On the Use of Prior Knowledge in Deep Learning Algorithms

by

Kwabena K. Arthur
S.B., Massachusetts Institute of Technology (2017)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author .....

Department of Mechanical Engineering
May 15, 2020

Certified by .....

George Barbastathis
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....

Nicolas Hadjiconstantinou
Chairman, Committee on Graduate Theses

On the Use of Prior Knowledge in Deep Learning Algorithms

by

Kwabena K. Arthur

Submitted to the Department of Mechanical Engineering
on May 15, 2020, in partial fulfillment of the
requirements for the degree of
Masters of Science in Mechanical Engineering

Abstract

Machine learning algorithms have seen increasing use in the field of computational imaging. In the past few decades, the rapid computing hardware developments such as in GPU, mathematical optimization and the availability of large public domain databases have made these algorithms, increasingly attractive for several imaging problems. While these algorithms have exceeded in tests of generalizability, there is the underlying question of whether these "black-box" approaches are indeed learning the correct tasks. Is there a way for us to incorporate prior knowledge into the underlying framework? In this work, we examine how prior information on a task can be incorporated, to more efficiently make use of deep learning algorithms. First, we investigate the case of phase retrieval. We use our prior knowledge of the light propagation, and embed an approximation of the physical model into our training scheme. We test this on imaging in extremely dark conditions with as low as 1 photon per pixel on average. Secondly, we investigate the case of image-enhancement. We take advantage of the composite nature of the task of transform a low-resolution low-dynamic range image, into a higher resolution, higher dynamic range image. We also investigate the application of mixed losses in this multi-task scheme, learning more efficiently from the composite tasks.

Thesis Supervisor: George Barbastathis
Title: Professor of Mechanical Engineering

Acknowledgements

Firstly, I would like to thank Prof. George Barbastathis. In the past three years, he has been a great influence, guiding and encouraging me in my pursuits. From him, I've learned to be a better thinker and doer, and probably most importantly, to ask a lot of questions and never stop learning.

I would like to thank my amazing group mates. I would like to thank Alexandre Goy, Ayan Sinha, Shuai Li, Mo Deng, Iksung Kang, Subeen Pang, Xavier Sabier and so on. Working with you all has been a great experience, and I know you'll all see success in your futures.

I would like to thank the friends and church-family I have made in my time here. I would like to thank my family, my parents Adelaide Amofah and Wilson Arthur, as well as my siblings, Maame and Kwaku Arthur, for their unending love, and unwavering support.

Lastly, and most importantly, I would like to thank God. Let everything I touch and anything that I do, show your light in this world.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Computational Imaging

Computational Imaging (CI) is the combination of imaging systems and computational algorithms, that seeks to reconstruct an estimate of an unknown object, based on the physical measurements, and often times including prior knowledge of the class of objects [1–4]. A typical computational imaging apparatus is depicted in Fig 1.

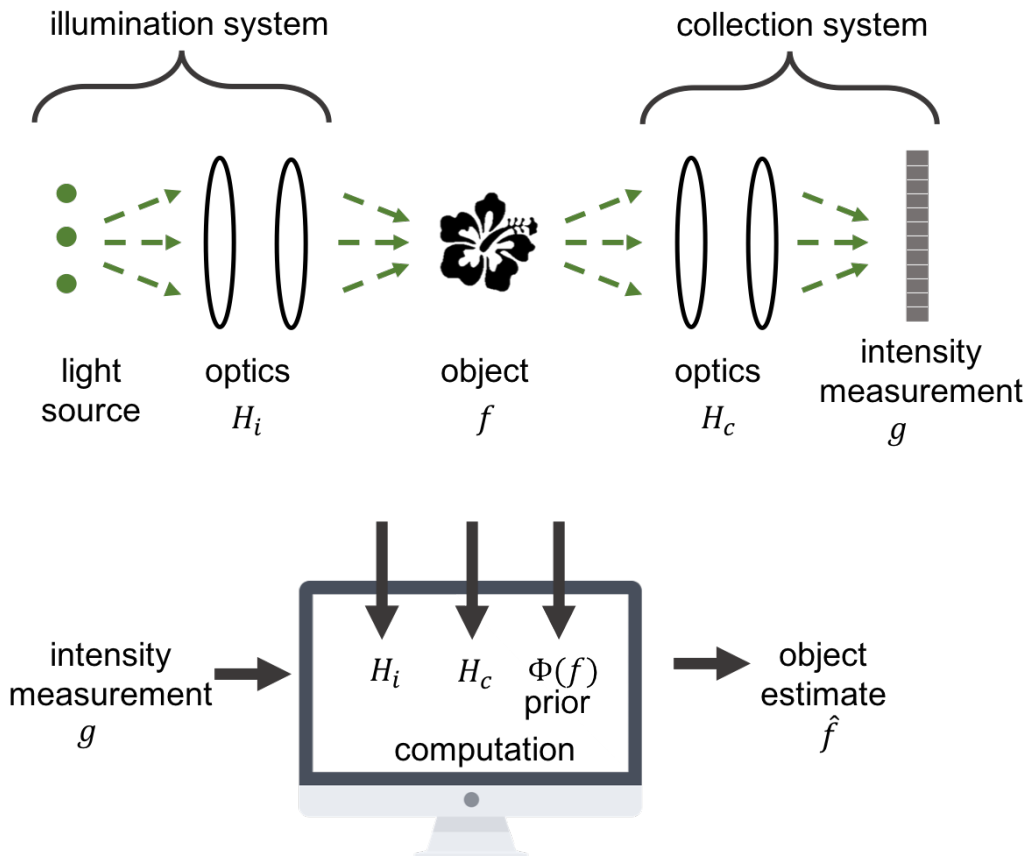

Figure 1: General computational imaging system.

The object often has features or attributes related to its class or unique to the specific object. The analytic expression of these features, which is typically prior knowledge, is known as the regularizer or object prior. The relationship from the object to the raw measurement, subject to the illumination and collection system, is known as the forward operator. The goal of computational imaging is thus to invert the action of this forward operator, in order

to reconstruct the object from the raw measurements. The prior is particularly helpful, as it constrains the space of possible solutions when the forward operator is particularly ill-posed.

CI has found widespread use in biomedical imaging, astronomy, security, manufacturing, etc. Quite often, the raw measurement is not as useful as the information present in the object domain. This is because important information about the object may be occluded by noise, be lost by the system limitation, or be hidden, entangled in the process of measurement. CI is able to recover this important information, therefore relaxing the requirements of imaging systems: the measurements no longer need to represent the exact fidelity (e.g. as resolution, time-sensitivity, contrast) needed for the task. This greatly reduces the required complexity of imaging systems, as well as, allowing for a much wider range of information to be reconstructed from incomplete measurements.

