Application of Deep Learning Techniques to Digital Holographic Microscopy for Numerical Reconstruction

Raghavendra Vijayanagaram [1]

3pc GmbH Neue Kommunikation, Berlin, Germany
rvijayanagaram@3pc.de

Abstract. Digital Holography is an emerging field of a new paradigm for general, as well as microscopic imaging applications. The goal of the project is to conduct experiments for the processing of digital holograms using deep neural networks as an approach to AI-based/semantic large scale and complex information and content analysis. With the usage of deep neural networks, this project develops new techniques for a more robust and performant processing of digital holograms, especially for the task of numerical hologram reconstruction.
We aim to explore the semantic segmentation network models, and challenge to improve the performance of processing holograms. Among the most popular network models, the architecture called U-Net was chosen to achieve this task. In these works, several experiments were conducted applying this method to artificially generated holograms, with promising results. These positive results lead to the believe that a significant improvement could be made on real holographic data, that could lead to achieving a more robust hologram reconstruction.

Keywords: Digital Holography · Deep Neural Network · U-Net.

1 Introduction

1.1 Digital Holography

Hologram refers to an image that is created when the light diffracts when meeting an object. The first practical optical hologram was recorded of a 3-D object. Various types of Holograms, such as transmission, rainbow, reflection holograms have been developed. Digital Holography (DH) was developed as a new tool to record 3D images of objects such as microspheres, cells and other biological specimens. In addition to that, DH was also designed to capture the motion of objects in water and bacterias in cells.

One of the differences between a hologram and a normal image is, that in an image record only the light intensity is captured, while a hologram record captures both intensity and phase of a light field. Holograms contain a 3D image structure of the object. This means that the reconstructed image is visible from different viewpoints. In general, holography is based on the physical properties of light propagation, most importantly the diffraction of light and the interference of light waves.

Digital Holography is a well-known technique to obtain volumetric information about a sample of interest e.g. particles in the solution of water processing. Beside a simpler setup, DH can be performed at video rate, which also imposes other challenges such as the processing of much more data, hence an automatic system is of high-value [1]. To

[1] Master Thesis, March 08, 2019. A Thesis submitted in partial fulfilment of the requirements for the Degree of Master of Information Engineering (MIE) at University of Applied Sciences, Kiel, Germany.

recreate the object, the hologram is recorded digitally and reconstructed. To perform this, the amplitude and phase of the object waves are computed based on the diffraction theory of light. This derivation of the wave front in a particular plane is called numerical reconstruction.

1.2 Project Motivation

For the automatic processing of holograms in DH, two main tasks are of interest: finding the right reconstruction distance and performing the numerical reconstruction for a given hologram. Usually, both tasks are performed separately and may result in very long processing times. In the first phase, it has to be evaluated if these two tasks can be performed by a dedicated deep neural net. If each problem can be solved by a neural net, the idea is to solve both problems with a single pass through a single neural network.

The main goal of the project is:

- Detection of particle in streaming water.
- To explore if deep neural nets can be used to process digital holograms.
- To evaluate if the usage of U-Net will speed-up the process of holography and increase overall performance and accuracy.

We proposed a modified U-net model which can improve the performance of a wide range of images. We showed promising results on both computer-generated and real hologram datasets. Figure 1 shows the vision of the project.

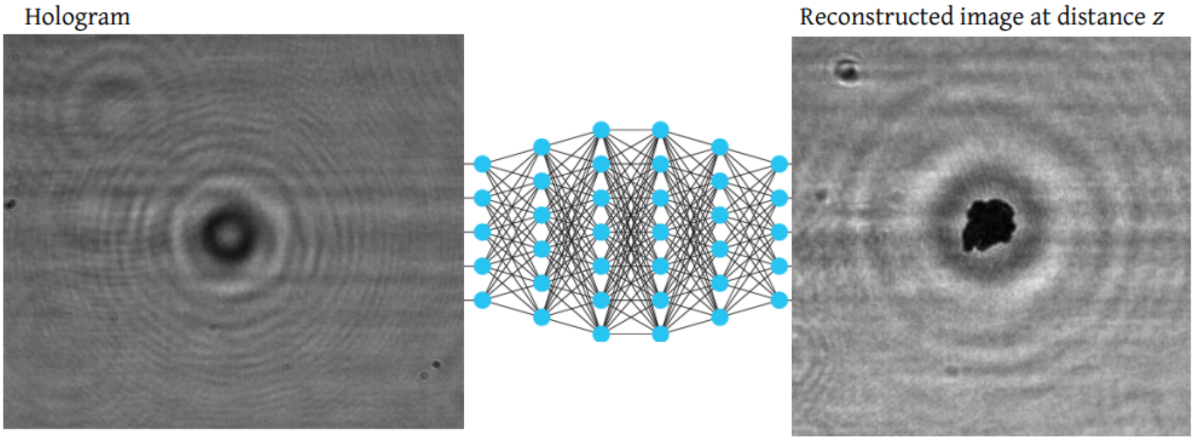


Fig. 1: Project Vision

2 Background

2.1 Loss Functions

In any learning network model, an error of fit is estimated to decide whether the model is a good fit or not. The error is, the difference between ground truth g and the predicted output p.

$$L(x) = \frac{1}{n}\sum_{i=1}^{n}(g_i - p_i) \qquad (1)$$

The function which is used to evaluate error is called loss function L(x). There are different loss functions applied for different types of tasks. Among which the best fitted loss functions for the tasks will be identified. It is a backbone of the algorithm to train the network. The goodness of the model depends on the range of error in the network, which is calculated by loss functions.

In this task, the loss is evaluated from the image pixels, which are compared between ground truth object and predicted the output. Currently, for the proposed network model, which is working on different datasets: artificially generated holograms and real holographic data, there are few interesting loss functions. Those are Cross Entropy, Huber Loss, Mean Squared Error and Mean Pairwise Squared Error.

