PR-DAD: Phase Retrieval Using Deep Auto-Decoders

Leon Gugel and Shai Dekel,School of mathematical sciences, Tel-Aviv university

***Abstract*—Phase retrieval is a well known ill-posed inverse problem where one tries to recover images given only the magnitude values of their Fourier transform as input. In recent years, new algorithms based on deep learning have been proposed, providing breakthrough results that surpass the results of the classical methods. In this work we provide a novel deep learning architecture PR-DAD (Phase Retrieval Using Deep Auto-Decoders), whose components are carefully designed based on mathematical modeling of the phase retrieval problem. The architecture provides experimental results that surpass all current results.**

***Index Terms*—Phase retrieval, sparse representation, deep learning.**

I. Introduction

A. The Phase Retrieval problem and classical methods

The two-dimensional discrete Fourier transform $\mathcal{F}(x)$ of an image $x \in \mathbb{R}^{n\times n}$, can be represented by the magnitude

$$\omega(x) := |\mathcal{F}(x)| \in \mathbb{R}^{n\times n},$$

and the phase

$$\varphi(x) := \arg \mathcal{F}(x) \in [-\pi, \pi]^{n\times n},$$

where $\arg M$ denotes the argument of a complex matrix M applied element-wise. The Fourier phase retrieval is a famous ill-posed inverse problem where the goal is to recover x, or equivalently the phase $\varphi(x)$, using only a input the magnitude $\omega(x)$. The difficulty of the problem stems from the fact that the phase contains most of the information of the image. This problem arises in many areas in engineering and science and has a rich history tracing back to 1952 [14]. Important examples for Fourier phase retrieval naturally appear in many optical settings since optical sensors, such as a charge-coupled device (CCD) and the human eye, are insensitive to phase information of the light wave. A typical example is coherent diffraction imaging (CDI) which is used in a variety of imaging techniques (see [1] and references therein). In CDI, an object is illuminated with a coherent electro-magnetic wave and the far-field intensity diffraction pattern is measured. This pattern is proportional to the object's Fourier transform and therefore the measured data is proportional to its Fourier magnitude. Phase retrieval also played a key role in the development of the DNA double helix model [7]. Additional examples for applications in which Fourier phase retrieval appear are X-ray crystallography, speech recognition, blind channel estimation, astronomy, computational biology, alignment and blind deconvolution (see [1] and references therein).

The classical techniques for phase retrieval are iterative methods such as the alternating projection (see the survey [1]). The general scheme of the alternating projection at each step k is

(i) compute the Fourier transform $\mathcal{F}(x_k)$ of the current estimated image x_k,
(ii) keep its phase information $\varphi(x_k)$, and replace the magnitude by the known ground truth magnitude $\omega(x_k) = \omega(x)$,
(iii) compute the inverse Fourier to obtain a temporary estimate $\tilde{x}_{k+1}$,
(iv) impose certain known constraints, if needed, on $\tilde{x}_{k+1}$ (e.g. real non-negative pixel values), to obtain x_{k+1}.

The PhaseCut method [17] is based on the following minimization formulation for the the input modulus ω, unknown image $x = \{x_{j,k}\}$ with unknown phase $\varphi = \{\varphi_{j,k}\}$

$$\min_{x,\varphi} \|\mathcal{F}(x) - \omega \cdot \varphi\|^2, \quad \text{s.t } |\varphi_{j,k}| = 1, \forall j, k.$$

There are several ways to relax this formulation and derive from it a minimization problem in the phase only, especially if x is known to be real.

B. The learning setup

When we apply learning methods to an inverse problem such as phase retrieval, we need to clarify if we are attempting to solve the problem in the supervised, semi-supervised or unsupervised setting. First observe that one can easily compute the Fourier magnitude values for any ground truth image and therefore such pairs can be used for supervised training.

- **Supervised:** In this case we provide the trained model access to pairs of Fourier magnitude inputs and their corresponding ground truth images. Using these pairs, one can apply a loss function such as Mean Square Error (MSE) between the predicted and ground truth images that will drive the minimization of a gradient descent method during the training of the model.
- **Semi-supervised:** In this setting only a partial subset of the Fourier magnitude inputs has corresponding ground truth images. This may happen in cases where we have acquired the Fourier magnitude of data through an imaging process, but we do not have knowledge about the ground truth image, except perhaps for the fact that it in a certain given image class with certain characteristics. This typically implies that to use the Fourier magnitude inputs which have no matching ground truth pixels during the training process, one needs to add additional loss mechanisms. One such loss function is the cycle loss which computes the Fourier magnitude of the images

generated by the model and then compares them with the input Fourier magnitudes. Another loss is the adversarial loss where a discriminator network is trained to provide a prediction if the image generated from the Fourier magnitude is plausible, i.e. if it belongs to the given class of images.