1.2 Inverse Algorithms

By combining the action of the illumination and collection operators, the process of measuring an object can be simplified as:

$$g = H_c H_i f = Hf \tag{1}$$

As previously mentioned, H is often times ill-posed due to the limitations of the system. The measurement g is also typically subject to noise. The common forms are additive white Gaussian noise and Poisson noise. We discuss this later in more detail in Section 2.

A CI algorithm amounts to the following optimization problem:

$$\hat{\mathbf{f}} = \underset{f}{argmin}\, \psi\{H(\mathbf{f}), \mathbf{g}, \Theta(\mathbf{f})\} \tag{2}$$

where ψ is the functional and Θ is the regularizer. For the case of Tikhonov regularization,

it takes the explicit form of:

$$\hat{\mathbf{f}} = \underset{f}{argmin} \{\|Hf - g\|_2^2 + \alpha\|f\|_2^2\} \tag{3}$$

where $\|*\|_2$ is the L^2 norm. The first term is the fidelity term, constraining the fitness by ensuring our estimate creates a measurement that matches our observation. The second term is the regularization, which ensures that our estimate conforms to the class of objects. The regularizer parameter α controls the importance of the two terms: a high α would prioritize prior knowledge to physical measurement, while a low α does the opposite. There is no general α that fits every situation. The selection of the optimal α is crucial, but not straightforward, and is particularly so, when the fidelity term and regularizer term are formed of different functions. Note that this choice of regularizer allows the inverse problem to be simplified, referred to as the Wiener filter [5]:

$$\hat{\mathbf{f}} = (\hat{H}H + \alpha\mathbf{1})^{-1}H^T g, \tag{4}$$

Where H^T is the transport os H and $\mathbf{1}$ is the unit tensor of appropriate dimension. Other more complex regularizers are fidelity terms are chosen. The optimization problem is typically solved through gradient descent optimization.

Gerchberg-Saxton algorithm (GS) is another classical CI technique. GS frames the problem in terms of object domain and measurement domain constraints, and alternately optimizing its estimates in both domains. In the case of phase retrieval, the measurement domain constraint is the intensity of the incident field. In the object domain the constraint is the source phase. The typical flow of a GS algorithm is shown in Fig 2.

GS is typically employed in phase retrieval problems. In the problem of phase retrieval, the phase of a target object is to be obtained from its intensity measurement. The forward operator in this case is the the Fresnel free space operator. We use the GS algorithm is used later in our physics-informed DNN approach in Chapter 2.

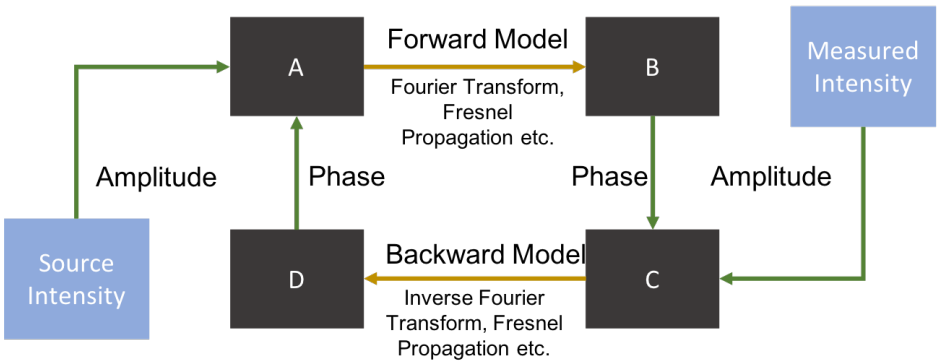

Figure 2: Typical flow of the Gerchberg-Saxton algorithm. The field is estimated in both the object plane and measurement plane, and the source and raw measurement information are applied to the estimates.

1.3 Machine Learning

Machine Learning (ML) is a class of algorithms that allows computational systems to learn complex mappings from given input to given output. They've seen various application in a variety of fields including classification, regression, decision making etc. Machine Learning approaches falls into two general groups: supervised learning and unsupervised learning. In supervised learning, the algorithms work on data-label pairs. In this scheme, the algorithms learn to map a given input to a predetermined label. The data can be presented in its unedited form, or features of interest can be pre-extracted by human operators, and then presented to the algorithms. In the unsupervised case, there is no label; algorithms seek to find underlying structure and information in the data.

Deep learning (DL) is a class of supervised learning where network models are trained to generate an output for a given input. In DL, the data is presented without much pre-extraction, and the algorithm must learn the important features for the desired task. The specific architecture of DL is the neural network: a multilayered network of simple units that when aggregated, can approximate arbitrary complex functions. DL has seen increased popularity as hardware improvement and the availability of data has made them more approachable. They are also very robust, and are capable of approximating a wide range of functions and tasks.

1.4 Fundamental Architecture

The fundamental unit of each layer called a node or neuron. As shown in Fig 3, a neuron takes in an array of inputs, combines them with appropriate weights and bias term, to compute the activation. The output of the neuron is then acted upon by the activation function.

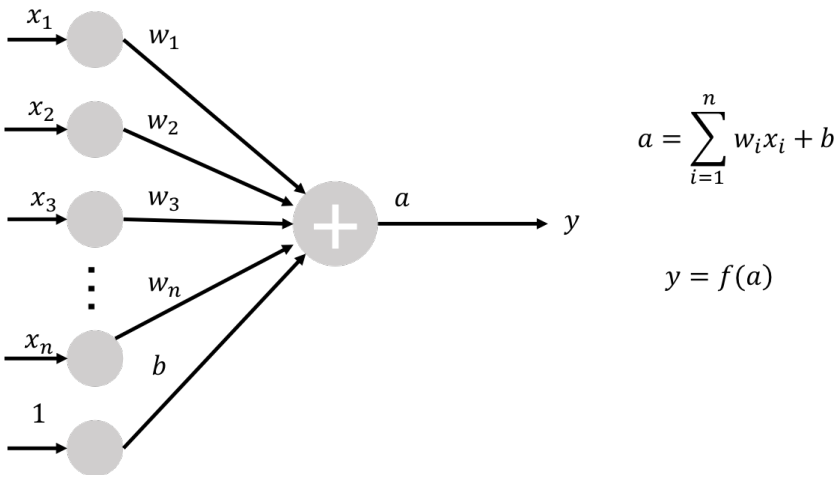

Figure 3: Individual neuron and its activation function, f.

The selection of appropriate activation function is dependent on the task. The activation functions introduces nonlinearity to the NN scheme. The complex connections and nonlinearities are what allows NN to widely approximate arbitrary functions. The weights are the memory of the neural network, that are learned to map the input to the output. A whole range of activation functions are used in practice. The most widely used is the rectified linear unit (ReLU) activation function.

$$ReLU(a) = \begin{cases} 0, & \text{if } a \leq 0 \\ a, & \text{otherwise.} \end{cases} \tag{5}$$

The selection of the activation function, much like the selection of the overall network architecture is a hyperparameter to be set by the designer of the network.