Cross Entropy: It measures the performance of a model, whose outputs are a probability value between 0 and 1. This loss function is also called log loss or sigmoid cross entropy because it applies with the sigmoid activation function. The formulation of loss is defined as:

$$C = -\frac{1}{n}\sum_{x}[glog(p) + (1-g)log(1-p)] \tag{2}$$

where n = total number of items, x = sum of training inputs, g = ground truth, p = prediction.

Huber Loss: It is used in robust regression problems. It inherits the properties from two popular different loss functions: Mean Absolute Error (MAE) or L1 loss and Mean Square Error (MSE) or L2 loss [2]. For minimizing the loss, if the error is small then the loss inherits the properties of MSE or if its high it takes the MAE. This especially leads to robustness to outliers when the estimated data is very noisy. The Huber loss is defined as follows:

$$L(a) = \begin{cases} \frac{1}{2}(g-p)^2 & if|(g-p)| < \delta \\ \delta(g-p) - \frac{1}{2}\delta & \text{Otherwise} \end{cases} \tag{3}$$

where g = Ground Truth, p = Prediction output.

Mean Pairwise Squared Error(MPSE): MPSE is similar to Mean Square Error (MSE) [2]. It measure the difference between pairs of corresponding elements predictions and ground truth values. The formulation of MPSE is

$$L(a) = \frac{1}{n}\sum_{i=1}^{n}|p(x_i) - g(x_i)|^2 \tag{4}$$

Where n is the number of pixels in the ground truth, p(x) is the prediction output, g(x) is the ground truth.

2.2 U-Net Architecture

U-Net stands for Unity Networking. U-Net is one of the most popular end-to-end Autoencoder networks for semantic segmentation. It was developed and first used by R. Fischer for biomedical image segmentation at the Computer Science Department of the University of Freiburg, Germany [3]. Semantic Segmentation is the partition of an image into different regions. For example, classifying each pixel that belongs to a person, place, object or any entity in the dataset. Auto-encoder is a type of neural network which is used to learn efficient data encoding.

The main goal of this network is to learn to compress the data from the input layer, into a short code manner and then to compress it into a size that is similar to the original image used as input. Encoder gradually decreases the spatial dimensions of the input image, and the decoder gradually recovers the object information and its spatial dimensions of the image. For this decoding, the recovered objects information, have shortcut connections. It is purely based on traditional convolutional neural networks. In this network, the main advantage is when upsampling will be, to learn deeper and concatenating the feature resolutions from the down part.

3 State of the Art

In the book, *Digital Holography* Pascal Picart and Jun-chang Li state the fundamentals of holography [4]. They explain the process of digital holography and wave propagation theory. A hologram contains encoded 3-D information of sample particles, which are obtained with the help of interference of light. A drawback of this approach is to decode an image from holograms, phase recovery and amplitude, these steps are necessary for the reconstruction process. Furthermore, the distance of the object from the sensor is evaluated. Standard methods to recalculate the image with the distance generate high amounts of images. These images need to be analyzed and a decision regarding which of the objects will be reconstructed needs to be made. This whole process is very time-consuming.

In the article *Practical algorithms for simulation and reconstruction of digital in-line holograms* Tatiana Latzchevskaia and Hans-Werner Fink [5] describe the methods for simulation and reconstruction of in-line holograms, which are recorded with the plane and spherical waves. In holograms specifically, the optimal parameters related to distances, sampling rates, and other major factors, help to reconstruct the holograms, which are easily evaluated. They showed some results based on numerical procedures which are helpful for the reconstruction of recorded holograms. In our research work, we observed the numerical procedures to reconstruct the recorded holograms on our task.

In his article, Yair Riverson et.al.[6] implemented an approach for phase recovery and holographic image reconstruction using deep neural nets. This article demonstrates the application of neural networks for learning phase recovery, with holographic images after training. A Convolutional Neural Networks (CNN) performs the auto-focusing sample object image and extends the depth of field (DOF) method to get the image reconstruction. In the first step of this auto-focusing process, the neural network can train on the data. The setup of holograms is specified by particular distances of the image set-up environment. As a result, the image reconstruction quality is bad. Later deep learning-based holographic image reconstruction method performs the auto-focusing and image phase recovery, using the sample hologram intensity. This approach is called *Holographic Imaging using Deep Learning for extended Focus (HIDEF).*

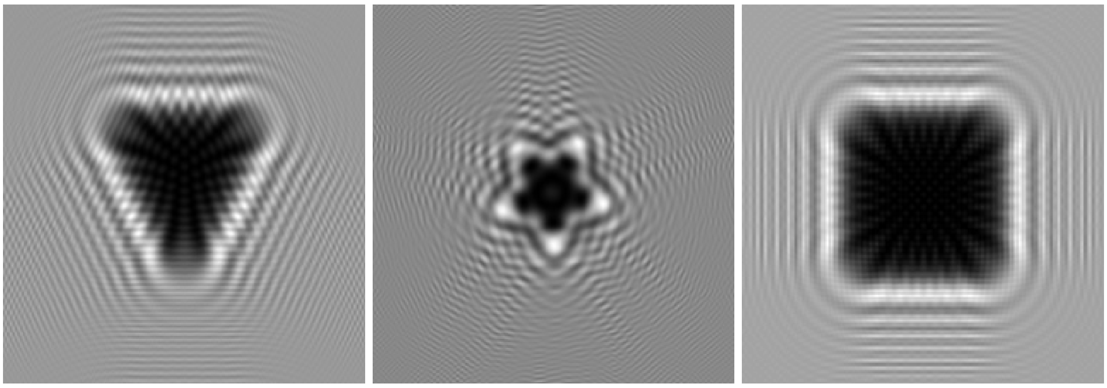
Olaf Ronneberger and his team at the University of Freiburg implemented a new approach especially applicable for reconstructing the medical images of objects which are smaller than 50 micro millimeters [3]. They proposed a new method called a slidingwindow convolutional network for semantic segmentation problems. U-Net is based on convolutional architecture for fast and precise segmentation of images. The proposed architecture contains the two different paths: contracting path that is used for capturing the context and expanding path that enables the precise localization of the object. This semantic segmentation network is very useful for our hologram reconstruction task.