- **Un-supervised:** Here, we work with a dataset that has only Fourier magnitude inputs with no ground truth images at all. In this case we can only use loss functions such as the cycle loss to drive the training of our model. One can not use the adversarial loss since there are no ground truth images that can be used as reference for the discriminator. However, if there exists some general prior knowledge on the structure of the given class of images, such as sparse Gaussian blobs, one can potentially transfer this knowledge into the form of a regularization loss function on the model's output images during training.

In this work we assume that we are in the supervised or semi-supervised regimes, where we have a sufficient amount of training image samples from the given class. To achieve this, in some cases, we augment the given ground truth images by applying certain carefully designed transformations, thereby enriching the training set. One can also envision enriching the training dataset by creating synthetic data that faithfully represents the given class. For example, in the setting of x-ray crystallography one can generate synthetic virtual molecules from which one can compute pixel image slices. In this work, our fundamental assumption is that an inverse method based on learning can truly outperform the classical methods on a given class of images, if that class has sufficient structure and the learning algorithm can be trained to 'understand' that structure.

C. Overview of Recent Deep Learning based methods

We now review some recent work where deep learning methods are applied to the problem of phase retrieval.

The DeepPhaseCut architecture [4] starts with a modified U-net generator $\mathcal{G}_\Theta$ that takes as input the Fourier magnitude and predicts the Fourier phase. We note in passing that applying a convolutional network, such as a U-net, on a frequency representation, is perhaps not optimal, since there are typically no spatial correlations between 'neighboring' Fourier coefficients or their respective magnitude. The predicted phase is then multiplied by the given magnitude to give a predicted Fourier transform. Then, an inverse Fourier transform is applied to obtain a predicted intermediate image. The intermediate image is then fed into an enhancement network $\mathcal{H}_\Psi$ to obtain the predicted image. The network is trained using several losses such as a cycle consistency loss, where the Fourier magnitude is extracted from the predicted image and compared to the input magnitude. The authors also trained discriminator classification networks that provide a score relating to the belief that an image is a ground truth image or an image produced by the generator network from Fourier magnitude data. This allows to use adversarial loss during training. Lastly, although the DeepPhaseCut is somewhat equipped with a adversarial networks and a combination of adversarial and cycle loss function to deal with the unsupervised phase retrieval problem, it also employs during the training process a supervised cycle loss component between ground truth images and the predicted images.

In [16], similar concepts were used, namely, a generator was trained to take as input the Fourier magnitude and output a predicted image. The generator was trained with a linear combination of conditional and advesarial losses. Here as well, a discriminator was trained simultaneously to provide the advesarial loss. The authors of [16] note that a generator architecture based on fully connected layers provides better empirical results than a convolutional architecture, which aligns with our understanding.

In [15] the authors propose to use a Cascaded Phase Retrieval (CPR) neural network (NN) architecture consisting of a sequence of sub-networks $G^{(1)}, \ldots, G^{(q)}$. Each subnetwork $G^{(i+1)}$ is fed as input the known magnitude $\omega(x)$ and $\hat{x}^{(i)} \in \mathbb{R}^{n_i \times n_i}$, an estimate of the image at some given (lower) resolution which is the output of the subnet $G^{(i)}$. The last subnet $G^{(q)}$ predicts the image x at the full resolution. The CPR network is trained with a loss function that incorporates all of the elements of the sequence of multiresolution approximations $\{\hat{x}^{(i)}\}$.

II. The Auto-Encoder/Decoder Network

As already stated, in this work we assume that we have a sufficient amount of training image samples from the given class which allows us to first learn some aspects of the structure of the class during a preprocessing stage. To be more specific, for each given class we design or train a representation space in which the images of the class have a sparse representation. For piecewise constant images such as the MNIST dataset [11], one can use the Haar wavelet orthonormal basis representation. For the fashion-MNIST dataset [19], whose images also have some textured regions, we use the Haar wavelet packet basis representation (see Subsection II-A below). For real-life image classes such the celebA [12], we train carefully designed auto-encoder/decoder DL architectures (see Subsection II-B below).