The neurons are arranged in layers. The number of neurons in a single layer is the width,

and the number of layers is the depth of the NN. In general, NN layers fall into the categories of input layers, hidden layers and output layers. The topology and overall size of networks vary greatly, and are model hyperparameters that are selected for the task in scope.

Fully connected networks are seldom used in computational imaging. Image translation would require an unwieldy number of connections, and by extension, weights. Convolutional neural networks have emerged as powerful tools in this domain. For computer vision and computational imaging, they are able to efficiently perform complex tasks. CNNs have specialized layers used in their architecture:

- Pooling Layer. Pooling layers reduce the size of their inputs. This serves the purpose of condensing information and reducing the number of pixels being operated on as networks get deeper. There are two main variants of pooling layers. Max pooling reduces the input by selecting the maximum value from a scanned window. Average pooling reduces the input by simply taking the mean of the scanned window. The size of the pooling window is a user defined hyper parameter.

- Unpooling Layers. Unpooling layers are the counterpart of pooling layers. It increases the lateral size of its inputs. A standard unpooling layer performs this task by repeating values from the input into a predetermined size.

- Skip Connections. Although skip connections are not exclusively used in CNNs, they are highly effective in this domain. Skip connections ensure that networks maintain their performance, even when made deeper. In the case of CNNs, they serve the crucial role of passing higher-frequency (high resolution) information on to deeper layers.

- Convolutional Layers. Whereas a fully connected network may connect every node in a layer to all nodes in the previous layer, convolutional layers connect a need to a smaller neighborhood of nodes in the preceding layer. Even more crucial, this weighted connection is shared across the entire layer. Typically, several convolutional "filters" are used per layer. These "channels" allowing different kinds of features to be extracted.

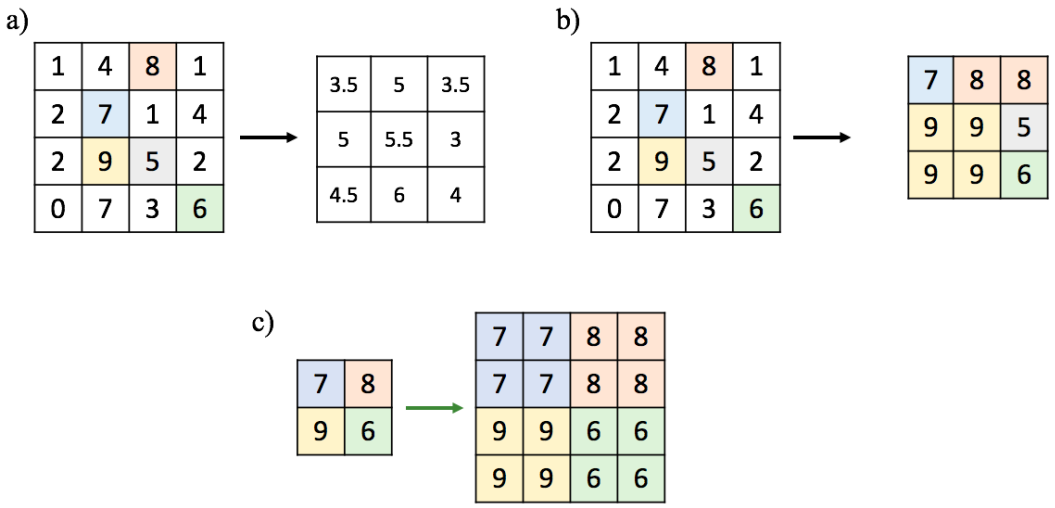

Figure 4: The effect of a) average pooling layer b) Maxpooling layer c) unpooling layer. The filter size is 2 ×2.

The key advantage of these layers is that they convey the translational equivariance nature of natural images. They also maintain local correlations of pixels, and reduce the overall size (number of trainable weights) of the network.

1.4.1 Training

As described earlier, neural networks are *trained* to accomplish specific tasks. In the case of computational imaging, the training data is often in pairs of inputs and outputs, and the process of training refers to the process by which the weights of the network are tuned such that the output of the network closely estimates the desired ground truth. The measure of this discrepancy is called the training loss function. The form of the loss function is another hyperparameter. Popular training loss functions include mean absolute error (MAE), mean squared error (MSE), KL-divergence etc. Loss functions often times involve a regularization term, which constrain the space of possible outputs. This is the typical way in which priors are incorporated into learning architectures.

Networks are initialized with random weights, and the through training, the weights are progressively updated. This is performed numerically using gradient descent algorithms. The weights are connected from end to start using a technique known as backpropagation.

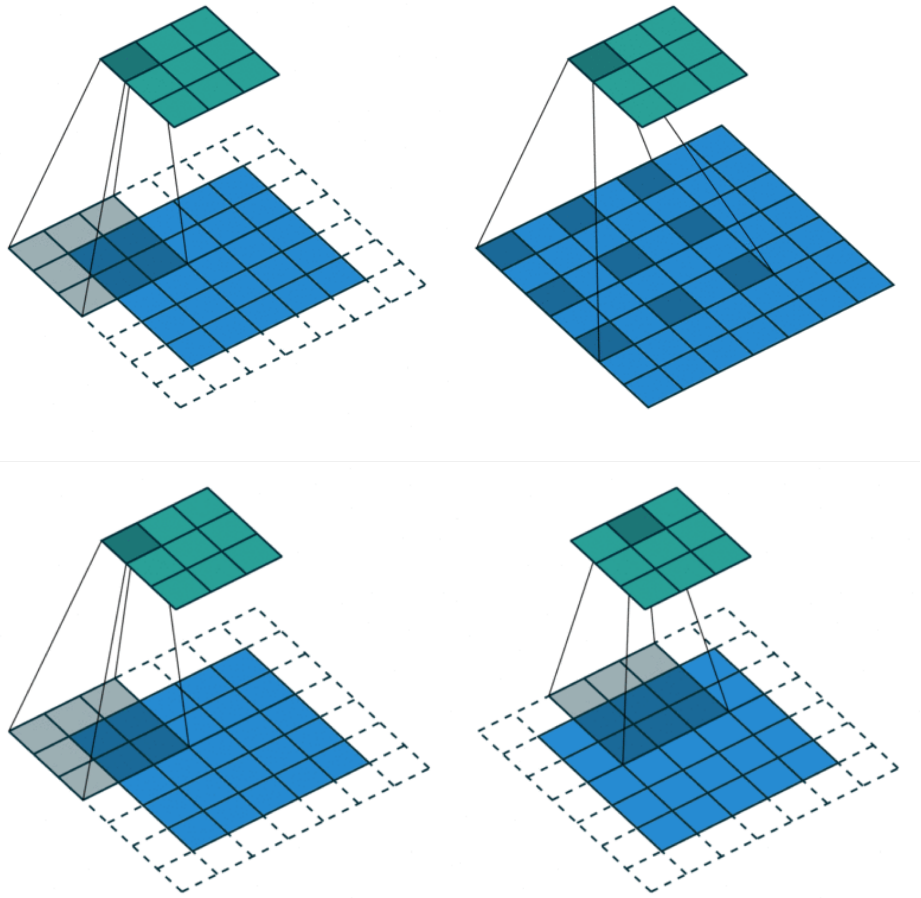

Figure 5: Various convolutional layers: a) convolutional layer b) dilated convolution c) strided convolution. Dilated convolution can be used in certain diffraction imaging, as the dilated convolutions aggregate diffraction effects. This figure is adapted from [6]

.