4 Experiments

4.1 Description of Dataset

Data preparation is an important step in any learning network model. The quality of input data strongly influences the results which are produced by the model.

As part of the baseline System, we used holograms that are artificially generated by a computer. These holograms are generated for three simple object shapes rectangle, triangle, and star. Each hologram was generated with varying rotations and object sizes and for a broad range of propagation distances. The figure 2 shows the generated hologram samples.



(a) Hologram of a Triangle (b) Hologram of a Star (c) Hologram of a Rectangle

Fig. 2: Generated Holograms of Triangle, Star, Rectangle.

After successfully training the baseline systems, We used two different datasets diffraction and real diffraction datasets for training real holograms. Both datasets contain generated holograms, however, they are based on real recorded holograms. The two reasons for using generated holograms are: it was not possible to record and especially annotate a huge amount of holograms, and data was needed which was much closer to real recorded holograms for conducting experiments.

4.2 More Realistic Data

The diffraction dataset was used to train the model in the first experiments. As mentioned above, this dataset consists of generated holograms based on recordings of real holograms. The process starts with the input of a given real hologram, cropped to a fixed region of 256x256 pixels. Afterward, a reconstruction step is performed with the help of a standard reconstruction algorithm so-called Fresnel algorithm and a binary mask is applied to the reconstructed image. The binary mask marks are region containing the object of interest. Using the binary mask, the object is cropped which is done to remove all the background noise hence then left out only with object shapes. Afterward, various geometrical transformations are performed to the object e.g. rotations, translations. The transformed input is fed into the hologram generator which will generate holograms for lots of different propagation distances. This process results in a large dataset of hologram images and their respective ground truths. The total process is depicted in the figure 3.

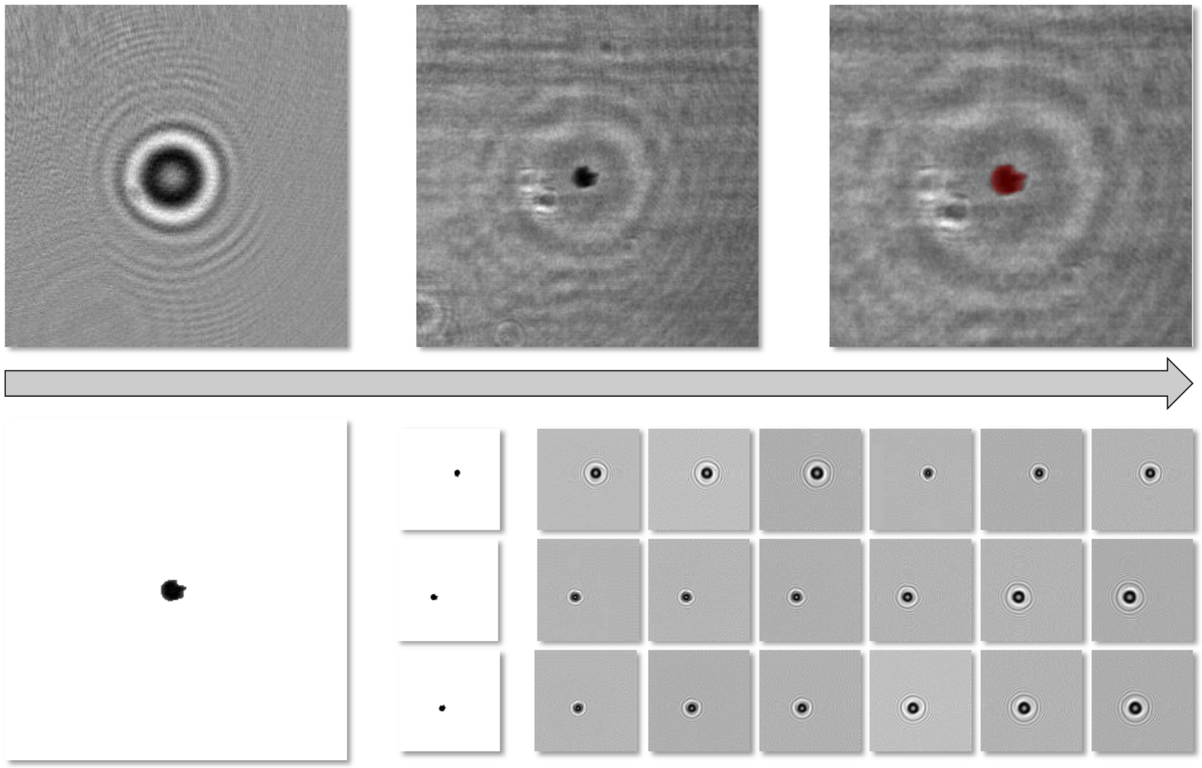


Fig. 3: Data Generating Process

From the first set of experiments gave us good results on the training set, the test set as well as the blind test sets. The results of real holograms were still not satisfying. That is why we implemented different strategies to solve this: we applied different levels of noise to the dataset to see if this helps to generalize on real holograms and to improve the hologram generation of even more realistic holograms. The holograms are similar to the previous dataset with an important difference: did not remove the background noise after the reconstruction step, hence then the generated holograms will contain interference patterns from several noise sources like interference from other particles, sensor noise or dust particles in the background.

This limits the number of image transformations which could be applied and therefore reduces the amount of generated data. The ground truth of this set still contains the target objects without the noise background. It means the goal is to try to learn a model which is implicitly removes the noise during the hologram reconstruction. This dataset consists of 30100 hologram images. A few sample images are illustrated below in 4.