Once we find a good encoder-decoder pair that provides sparse representation for the image class, we extract the decoder part and plug it into our phase retrieval inference network. The central idea of our phase retrieval architecture is to use the prior knowledge about the encoded sparse structure of the class and ensure the network first maps the input Fourier magnitude to this space. Once this representation is obtained by the first part of the network, it undergoes enhancement and then is auto-decoded by the pre-trained decoder to provide the output approximate image. It is crucial to observe that it is the existence of the pre-trained decoder component that 'forces' the network to transform the input Fourier magnitude to the desired encoded form. As we explained, there are two options to obtain an encoder-decoder pair that is adapted to the image class: using a carefully selected fixed transform or training a neural network architecture:

A. Encoding using a fixed transform

In some cases, one can simply select a certain transformation and its inverse as the encoder-decoder pair. This is especially useful in the semi-supervised or unsupervised settings where we do not have enough ground truth images to train an encoder-decoder network. There are three main properties that such a choice should satisfy:

(i) Sparsity - The images from the given class should be sparse in the given transformed representation. Let $T(x) = \{\alpha_k(x)\}_k$ be a transformation of an input image x into a coefficient representation. Then a popular choice for a sparsity measure is requiring the l_1 norm $||T(x)||_1 = \sum_k |\alpha_k(x)|$, to be minimal.

(ii) Automatic differentiation of the inverse transform - The implementation of the inverse transform T^{-1} needs to be plugged into a neural network, such as the PR-DAD network, as a sub-network and undergo backpropogation during the training of the rest of the network.

(iii) Stability of the inverse transform - The inverse transform T^{-1} should satisfy some stability condition, such as the frame condition

$$A||\alpha||_2^2 \leq ||T^{-1}\alpha||_2^2 \leq B||\alpha||_2^2, \quad \forall \alpha = \{\alpha_k\}_k, \quad (1)$$

where $0 < A \leq B < \infty$. This ensures stability of the backpropogation process during training.

Let us now provide some concrete examples for such transformations. An image class such as the MNIST dataset [11] of small grayscale hand-written digits is a prototype example for a class of piecewise constant functions. It is well know that the Haar wavelet representation [5] provides a sparse representation for such images. Its simple implementation supports automatic differentiation and in this special case of an orthonormal transformation one may select $A = B = 1$ in (1).

The fashion-MNIST dataset [19] consists of small grayscale images of clothes and accessories such as t-shirts, trousers, hand bags and shoes (some samples are depicted in Figure 1). Some of the items, such as the t-shirts contain some texture components. An orthonormal transform that provides better sparsity for such data is the Haar wavelet packet transform depicted in Figure 2 (see e.g. [10] for texture classification using the wavelet packet transform). Compared to the more basic Haar transform, the packet transform further decomposes the subbands of the basic transform to time-frequency elements, which better capture the local texture patches, i.e. allow a sparser representation. With the Haar packet transform a given image is decomposed into 4 subband blocks. First we filter each row using the low-pass and high-pass filters

$$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right), \qquad \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right),$$

and down-sample by a factor of 2. For an image of dimension $2^n \times 2^n$, this process gives a low-pass block and high-pass block, each of dimension $2^{n-1} \times 2^n$. Next we filter each column in the same manner, thereby obtaining 4 blocks, each of dimension $2^{n-1} \times 2^{n-1}$. The 4 blocks are sometimes labeled by: LL,LH,HL and HH, where L=Low and H=high. In the packet transform these blocks are recursively filtered and subdivided, where the last blocks that are decomposed are of dimension 2×2.

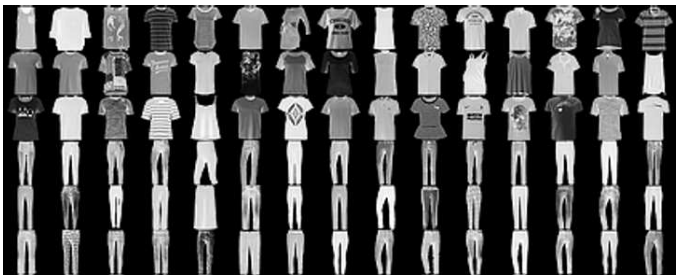

Fig. 1. Samples from the Fashion MNIST dataset

Fig. 2. Haar Wavelet Packet transform (from [9])

B. Training an encode-decoder pair

A robust alternative to pre-selection of a fixed encoder-decoder pair, is to train such a pair, with the goal of learning a set of nonlinear projections of the images from the given class onto a sparse representation space. Here, we review one such useful architecture where the encoding space has a structure of over-redundant low-resolution components. For a class of images of dimensions 32×32, the encoding space is composed of 64 or 128 feature maps, each of dimension 8×8. In such a case, although the dimension of the encoding space can be larger than the image dimensions, the goal is that only a small portion of the encoding neurons have significant activations for a given image. In Figure 3 we see a depiction of the encoder-decoder architecture. The encoder part consists of 3 'DownConv' blocks. The 'DownConv' block consists of 2 convolutional layers, each with convolutions of size 3×3, stride$= 1$, batch normalization and the ReLU or PReLU nonlinear activations. Each 'DownConv' block concludes with an average pooling operation of 2×2. The decoder subnet that recovers an image from the sparse representation has an almost symmetric architecture. The first 3 blocks are upscaling 'UpConv' blocks, where each block is composed of an upsampling bilinear interpolation operator, followed by 2 convolution layers identical to the encoder convolution layers. The final decoder block consists of 1 convolutional layer whose output is the decoded image. Figure 4 depicts the training loss function used for training the encoder-decoder architecture. It combines the l_1 sparsity regularization of the encoded representation space with the Mean Squared Error (MSE) loss between the input images to the encoder and the outputs of the decoder.

