org/abs/1602.07360> [Accessed 29 September 2020].

[13]Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv:1701.07875.

[14] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of Wasserstein gans. In Advances in neural information processing systems (pp. 5767-5777).