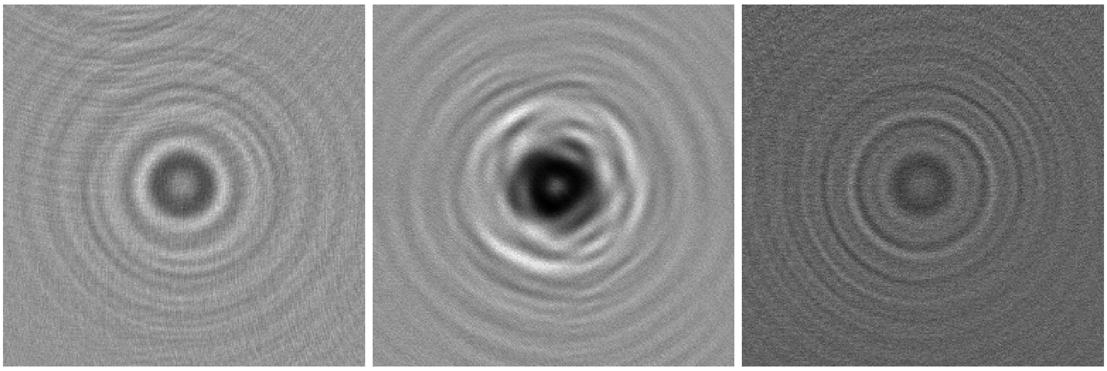


Fig. 4: Real Diffraction dataset example holograms

4.3 Evaluation Datasets

The trained models were evaluated using several evaluation sets. The sets represent different levels of difficulty. The list of evaluation dataset are as follows:

- **Training Set -** This is the simplest set which includes the training sets, depending on the experiment. This set will always be similar to the training set.
- **Test set -** This is the normal set from the training corpus, this is as usual a subset of the training set, which was not used for the model training or weight updates but only for testing. This set does not contain the same images as found in training, they will be still very similar because small steps in the propagation distance used during the hologram generation will lead to the very small difference from one image to the next.
- **Blind test -** This set contains the same training datasets but with different transformations e.g. rotations which were not part of the training. This means, still the same objects as used in the training model but the objects have different rotation.
- **Blind test of another dataset -** These sets are even more difficult, it contains holograms of objects that are not part of the training corpus e.g. if the model was trained on the diffraction corpus, then the evaluation was done on real diffraction set. They also contain different translations of objects.
- **Real holograms blind test -** The most difficult evaluation data for the network will always be the real holograms test set. It contains the unaltered holograms as they are recorded by the in-line holography system. It is also important to note that this subset does not contain any holograms of the particles which were used for hologram generation.

5 Implementation

5.1 Building Input Pipeline

The first part of the experiment is to prepare the holograms for the input pipeline. Handling this large amount of data is quite a challenging job. In general, the baseline system, we used so far loading the artificial holograms and preprocessing the holograms was done with external libraries and then into the model with the feed dict method.

Tensorflow introduced the new version of Dataset API in an easy way to create a simple record-oriented format which is popular for handling large data called TFRecords [7].

TFRecord is nothing but a binary file format or TensorFlow file. In a layman definition, when to train a deep network, the input data has two ways to feed the network model. Loading the data with pure naive code and feed it into the computational graph which we used in our baseline system. The other way is to use an input pipeline that takes the list of filenames, shuffles them, creates a file queue and decodes the data. This can be done with a TFRecord which was used in our current model. The advantage of using this type of file format is, it takes less space on disk, takes less time to copy and can be read from disk much more efficiently.

5.2 Model Definition

Figure 5 illustrates the current model architecture which is used. We conducted all the experiments with this model as default and in between updated the hyperparameters.

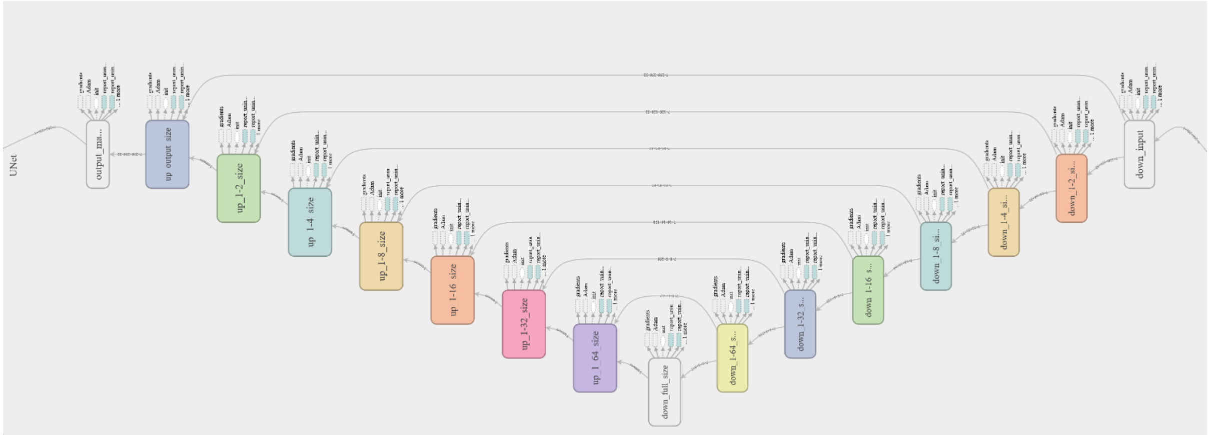


Fig. 5: U-Net Model Architecture

The experiments start with preparing the input pipeline for passing the data into the model. In the above figure, the right-hand side colored boxes represent the downsampling or contracting path and left-hand side colored boxes are upsampling or extracting path. These blocks are joined together with grey arrows marks are addressed by skip connections. The skip-connection is a technique that creates a shortcut to ignore the unnecessary next layers. The feature map is passed through the two types of connections: the continuous connection and the skip connection or shortcut. The adjacently previous layers as continuous connections and the connection that skips the layers are skip-connections. There is a ReLU activation function and batch normalization are succeeding the above each convolutional layer.