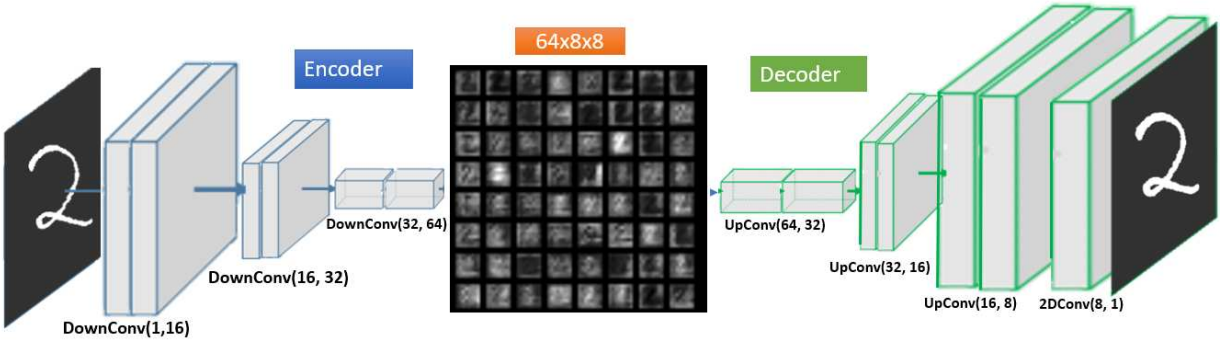

Fig. 3. Encoder-decoder architecture: the case of low resolution dictionary

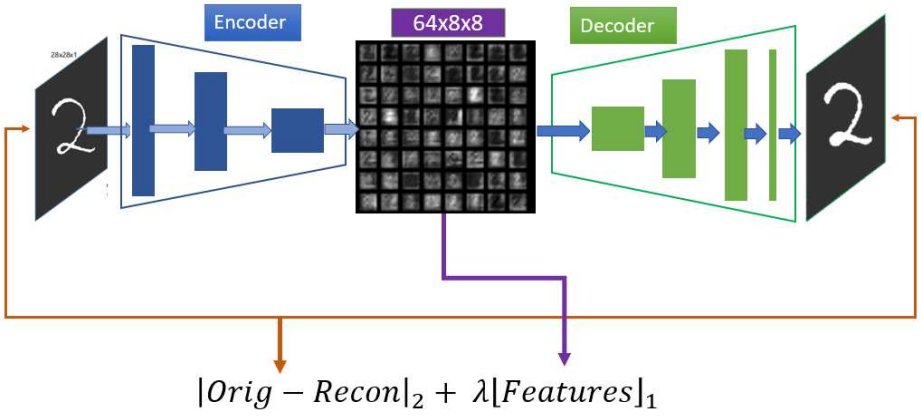

Fig. 4. Encoder-decoder: training loss function

III. The PR-DAD architecture

Figure 5 depicts the components of our PR-DAD architecture. It compromises of 3 subnets:

(i) The Fourier magnitude to encoder representation subnet - this subnet is designed to take as input the Fourier magnitude, which is frequency related data and convert it to the pre-selected or pre-trained representation space.
(ii) The encoder representation enhancement subnet - the role of this subnet is to enhance the encoded representation before it is fed into the decoder.
(iii) The decoder - this is the pre-selected or pre-trained decoder component that is plugged into the PR-DAD architecture. It is crucial to understand that it is the decoder that drives the training process, in the sense that the two subnets leading to it need to provide the decoder with a good approximation of the representation of the ground truth image, so as to minimize the training loss (see Subsection III-D for the training loss functions).

Observe that we do not attempt to reconstruct the actual phase at any stage of the network. Indeed, in our experiments, we found out that attempting to directly recover the phase so as to combine it with the known magnitude degrades the performance and is less stable then predicting a sparse representation from the magnitude, from which in turn one recovers the image using the decoder.