The learning rate determines the step of each update, and several methods exist to select learning rates. Another important algorithm is batch-normalization, which aids in rapid convergence. Batch normalization seeks to normalize the weights in the network, making deeper layers less sensitive to changes in preceding layers.

Important to learning is data. For typical imaging problem, several tens of thousands of examples need to be readily available for training networks. This is due to size and depth of the networks. Data is typically split into training and testing set. It is essential that the two sets are from the same general distribution. Also, networks are never exposed to testing data during training. This is a problem, as networks can be trained for indefinite periods of

time, which might lead to overfitting. An overfitted network will perform well on training data, but fair poorly on the test data. In a sense, the network might be memorizing features in the training set that were not intended to be memorized. As such, a third set of data is produced from the general distribution. The validation set is generated to intermittently test the networks performance, ending training before the network overfits.

DL holds several advantages over traditional CI algorithms. They are much faster in inference. With the most recent hardware, most NN can generate estimates in a matter of subseconds, while a traditional CI algorithm with require several iterations to reach reasonable convergence. Traditional CI algorithms often require the exact form of the operator and knowledge prior in order to operate. This is particularly preventative as often times, the closed for optimal priors are unknown. DL algorithms in contrast learn the priors and operators directly from the data.

DNNs have been increasingly used in computational imaging problems including phase retrieval [7–9], tomography [10, 11], ghost imaging [12], lensless imaging [13–15], imaging through dense scattering media [14, 16]. A more comprehensive analysis of the use of DNN in imaging can be found in [17].

1.5 Thesis Structure

This thesis contains three chapters discussing various approaches to incorporating prior knowledge into deep learning algorithms. Chapter 2 describes the approach of embedding physical knowledge of the imaging system into a CNN. Chapter 3 describes incorporating prior knowledge on the composition of the task into learning architecture. Finally, Chapter 3 draws conclusions from the work, discussing its successes and drawbacks, as well as proposing future avenues of investigation. While the physics-informed network was learns underlying physics of the optical imaging array, the operator H, the MTL network learns the effects of the detector in its limited resolution and memory, the operator H_c.

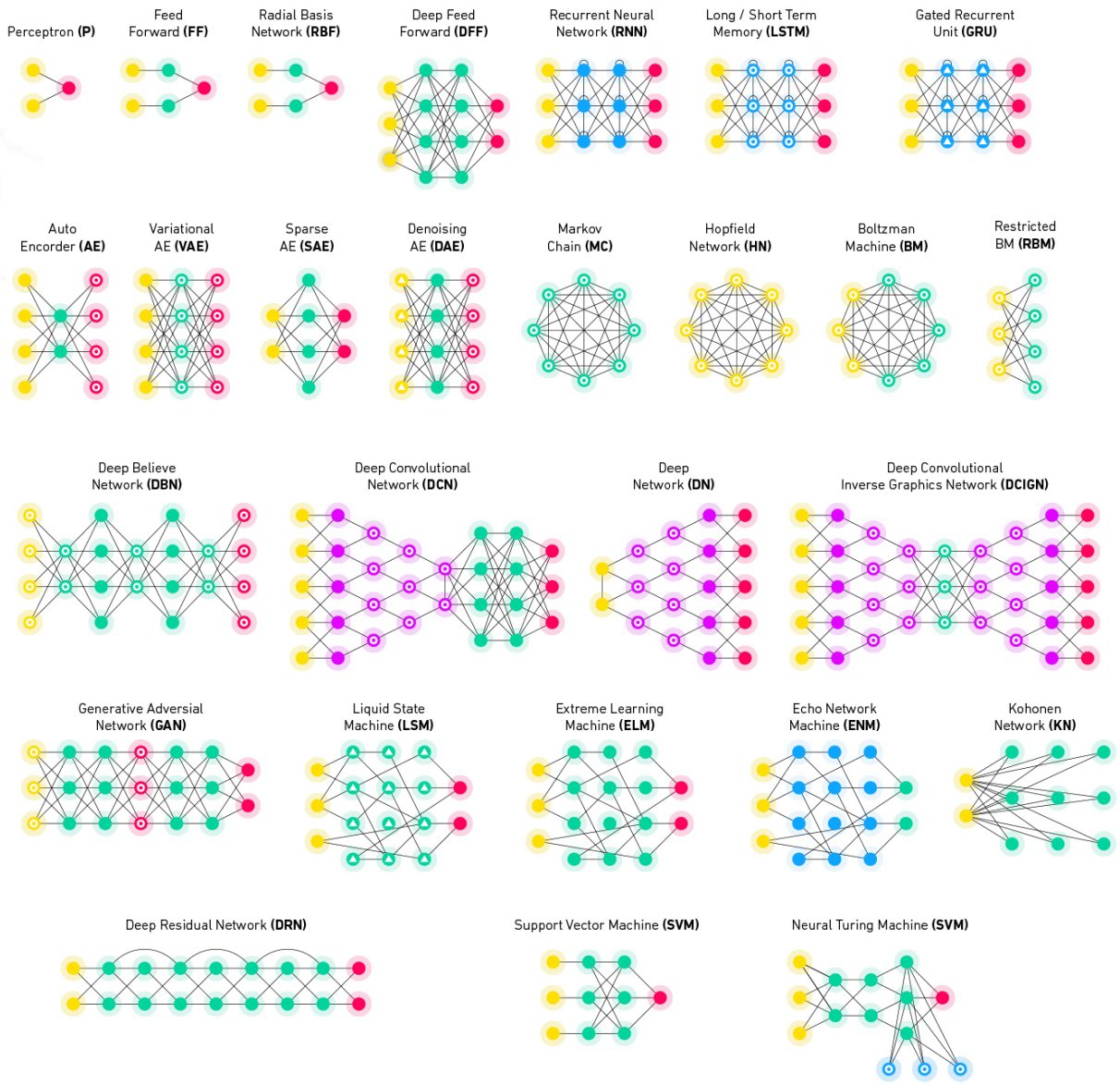

Figure 6: Various deep learning architectures. This figure is adapted from [18]

.

2 Prior Knowledge of Physics

2.1 Background

Quantitative phase retrieval (QPR) is a classical inversion problem in computational imaging. The objective of QPR, is the reconstruction of phase information from a measurement. This phase information is lost since conventional imaging systems, such as cameras, are incapable of recording the phase of incident illumination, only its intensity.