The feature maps are afterward fed to the next block which is similar to the first block. The second block convolves and down-sampled from 32 feature maps with the size of 128x128 feature maps with the size of 64x64. Furthermore, the feature maps fed to the five consecutive blocks with the same configuration of parameters, which are 256 kernels with the size of 3x3. The output of the seventh down-sampling block which is the last

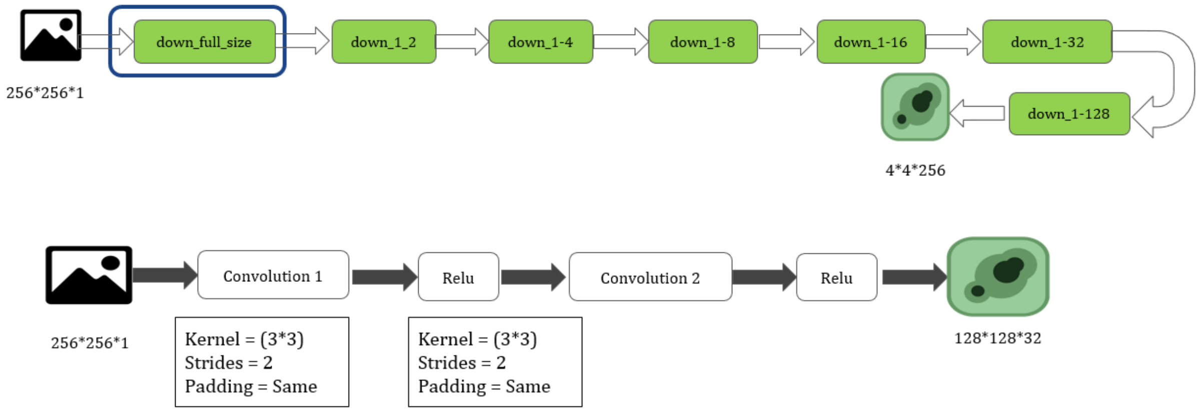


Fig. 6: Schematic representation of downsampling process

block of this part is 256 feature maps with a size of 4x4. The schematic representation of downsampling was shown in figure 6.

In the Up-sampling or extraction path, the Eighth block convolves and up-sampled from 256 feature maps of the 4x4 size to 256 feature maps with the size of 8x8. This block of layer contains the transposed convolution layer with 256 kernels with the size of 3x3 slicing 2 pixels for each step. The decoding block of layers contains the concatenation operation with the feature maps from the preceding layers through skip connections. After that, it combined feature maps followed by 3 regular convolutional layers with ReLU as activation function, batch normalization, and dropout. The schematic representation of upsampling was shown in the figure 7.

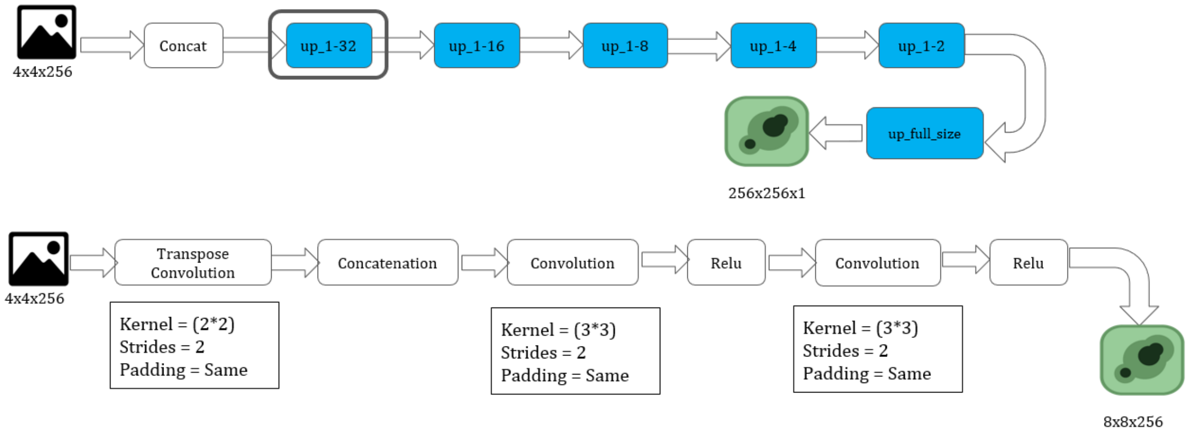


Fig. 7: Schematic representation of upsmapling process

Furthermore, the feature maps are fed to the five consecutive Up-sampling blocks with the same configuration of parameters, which are 32 kernels with a size of 3x3. The output of the final block is a single feature map with the size of 256x256.

6 Results and Analysis

The model was trained on difficult datasets using an NVIDIA 12 GB Graphical Processing Unit. The default network settings are as follow:

- Used Optimizer Adam
- A learning rate of 0.0001
- image size of 256x256 pixels
- Used training data is shuffled randomly

6.1 Base Line Results

The U-Net model was trained by Adam optimizer with an initial learning rate of 0.001. The model was learned with the number of epochs is 200 with the size of 64 training holograms. The criterion of evaluation is Sigmoid cross entropy.

The best results come from holograms with very small distance 0.0001 millimeter, while the worst outputs were generated by a large distance of 0.0600 millimeter. Different rotations didn't affect the result. The loss curve of the training process is depicted in figure 8. The loss curve decreases over the training epochs via Sigmoid Cross Entropy Loss function.