A. The Fourier Magnitude to Encoder Representation Subnet

The goal of this subnet is to predict from the Fourier magnitude input, the encoder representation of the predicted image. In general, there is no immediate spatial correlation between neighboring Fourier coefficient values in the frequency domain. Therefore, in contrast to some previous work, we prefer to process the input Fourier magnitude data using a relatively shallow Multi-Layer-Percepton (MLP) architecture, over a potentially deeper architecture of convolutional layers. In the MLP architecture, each layer contains a full affine transformation where any input value may contribute to any output value. In all our experiments we use an MLP consisting of 4 layers. It is important to point out that the nonlinear activation function we use is the Parametric ReLu (PReLu), given by

$$\sigma_a(z) := \begin{cases} z, z > 0, \\ az, z \leq 0, \end{cases}$$

where at each layer, the coefficient a is a parameter of the network. The reason we do not use the ReLU activation function

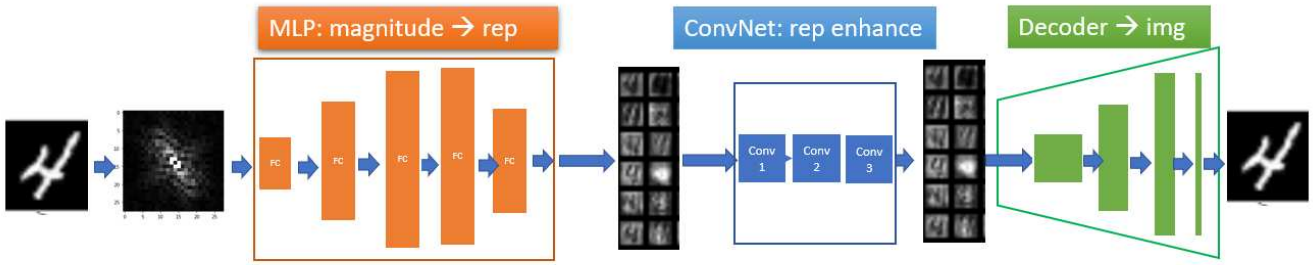

Fig. 5. PR-DDL architecture

with the fixed parameter $a = 0$, is that we need to ensure that the MLP subnet is consistent with the encoder representation which may require negative values of representation neurons. For example, in the case where the encoder is selected to be the Haar transform, the representations are real wavelet coefficients with potentially negative values.

The dimensions of the output of the last layer of the MLP are set to the dimensions of the auto-encoder features. For example, if we use a set of 128 auto-encoder feature maps, each of dimension 8×8 pixels, then the output of the last MLP layer is then of dimension $128 \times 8 \times 8$.

It is interesting to note that initially we tried to use a fixed inverse Fourier or inverse Discrete Cosine Transform (DCT) component that will take in the output of the MLP subnet, apply to it the inverse transform and then pass forward the data to the encoding representation layer. Such a component is used for example in the DeepPhaseCut algorithm [4] on output of a phase generator subnet after it is combined with the input magnitude. Our initial idea was that the MLP subnet will learn the Fourier transform of the encoding representation to which one should apply the inverse Fourier transform. However, in our architecture, since any linear transformation such as the inverse Fourier can be combined into the final affine transformation followed by a PReLU activation taking place in a MLP subnet, we found through ablation experimentation that the inverse Fourier component is not needed in our architecture, as it is in some sense already 'realized' by the MLP subnet.

B. The Encoding Representation Enhancement Subnet

This is an optional subnet of the PR-DAD architecture (see Figure 5). It assumes that the previous MLP subnet succeeded to convert the Fourier magnitude data to the encoding representation and its goal is to provide some capacity for the purpose of enhancing the representation, before it is passed to the decoder.

In the case where the encoding representation space was created by a convolutional encoder, the enhancement subnet is also convolutional. Let us assume that the representation space is composed of N feature maps, each of dimension $n \times n$ (e.g. $N = 128$, $n = 8$). Then, the input and output of each layer of the enhancement network are N channels/feature maps of dimension $n \times n$. We apply filters of $X - Y$ dimension 3×3, which implies each filter is of dimension $N \times 3 \times 3$. This determines that each feature map is enhanced using information from all other feature maps. Our implementation in the experimental results below uses 3 convolutional blocks, each consisting of 2 convolutional layers, each with batch normalization and PReLU activation.

C. The Decoder Subnet

This subnet is initially a fixed component of the network, whose architecture is the decoder part of the auto-encoder/decoder network that was pre-selected or trained during the preprocessing stage described in Section II. In the case where the decoder is trained, this subnet can use the fixed weights that were computed during the encoder-decoder training process. It is also possible to allow this subnet to participate in the training of the full PR-DAD network during the last few epochs, following a 'transfer learning' paradigm. Through experimentation, we found that this could slightly improve the performance in some cases.