Traditional methods exist to solve the problem of QPR: transport-of-intensity based methods [19], digital holography and inteferometry approaches, input-output or the aforementioned Gerchberg-Saxton-Fienup (GSF) iterative methods [20,21], optimization approaches [22] etc. Several of these methods either require additional apparatus, or multiple images, in order to reconstruct the lost phase information. We propose to a system that can recover lost phase information from a single intensity image, with no additional apparatus.

In this work, we employ a deep neural network (DNN) for QPR under very adverse light conditions. DNNs are particularly useful due to their versatility and generalizability. They can quickly generate reconstructions in a single step, while other algorithms such as the GSF, require several iterations for a single object. The drawback of the DNN approach is the need of large, diverse, well-curated datasets. This issue has largely been reduced as more image datasets such as ImageNet [23], MNIST [24], and Faces-LFW [25] are publicly available. The other issue, is the perceived "black-box" nature of the networks. How can one ensure that the network is learning the correct problem? We intend to directly utilize our knowledge of the physics in the reconstruction. This physics-informed approach ensures that our architecture incorporates the physical laws that govern the image formation on the detector. It has the added benefit of making the network more efficient, and robust enough to image under very noisy low-light conditions. We also employ the negative Pearson correlation coefficient (NPCC) as the training loss function.

2.2 Physics-Informed Approach

We implement a DNN approach to solving coherent phase retrieval under very low light conditions. The phase retrieval problem in this work can be expressed for a thin object as:

$$\mathbf{g}(x,y) = |F_L[u_{inc}(x,y)\mathbf{t}(x,y)e^{j\mathbf{f}(x,y)}]|^2 \tag{6}$$

where (x,y) are the lateral coordinates, $\mathbf{g}$ is the intensity measurement in the detector plane, $\mathbf{t}$ and $\mathbf{f}$ are, respectively, the amplitude and phase of the field immediately after the object, u_{inc} is the incident field in the object plane, and F_L is the Fresnel propagation operator over a distance L. We assume the object is purely phase, therefore $\mathbf{t}(x,y) = 1$, and we define $\mathbf{g} = H(\mathbf{f})$. The DNN is implicitly solving the problem:

$$\hat{\mathbf{f}} = \underset{f}{argmin}\, \psi\{H(\mathbf{f}), \mathbf{g}, \Theta(\mathbf{f})\} \tag{7}$$

where ψ is the functional to minimize. Θ is the regularizer imposing constraints on the possible solutions, $\hat{\mathbf{f}}$. In a classical optimization, the regularizer must be chosen a priori, with appropriate weight factor. This is often times problematic, as the closed form of the optimal regularizer is not always known. There is also the additional task of selecting the regularizer weight factor. In our case, the DNN learns to regularizer based on the class of objects it experiences during training.

2.2.1 Low Light Physics

We are particularly interested in applying our method to a photon-starved system. When the light source is weak, the detection signal-to-noise ratio (SNR) is ultimately limited by the quantized nature of light. This issue arises when the illumination source has a fixed power output, or when there is a limit to the total allowable radiation (e.g. in medical x-ray imaging). In this regime, shot noise cannot be avoided and regularization schemes

are required, as prior information is needed to offset the loss of information. As this noise becomes increasingly dominant, many reconstruction algorithms' performance deteriorates. We show that our trained network maintains its performance in these adverse conditions.

2.2.2 Approximant

QPR as posed in Eqn. 6 cannot be inverted directly for the sole reason that, conventional detectors are incapable of measuring the phase of an incident light field. Conventional detectors record the intensity (magnitude-square of the field). Several effects are therefore hidden in the measurement scheme including any noise and pertinent phase information. Conventional CI approaches such as the Gerchberg-Saxton-Fienup and gradient descent algorithms implicitly encode insights of physical propagation to generate reconstruction. They use the well known inverse Fresnel operator to propagate the incident field back into the object domain, allowing for increasingly accurate reconstructions. Since the incident phase is not known, an approximate phase is used in lieu, to allow for the projection of field back into the object plane.

We associate the phase of the incident beam in the detector plane with the square root of the intensity measurement to produce a complex field, which is then propagated back to the object plane. The phase of this complex field in the object plane is referred to as an "approximant" (or GS-approximant as it is inspired by the GS algorithm) and it is generally closer to the solution than the raw intensity measurement. Note that the adjoint of operator H, used in the gradient descent method, can also be used to generate an approximant, however, we will restrict our analysis to the GS-approximant. The approximant can be used in lieu of the raw measurement for the DNN training. This is an example of a "physics-informed" method as part of the physical process is embedded in the approximant itself.

2.3 Experimental Appartus

2.3.1 Optical System

The optical apparatus is depicted in Fig 7. The source illumination is a Helium-Neon laser emitting a continuous beam of wavelength 632.8 nm. We use a calibrated variable neutral density filter (VND), to control the intensity of light entering the system. The beam is also linearly polarized for maximal operation of the spatial light modulator (SLM). After spatial filtering and expansion of the beam, it is passed through the SLM (Holoeye LC2012). This SLM operates in transmission mode, and has a pixel size of 36 μm. The modulated light is then filtered by a second polarizer, before passing through a telescope apparatus, in order to fit the diffracted pattern within the detector. The telescope system reduces the size of the SLM surface by a factor of 2.3. We use an EM-CCD camera (QImaging. Rolera EM-C2) with a pixel size of 8 μm. The parameters of this camera are all controlled in software. The detector is then placed at a distance $\delta z = 400mm$ from the image plane, to allow for modulated light to propagate. A neutral density filter with optical density of 2 is placed in front of the detector to suppress background light and further control photon level range.

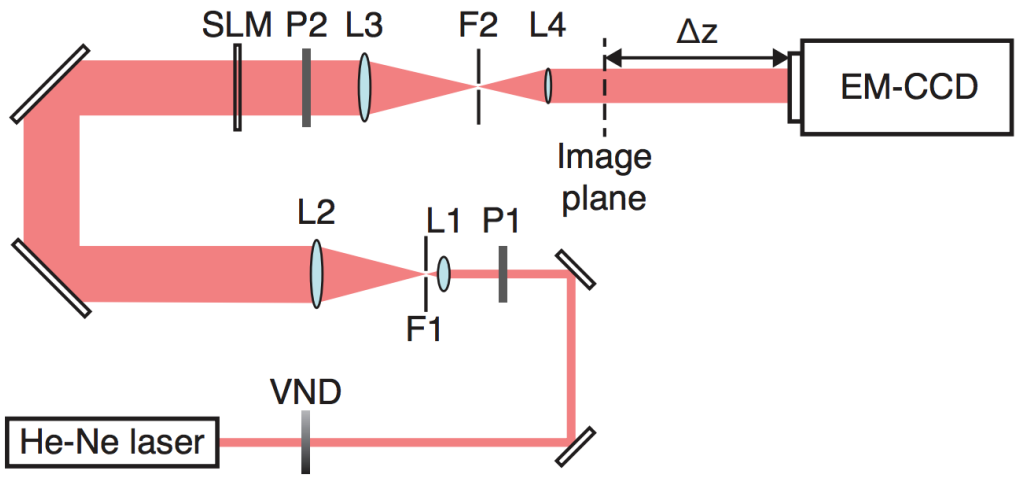

Figure 7: Optical apparatus. VND: variable neutral density filter, P1-P2: polarizers, L1: 10x objective, L2: 100 mm lens, L3: 230 mm lens, L4: 100 mm lens, F1: 5 μm pinhole, F2: iris. SLM: transmissive spatial light modulator.