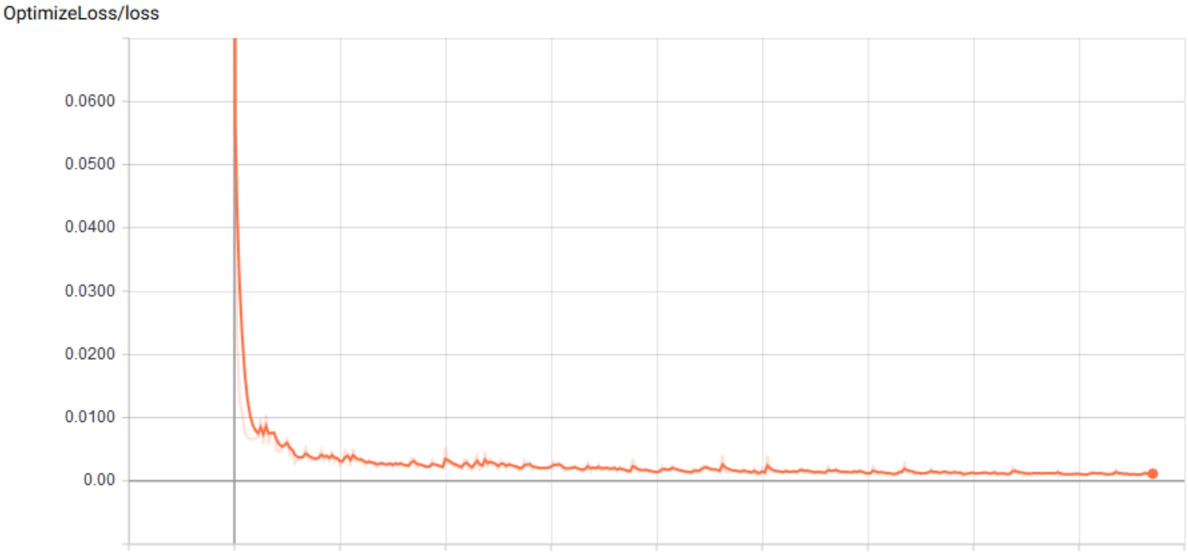


Fig. 8: Loss curve

Figure 9 illustrated the heat-map of U-Net with three shapes of objects. In the figure, the two curves of the rectangle and triangle were decreasing when the distance is increased, except the star object curve. The loss curve of star increased by small distances, but nearly leveled off in the large distances.

Based on these results, we achieved the state of art deep neural methods, with promising results. These positive results let us believe that a significant improvement could be made of real holographic data.

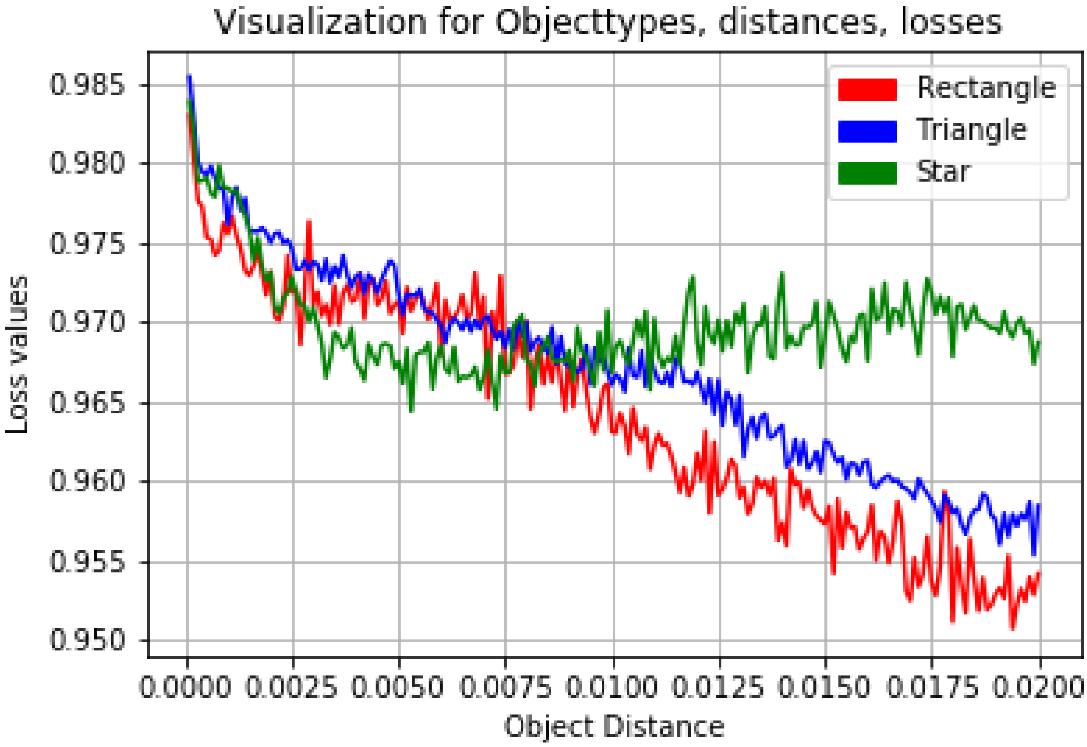


Fig. 9: Comparing with objects loss curve

Exp. Id	Loss Function	Activation Function	Experimental Observation
1	Sigmoid Cross Entropy	Linear Function	The training and predicted results are blurred
2	Sigmoid Cross Entropy	Linear Function with the subset of dataset	Images are blank
3	Mean Squared Error	Linear Function with the subset of dataset	Training images are getting better but evaluation images are still blank
4	Mean Squared Error	Linear	Trained images are better with full set, still validation images are not good

Table 1: List of Conducted Experiments

The first experiments were trained using the baseline architecture without changing any parameters. The idea was, to see if the baseline model can be using the real holographic dataset. The diffraction dataset, which continues to generated holograms based on real objects. After the training, the predicted results are blurred images. The reason for this might increase the complexity of data variations, which network was not able to learn. In the table 1 the list of each experiment were conducted with different hyperparameters and the results are not expected.

6.2 Improving generalization on unknown data

After the model could be applied to the generated dataset, the next experiments aimed at improving performance on more difficult data, like rotation or object shapes not part of the training set. In other words, the goal was to test the generalization capabilities of the model. Therefore, in addition to the default network training settings different loss functions: Mean Pairwise Squared Error, Huber Loss and Mean Squared Error in combination with the activation functions: linear and sigmoid were tested. The maximum training time for each experiment was 48 hours. The experiments which were conducted during the period time were shown in the table 2, the first column represents the experiment id, next columns are used loss function, data preprocessing methods, different datasets, output activation functions and results of each experiment.

Exp. Id	**Loss Function**	**Activation + Data Augmentation**	**Experimental Observations**
5	Mean Pairwise Squared Error	Sigmoid Activation	The training corpus curve is decreasing but evaluation curve in tensorboard fluctuates.
6	Huber Loss	Sigmoid Activation	The Reconstructed images are white background. The training was erroneous.
7	Mean Pairwise Squared Error	Sigmoid Activation	It shows better performance but the real hologram set images are predicted white circle images instead of circle objects
8	Mean Pairwise Squared Error	Sigmoid Activation + Added Noise	Both training and evaluation curves are decreasing. But some images are slightly predicted erroneous
9	Mean Pairwise Squared Error	Sigmoid Activation + Added Noise+ Random Standard deviation	The trained model was predicted well and showing good results

Table 2: List of Conducted Experiments with Diffraction.