D. The PR-DAD Training Loss Functions

Figure 6 depicts all of the loss function components used for the training of the PR-DAD architecture, where the actual loss function is a weighted sum of all of them. The following loss functions are applied for each training batch $B = \{x_i\}_{i \in B}$

(i) MSE loss of predicted images $\{\hat{x}_i\}_{i \in B}$, invariant to rotation by π

$$L_{MSE} = \frac{1}{\#B} \sum_{i \in B} \min(||x_i - \hat{x}_i||_2^2, ||x_i - rotate_\pi(\hat{x}_i)||_2^2).$$

(ii) Cycle loss with predicted magnitude

$$L_{mag} = \frac{1}{\#B} \sum_{i \in B} ||\omega(x_i) - \omega(\hat{x}_i)||_2^2.$$

(iii) Sparsity of predicted encoding representations $\{\hat{T}_i\}_{i \in B}$

$$L_{sparse} = \frac{1}{\#B} \sum_{i \in B} ||\hat{T}_i||_1.$$

(iv) MSE loss of predicted encoding representations $\{\hat{T}_i\}_{i \in B}$, invariant to rotation by π

$$L_{encode} = \frac{1}{\#B} \sum_{i \in B} \min(||T_i - \hat{T}_i||_2^2, ||T_i - rotate_\pi(\hat{T}_i)||_2^2).$$

Observe that the rotation of the representation space is an operation depending on the encoding method. For example, in the case of a representation by N 'low resolution' elements of dimensions $n \times n$, the rotation operation is applied separately on each element.

Then, the training loss is a weighted sum of all of the above losses

$$L = \lambda_{MSE}L_{MSE}+\lambda_{mag}L_{mag}+\lambda_{sparse}L_{sparse}+\lambda_{encode}L_{encode}$$

IV. Experimental Results

A. Overview of Datasets

For the experimental evaluation we used four "MNIST" datasets, each consisting of grayscale images of dimension 28×28:

(i) MNIST [11] - 70,000 images of hand written digits from 10 classes $1-10$,
(ii) EMNIST [3] - The balanced version of 131,600 images containing hand written letters and digits from 47 classes,
(iii) Fashion-MNIST [19] - 70,000 images of clothing from 10 classes such as: T-shirt, Trouser, Pullover, etc.,
(iv) KMNIST [2] - 70,000 images of 10 types of handwritten Japanese characters.

The 5th dataset we used is the more challenging CelebA dataset [12], which consists of 200,000 images of human faces (20 images of 10,000 different individuals in diverse poses and setting). The datasets underwent certain pre-processing for two purposes: We applied exactly the same cropping and resizing as in previous work, so we can compare the results (see tables below). We enriched the training sets using certain transformations. Table I summarizes all of the transformations for each given dataset.

B. Results in the Supervised Setting

The implementation of the PR-DAD algorithm can be found on the GitHub [8]. For the implementation of the Haar wavelet packet as an encoder-decoder option, we used some parts of the code from [9]. The training of both encoder-decoder and the PR-DAD architectures were done using a V100 Tesla GPU. For the training we used batches of 32-64. We applied the Adam stochastic gradient decent algorithm using the loss functions detailed in Subsection III-D

The metrics used for evaluation are on par with the previous work. Given two images $x, \hat{x}$ of size N we have:

(i) MSE loss - Mean Squared Error $\frac{1}{N}\sum_{i,j}(x_{i,j}-\hat{x}_{i,j})^2$, lower is better.
(ii) MAE loss - Mean Average Error $\frac{1}{N}\sum_{i,j}|x_{i,j}-\hat{x}_{i,j}|$, lower is better.
(iii) SSIM - Structural Similarity, higher is better.
(iv) PSNR - Peak to Signal Noise Ratio $10\log_{10}\left(\frac{255^2}{MSE}\right)$, higher is better.

In Tables II-VI we see the test results on the MNIST, EMNIST, KMNIST, Fashion-MNIST and CelebA datasets. In Figure 7 we see some samples of pairs of original and recovered cropped celebA images. It is very evident that on the more challenging dataset such as the celebA dataset, the PR-DAD demonstrates superior performance.

TABLE II
QUANTITATIVE COMPARISON ON THE MNIST DATASET

Model	MSE	MAE	SSIM	PSNR
PRCGAN [16]	0.0168	0.0399	0.8449	-
CPR [15]	0.0123	**0.037**	0.8756	-
PR-DAD Haar Packet	0.0106	0.0381	**0.8815**	39.4861
PR-DAD auto encoder-decoder	**0.0100**	0.0398	0.8799	**40.0208**

TABLE III
QUANTITATIVE COMPARISON ON THE EMNIST DATASET

Model	MSE	MAE	SSIM	PSNR
PRCGAN [16]	0.0239	0.0601	0.8082	-
CPR [15]	0.0144	0.0501	0.8700	-
PR-DAD Haar Packet	0.0119	0.0475	0.8710	38.4744
PR-DAD auto encoder-decoder	**0.0108**	**0.0422**	**0.8879**	**39.2972**

V. Conclusions

In this paper we presented a deep learning approach to the phase retrieval problem. The PR-DAD algorithm uses an encoder-decoder transform or network that provides a sparse representation of images from a given class. This facilitates solving the phase retrieval problem by an architecture that predicts from the Fourier magnitude data the image, without trying to predict the actual phase. We showed that our solution provides experimental results that are highly competitive.