2.3.2 Image Datasets

Two image datasets were prepared for the task of phase retrieval. The first was a set of Integrated Circuit layouts. The IC database was generated from a GDSII file of a typical circuit. Materials and thickness were assigned to each layer using a custom design process design kit (PDK), allowing for the phase delay caused by each layer to be simulated. This dataset is more restricted however, since there is a strong prior of rectangular Manhattan geometries. We also trained and tested our algorithm on the more general ImageNet database. This is much more general dataset containing natural images of animals, plants, scenes etc.

2.3.3 Experimentation

We generated data for both datasets, at various noise levels. The noise levels and photon counts for the experiments are summarized in Table 1. We generated 10,000 examples for each, with a train-validation-test split of 95%-4.5%-5%. The ground truth displayed onto the SLM was set to 256 by 256 pixels, and the detector images, originally 1002 by 1002, were interpolated using bilinear interpolation to be 256 by 256 pixels. In order to generated the approximant, each detector image is first zero-padded such that the inverse Fresnel propagator would create an approximate of size 256 by 256 pixels, performing both the Fresnel propagation and the resampling in one step.

2.4 DNN Architecture

We used a densely connected convolutional network for the QPR task. This is the same network used in [26] and the architecture is shown in Fig 8. We use five blocks for the convolution+downsampling followed by 5 blocks for deconvolution+upsampling. This is followed by two blocks for the final estimation. IDiffNet has internal feed forward connections that promote feature propagation, encouraging more direct connections in each block and allowing features to be reused.

Experiment	EM gain	Photon count ± 5 %	SNR	Limit SNR
1	1	1050	20	32
2	1	85	2.7	9.2
3	1	44	1.45	6.6
4	4.8	9.9	0.9	3.1
5	54	1.1	0.5	1.0
6	54	0.25	0.24	0.5

Table 1: Noise levels and photon counts for the experiments in Fig 7. The illumination conditions are the same for both the IC layout and the ImageNet datasets. The photon count is the effective number of photons after dividing by the quantum efficienty per detector pixel averaged over the whole detector field for the incident beam (no modulation on the SLM). The procedure for measuring the photon count is given in the supplementary material. The SNR is the mean of the incident beam signal divided by its standard deviation and averaged over the whole field of view. The limit SNR is the square root of the number of photons.

2.4.1 Loss Function

The loss function used in this work is the negative Pearson correlation coefficient (NPCC). For two images A and B, with pixel indexed by i, the NPCC is defined as:

$$NPCC(A,B) = \frac{-\Sigma_i (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\Sigma_i (A_i - \bar{A})^2 (B_i - \bar{B})^2}} \tag{8}$$

We used this in place of conventional loss functions such as mean square error, as it proved to be a better metric for training of DNNs in phase retrieval.

2.5 Results

Sample reconstruction from both IC layouts and ImageNet test sets are shown in Fig 9. From Fig 9 (g-j) and (o-r), it is clear that the DNN efficiently supresses the granularity typical of shot noise. The end-to-end reconstructions appear similar to a low-pass filtered version of the original image, particularly in the ImageNet dataset. IC layout reconstructions still maintain their sharp edges, as this feature is prominent in the class of objects. The physics-informed reconstructions are of higher fidelity, since high frequencies are provided to the

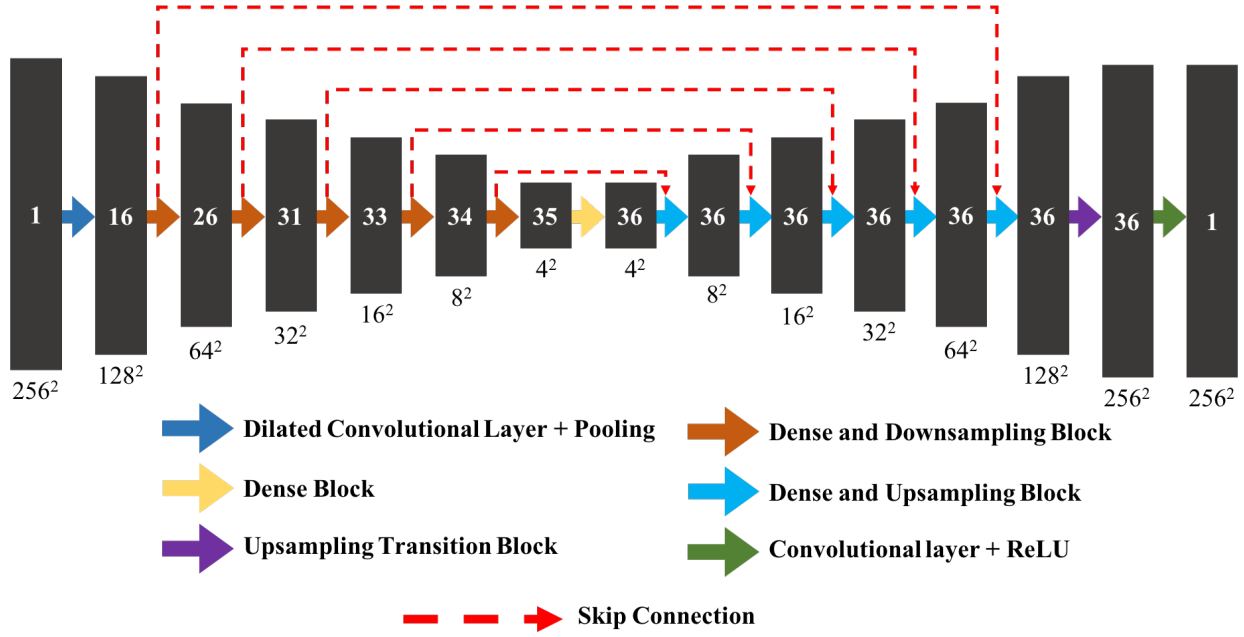

Figure 8: IDiffNet architecture used in phase retrieval problem. The numbers shown in each layer are the number of filter kernels used.

DNN by the approximant (as is visible in Fig. 9 q). In the low photon case of the IC layout (Fig. 9 t), the general pattern is recovered, though additional features have been included by the DNN. The DNN included these features due to the strong periodicity present in ICs.