7 Interesting Observations

7.1 Using more realistic training data

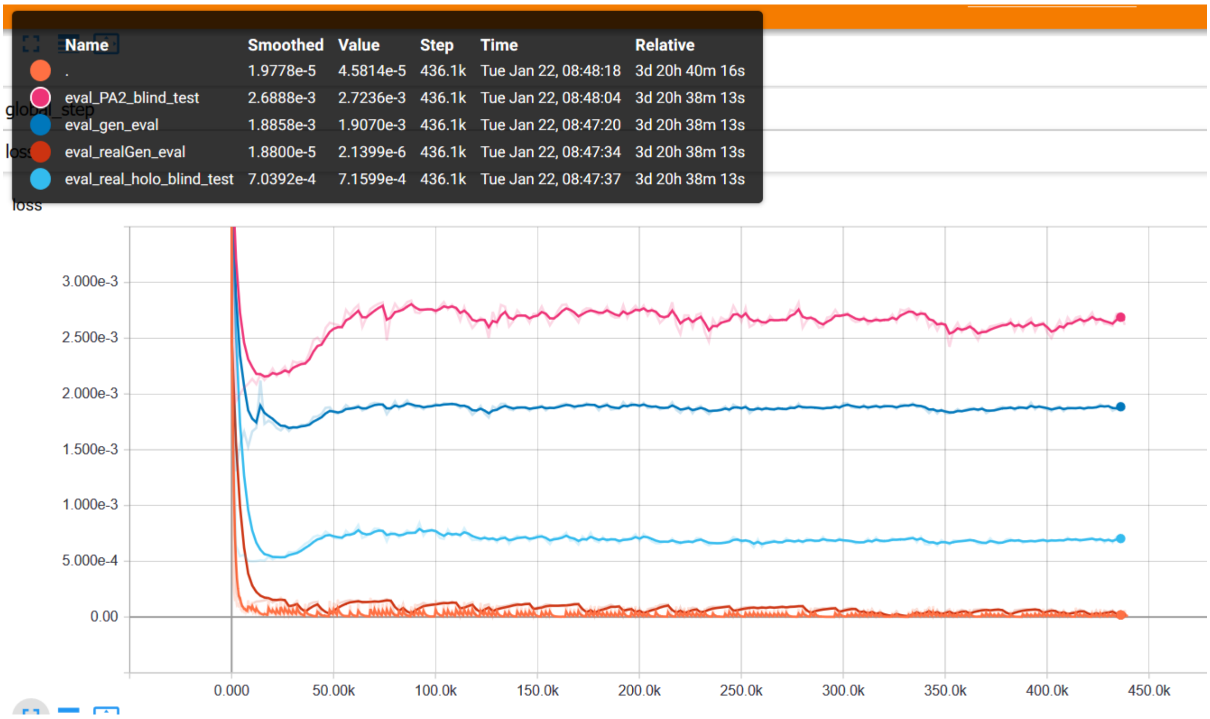
The previous experiment showed a good performance of the model on generated data, for different levels of difficulty. However, the model did not improve substantially for real holograms. Therefore, in the final experiments, the real generated hologram dataset was used, which resembles a more realistic holograms. The goal was to check if this kind of data will help improving on real holograms. For this again different combinations of loss and activation functions were used. The list of conducted experiments with this dataset is shown in table 3.

Experiment Id	Loss Function	Activation + Data Augmentation
10	Mean Pairwise Squared Error	Sigmoid Activation + (Added Noise + Random deviation + Image Rotation)
11	Mean Pairwise Squared Error	Sigmoid Activation + (Without Noise + Image Rotation)
12	Huber Loss	Sigmoid Activation + (Added Noise + Random deviation + Image Rotation)

Table 3: List of Conducted Experiments with Real Diffraction.

7.2 Exp.Id- 10 Quantitative Analysis:

The Analysis of this experiment was impressed. The results were improved in terms of both training and evaluation. The loss curve is depicted in the figure 10. During the training corpus, the curve is slightly fluctuating but in a decreasing order. Even the evaluation test which is a different level of difficulty levels set results were shown the good to compare previous steps and leveled off with the training corpus curve. The blind test set of real holograms was showed better than previous and the curve goes in decreasing point but in times the curve rose when the steps are large.