In the future we plan to expand the capabilities of the algorithm and apply it on microscopy and crystallography datasets. Such datasets will potentially require different encoder-decoder architectures. We also plan to try and develop capabilities in a semi-supervised setting, where there exists only a small amount of ground truth images. This makes training an encoder-decoder architecture more difficult.

References

[1] T. Bendory, R. Beinert and Y. Eldar, Fourier Phase Retrieval: Uniqueness and Algorithms, In: Compressed Sensing and its Applications (2017), 55-91.
[2] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto and D. Ha, Deep learning for classical Japanese literature, arXiv:1812.01718 (2018).

TABLE IV
QUANTITATIVE COMPARISON ON THE KMNIST DATASET

Model	MSE	MAE	SSIM	PSNR
PRCGAN [16]	0.0651	0.1166	0.5711	-
CPR [15]	0.0433	0.1034	0.6624	-
PR-DAD Haar Packet	0.0383	0.1027	0.6365	28.3249
PR-DAD auto encoder-decoder	**0.0380**	**0.0957**	**0.6605**	**28.4031**

TABLE V
QUANTITATIVE COMPARISON ON THE FASHION-MNIST DATASET

Model	MSE	MAE	SSIM	PSNR
PRCGAN [16]	0.0151	0.0572	0.7749	-
CPR [15]	0.0113	0.0497	0.8092	-
PR-DAD Haar Packet	**0.0078**	0.0471	0.8186	**42.1862**
PR-DAD auto encoder-decoder	0.0081	**0.0442**	**0.8242**	41.811

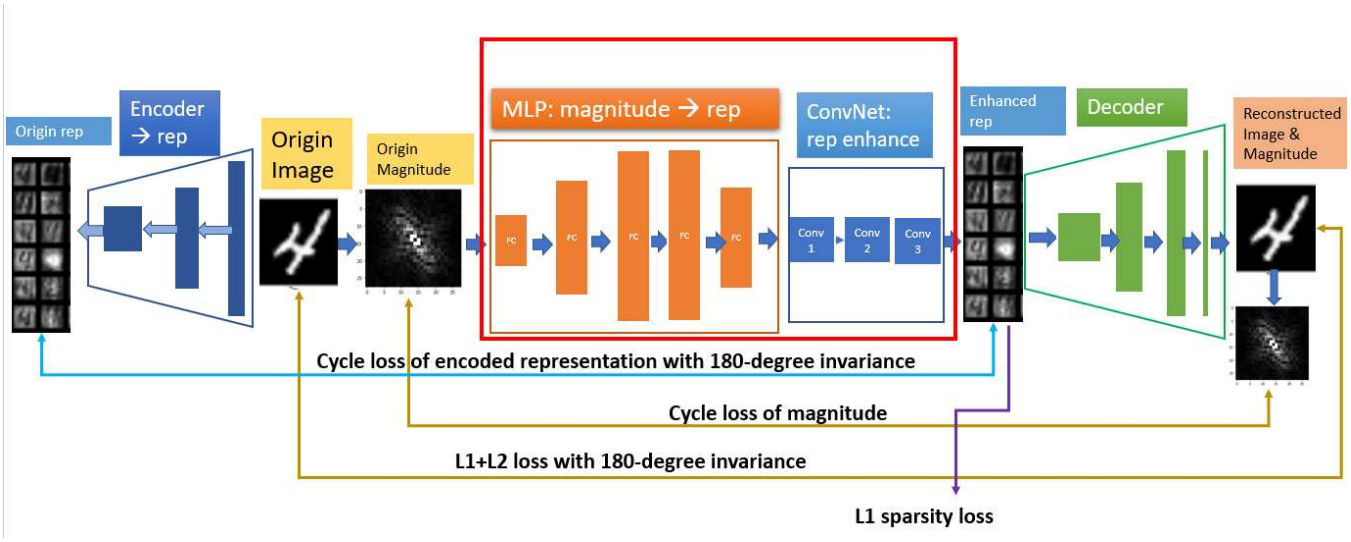

Fig. 6. Loss functions used during training

TABLE I
DATASET PRE-PROCESSING AND AUGMENTATION.