Fig. 10 shows the performance of the different met'hods at different photon levels. We use the PCC metric to evaluate visual quality of the reconstructions. In the case of the IC layout, the physics-informed methods outperforms the end-to-end approach, which in turn performs better than the GS algorithm. A similar result is obtained in the ImageNet data set, except that the difference between the physics-informed and the end-to-end scheme is less pronounced. The standard deviation of the reconstruction quality is much larger in the ImageNet test set, even at high photon levels. This trend does not appear in the GS reconstruction for high photon level, as their standard deviation remains equally large. This discrepancy confirms that the DNN efficiently exploits the strong prior in the IC layout geometry.

Since the PCC is invariant to the magnitude of the images (i.e. PCC(A,B) = PCC(αA,

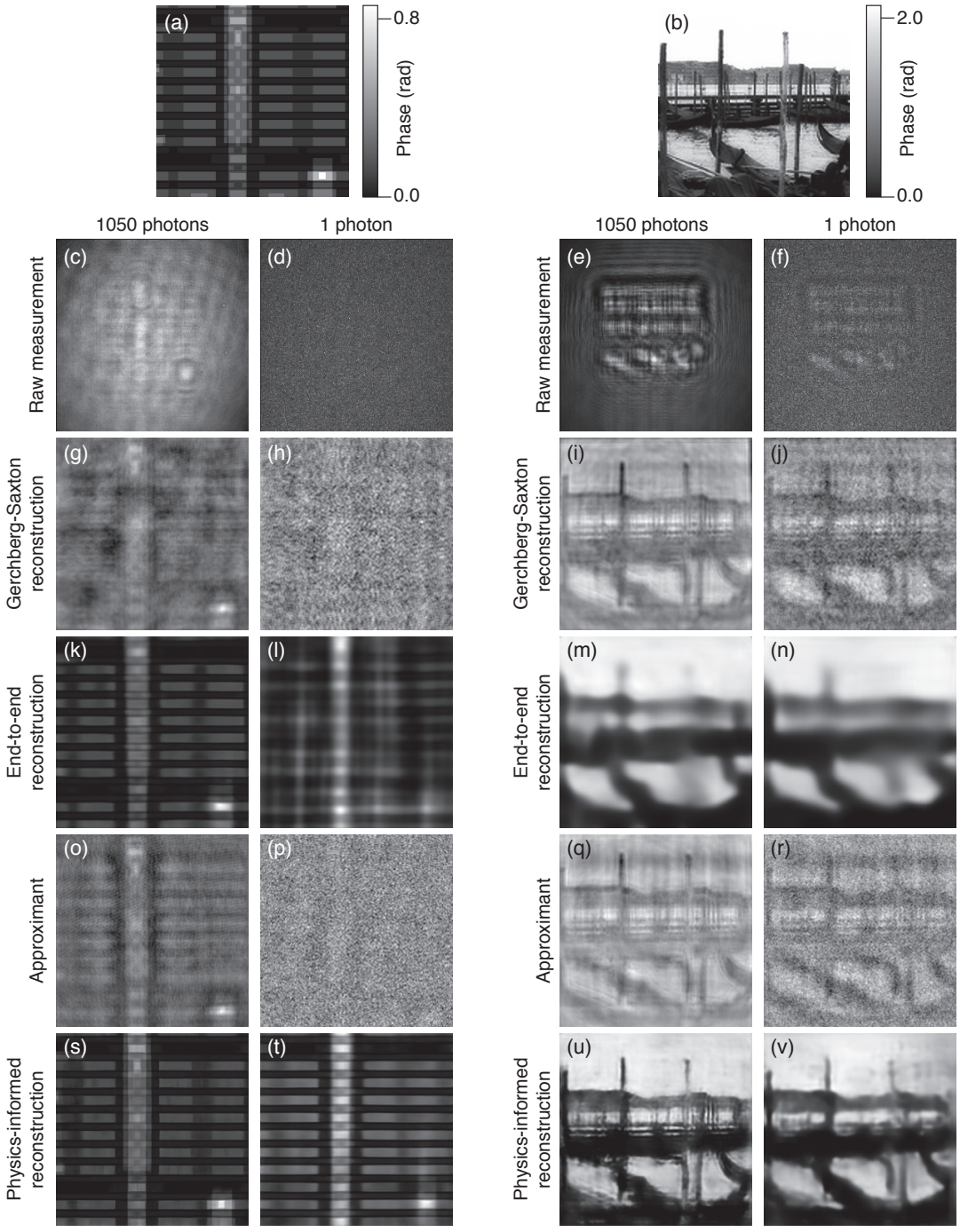

Figure 9: (a-b) Ground truth phase of one example from each test set of IC layouts and ImageNet. (c-f) Raw measurements in the detector plane. (g-j) Gerchberg-Saxton algorithm reconstructions from the raw measurements c-f. (k-n) DNN reconstructions with the end-to-end method. (o-r) Approximants in the image plane. (s-v) DNN reconstructions from the approximants o-r with the physics-informed method. For better display, the grayscales of all images have been normalized to range from the minimal to the maximal value.

βB), for real numbers α, β), the reconstructed phase images differ from the ground truth by some scaling factor. However, for a given DNN, the scaling factor is constant, and can be obtained by comparing the reconstructions and ground truths of the validation set. In practice, the scaling factor is obtained by comparing the histograms of the ground truths and reconstruction images.

For the both Fig. 10 and Fig. 9, it is clear that the approximant helps in recovering high fidelity images. This physical preprocessing step incorporates the known physical relations into the mapping from intensity image, to ground truth phase object. There are however several ways to compute the approximant. A gradient-descent approximate can also be used as an approximant. The GS-approximant we used here corresponds to half of the first iteration of the GS algorithm. In our investigations, further iterations did not yield any significance difference.