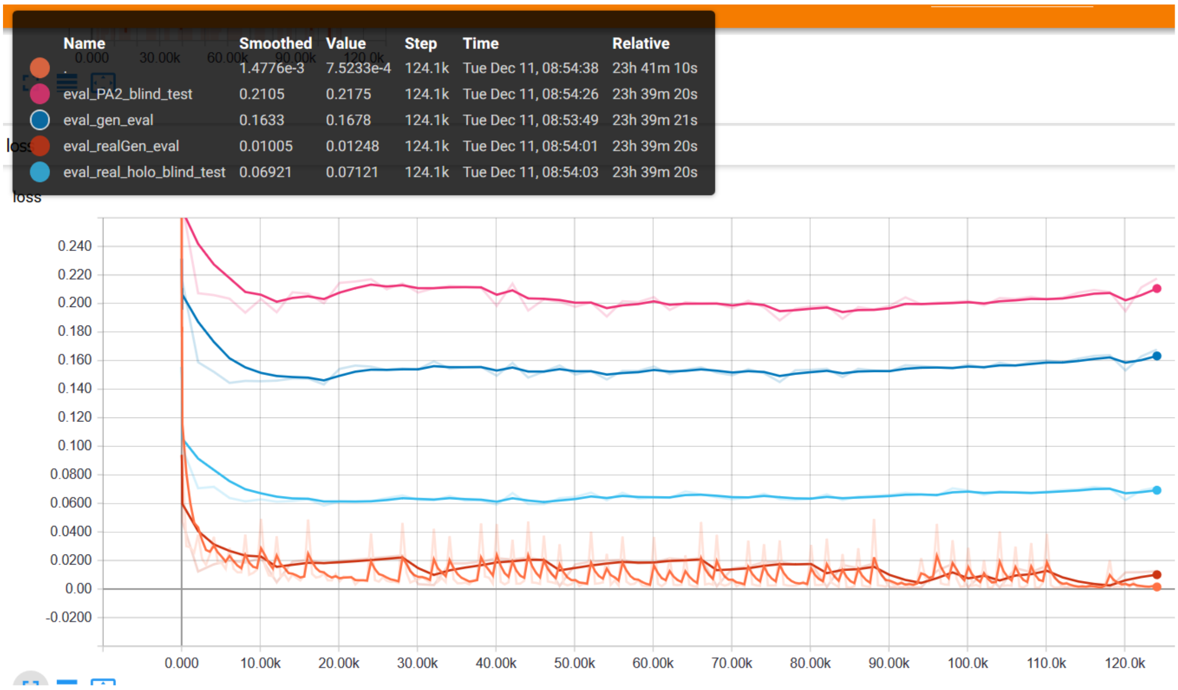


	Name	Smoothed	Value	Step	Time	Relative
●	.	1.9778e-5	4.5814e-5	436.1k	Tue Jan 22, 08:48:18	3d 20h 40m 16s
●	eval_PA2_blind_test	2.6888e-3	2.7236e-3	436.1k	Tue Jan 22, 08:48:04	3d 20h 38m 13s
●	eval_gen_eval	1.8858e-3	1.9070e-3	436.1k	Tue Jan 22, 08:47:20	3d 20h 38m 13s
●	eval_realGen_eval	1.8800e-5	2.1399e-6	436.1k	Tue Jan 22, 08:47:34	3d 20h 38m 13s
●	eval_real_holo_blind_test	7.0392e-4	7.1599e-4	436.1k	Tue Jan 22, 08:47:37	3d 20h 38m 13s

Fig. 10: Experiment Id:10 Training Loss

7.3 Exp.Id- 11 Quantitative Analysis:

The evaluation loss curve was leveled off with different blind datasets including the real holograms test set, in a small range of values. The loss curve decreases over the training steps were shown in figure 11. In the loss curves, the light blue color curve is the real hologram blind test set. It showed the best results among the previous experiments. It clearly showed that the experiment was a great success.



	Name	Smoothed	Value	Step	Time	Relative
●	.	1.4776e-3	7.5233e-4	124.1k	Tue Dec 11, 08:54:38	23h 41m 10s
●	eval_PA2_blind_test	0.2105	0.2175	124.1k	Tue Dec 11, 08:54:26	23h 39m 20s
○	eval_gen_eval	0.1633	0.1678	124.1k	Tue Dec 11, 08:53:49	23h 39m 21s
●	eval_realGen_eval	0.01005	0.01248	124.1k	Tue Dec 11, 08:54:01	23h 39m 20s
●	eval_real_holo_blind_test	0.06921	0.07121	124.1k	Tue Dec 11, 08:54:03	23h 39m 20s

Fig. 11: Experiment Id:11 Training Loss

7.4 Exp.Id- 12 Quantitative Analysis:

During the training corpus and evaluation, blind test sets were predicted great and decreasing enormously. In previous experiments the similar configuration without noise and augmentation operation was used, it predicts the blank images. But for this, it predicts good images due to the reason for applied noise and augmentation. We used this because to avoid the overfitting problem. This quite an interesting finding for this experiment. The evaluation loss curves were showed similar to the previous one but not accurate. The curve fluctuates when it reaches the large steps and predicts the overfitting.

8 Conclusion

The proposed network architecture U-Net has proved to be a suitable approach for reconstructing real recorded holograms. The network design is highly adaptable for solving the image translation problems, especially recorded holograms, regardless of ground truth. During the experimental process, different loss functions were used because the architecture had faced problems with reconstructing the ground truth images, particularly for real holograms. In the end, the U-Net has shown the best performance using the Mean Pairwise Squared Error Loss Function combined with the Sigmoid Activation Functions.

In previous work, U-Net was successfully applied with several medical images, but not on generated images of holographic microscopy. We applied this network to reconstruct objects from real holograms and showed this is an additional way to utilize the benefits of architecture. Furthermore, we modified the network by adding data augmentation and changing hyperparameters, which leads to an increase in its performance. The updated network was evaluated for several difficulty levels of different test sets, to test the generalization capability. The entire holographic reconstruction process from the U-Net architecture was dependent on the four major steps outlined in this experimental setup: datasets, input pipeline, training and evaluating observations.

References

[1] Myung K. Kim. "Principles and Techniques of digital holographic Microscopy". In: *CoRR* 21 (2011). `http://faculty.cas.usf.edu/mkkim/papers.pdf`, pp. 85–91.

[2] "Visualization of some loss functions for Deep Learning with Tensorflow". In: (2018 (Accessed: February, 2019) @Online). https://mc.ai/visualization-of-some-loss-functions-for-deep-learning-with-tensorflow/.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). `https://arxiv.org/pdf/1505.04597.pdf`.

[4] Pascal Picart and Jun-chang Li. *Digital Holography.* ISTE Ltd and John Wiley Sons, Inc., 2012. ISBN: 978-1-84821-344-9.

[5] Tatiana Latychevskaia, Loı, and Hans-Werner Fink. "Practical algorithms for simulation and reconstruction of digital in-line holograms". In: *CoRR* abs/1412.23674v7 (2016). `https://arxiv.org/pdf/1412.3674.pdf`.

[6] Yair Rivenson et al. "Phase recovery and holographic image reconstruction using deep learning in neural networks". In: *CoRR* abs/1705.04286 (2017). `http://arxiv.org/abs/1705.04286`.

[7] Daniil Pakhomov. "Tfrecords Guide". In: (2016 (Accessed: February, 2019)@Online). `http://warmspringwinds.github.io/tensorflow/tf-slim/2016/12/21/tfrecords-guide`.