Transform	**MNIST**	**EMNSIT**	**KMNIST**	**Fashion Mnist**	**CelebA**
Resizing	32×32	32×32	32×32	32×32	64×64
Center Cropping	No	No	No	No	Yes
Normalization (μ, σ)	$(0.1307, 0.3081)$	$(0.1307, 0.3081)$	$(0.1307, 0.3081)$	$(0.1307, 0.3081)$	$(0.5, 0.5)$
Fourier Magnitude zero padding	0.5	0.5	0.5	0.25	No
Bernoulli probability ρ	0.25	0.25	0.50	0.25	0.50
Random Horizontal Flipping with probability ρ	No	No	No	No	Yes
Random Free Rotation with probability ρ in range $(\theta, -\theta)$	No	No	No	$(1.0, 2.5)$	No
Random Free Translation with probability ρ in range τ_1, τ_2	$(0.025, 0.025)$	$(0.025, 0.025)$	$(0.025, 0.025)$	$(0.0125, 0.025)$	$(0.025, 0.025)$
Random Free Scaling with probability ρ in range (r_1, r_2)	$(0.9, 1.2)$	$(0.9, 1.2)$	$(0.9, 1.2)$	$(0.95, 1.1)$	$(0.9, 1.2)$
Random Gaussian Blur, prob ρ, kernel k with σ	No	No	0.0	No	$(0.5, 1.5)$
Random Gamma Correction, prob ρ in range (γ_1, γ_2)	No	No	0.0	No	$(0.85, 1.125)$

Fig. 7. Top - cropped CelebA original images, bottom - cropped celebA recovered images

TABLE VI
QUANTITATIVE COMPARISON ON THE CROPPED CELEBA 64×64 DATASET

Model	MSE	MAE	SSIM	PSNR
PRCGAN [16]	0.0138	0.0804	0.6779	n/a
HIO [6]	n/a	n/a	0.472	19.573
PhaseCut [17]	n/a	n/a	0.7600	25.3600
On-RED [18]	n/a	n/a	0.4940	19.7960
PrDeep [13]	n/a	n/a	0.7380	26.0579
DeepPhaseCut [4]	n/a	n/a	0.8540	27.1190
PR-DAD	**0.0025**	**0.0340**	**0.8815**	**51.9661**

[3] G. Cohen, S. Afsher, J. Tapson and A. Schaik, EMNIST: Extending MNIST to handwritten letters, IEEE international joint conference on neural networks 2017.
[4] E. Cha, C. Lee, M. Jang and J Ye, DeepPhaseCut: deep relaxation in phase for unsupervised Fourier phase retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear.
[5] I. Daubechies, Ten lectures on wavelets, SIAM 1992.
[6] J. R. Fienup, Phase retrieval algorithms: a comparison, Applied optics 21 (1982), 27582769.
[7] L. Garwin and T. Lincoln, A century of nature: twenty-one discoveries that changed science and the world, University of Chicago Press, 2010.
[8] L. Gugel, PR-DAD code, https://github.com/gugas81/pr-dad.
[9] H. Huang, R. He, Z. Sun, T. Tan, Wavelet-SRNet: A Wavelet-Based CNN for Multi-Scale Face Super Resolution, Proc. IEEE international Conference on Computer Vision (ICCV), 2017, 1689-1697.
[10] A. Laine and J. Fan, Texture classification by wavelet packet signatures, IEEE Transactions on pattern analysis and machine intelligence 15 (1993), 1186-1191.
[11] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998), 2278-2324.
[12] Z. Liu, P. Luo, X. Wang and X. Tang, Deep learning face attributes in the wild, Proceedings of the IEEE international conference on computer vision (2015), 3730-3738.
[13] C. Metzler, P. Schniter, A. Veeraraghavan and R. Baraniuk, prDeep: Robust phase retrieval with a flexible deep network, ICML 2018, 35013510.
[14] D. Sayre, Some implications of a theorem due to Shannon. Acta Crystallographica, 5 (1952), 843843.
[15] T. Uelwer, T. Hoffmann and S. Harmeling, Non-iterative phase retrieval with cascaded neural networks, ICANN 2021.
[16] T. Uelwer, A. Oberstra and S. Harmeling, Phase retrieval using conditional generative adversarial networks, ICPR 2021.
[17] I. Waldspurger, A. dAspremont and S. Mallat, Phase recovery, maxcut and complex semidefinite programming, Mathematical Programming 149 (2015), 4781.
[18] Z. Wu, Y. Sun, J. Liu, and U. Kamilov, Online regularization by denoising with applications to phase retrieval, Proc. IEEE International Conference on Computer Vision Workshops 2019, pp. 00.
[19] H. Xiao,K. Rasul and R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, arXiv:1708.07747 (2017).

VI. BIOGRAPHY SECTION

Shai Dekel Shai is a visiting associate professor at the school of mathematical sciences, Tel-Aviv university.

Leon Gugel Leon is a senior deep learning expert at D-ID