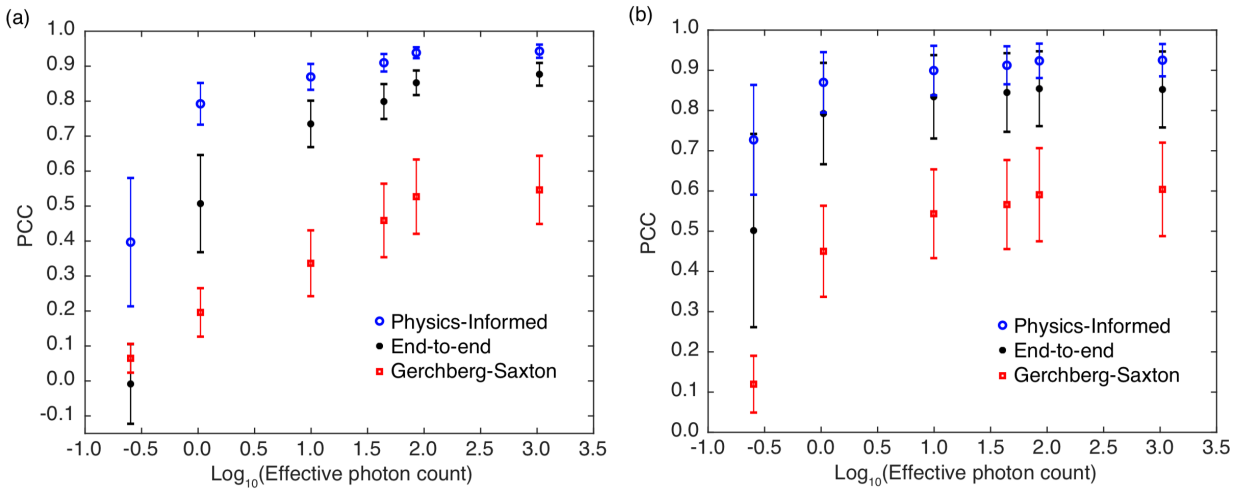

Figure 10: Pearson correlation coefficient between the ground truth and the DNN reconstructions. (a) IC layout data set. (b) ImageNet data set. The markers indicate the mean over the test set (50 examples) and the error bars $\pm$ 1 standard deviation from the mean.

2.6 My Contributions

The work in this chapter was led by Alexandre Goy. My contribution was in building up the computation hardware, generation of simulations, and training the neural network.

3 Prior Knowledge of Task Composition

3.1 Background

As previously stated, the availability of large labelled datasets is crucial for most deep learning algorithms. Supervised Learning, which forms the bulk of recent progress in machine learning, requires the data-label pairs, as the features important to the task are left for the algorithm to learn. Transfer Learning evolved as a natural work-around for this issue of large datasets. In Transfer Learning, a network to be trained for a particular task A, is first pre-trained to learn an adjacent task B. The assumption is of course, that ample labelled data exists for B, and that the tasks for A and B are reasonably adjacent. Multi-task Learning (MTL) is a similar technique. The goal of multitask learning is to learn multiple tasks (Task A, Task B, Task C) and share information learned by the separate networks towards those tasks.

MTL has been successfully implemented in several application [27–29]. MTL allows networks to accomplish tasks (for example Task C) by employing a wider range of important features (e.g. from Tasks A, B and C), that it would not have learned if it was focused solely on performing Task C. It forces networks to generalize and learn domain-specific as well as task-specific features.

We employ this MTL scheme in the joint task of image enhancement. We seek to take a low resolution low dynamic range (LR-LDR) image and transform it into a high-resolution high dynamic range (HR-HDR) image. We decompose the tasks into super-resolution and inverse tone mapping. Note that this is a joint-task, a subset of multi-task, since the target task is composed of the two lesser tasks. We in particular seek to use mixes losses for each task, allowing the smaller networks to be more task specific, but leading to a more generalized MTL network.

3.1.1 Super-resolution

Super-resolution is a computer vision problem, that has been greatly advanced through the use of deep neural networks. Super-resolution seeks to take a low-resolution image (LR) and transform that into a higher resolution image (HR). The limited resolution maybe be due to physical constraints of the imaging system (e.g. upsampling by the detector), or due to limitation in hardware (e.g. blur by the optics or motion). CNNs have proven to be very efficient in this domain, and several such networks have greatly improved upon prior SR algorithms. Several DNNs have been used for the SR task [30–34].

3.1.2 Inverse Tone Mapping

The dynamic range of an image is the range of intensity values a pixel in the image can assume. The human eye has a much higher dynamic range, than a standard image. This is because standard dynamic range images are restricted to 8-bit (can assume 256 possible values). Inverse tone mapping is a another computer vision task, where the dynamic range of an image is increased. The goal isn't to artificially produce more bits, but rather, probabilistically represent the dynamic range of the original scene in which the image was taken. Again, deep networks have excelled at this task [35, 36].

3.2 MT Network

We propose U-Net architecture for the combined task of SR and DR. Individual U-Nets are trained for the singular tasks of SR and DR. The encoder layers of the individual tasks are the used in the combined task, with skip connections passing information from shallower networks to deeper networks. Consequently, the work of converting LR-LDR to HR-HDR is handled by the decoder of the combined task network.

The architecture of our network is illustrated in Fig. 11. The main layers used are Convolutional, ReLU and MaxPooling layers. The number of filters in the convolutional layers increase with depth as the complexity of features being extracted increases. Skip

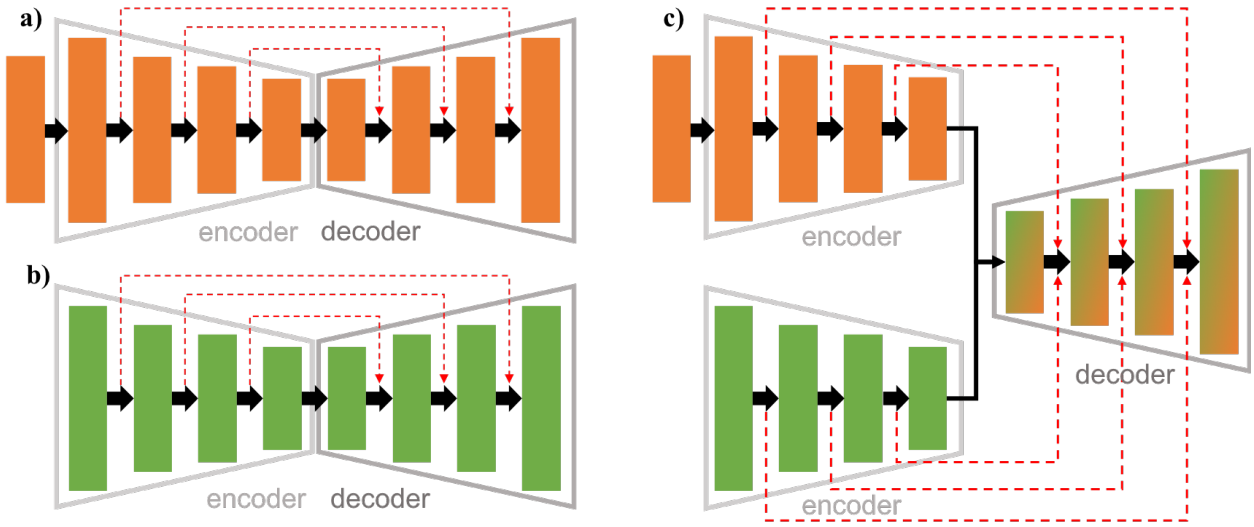

Figure 11: Diagrams of a) SR network b) ITM network c) MT network. The encoders of the individual tasks are shared with the joint-task, whose decoder is trained to combine information from both encoders.

connections pass higher resolution information to deeper layers.

3.2.1 Individual Tasks

The architecture for the U-Net used in the individual tasks is shown in Fig. 12. It is a fully convolutional network with skip connections. The skip connections are important in passing in higher resolution details to deeper networks for reconstruction. They also ensure that the depth of the network is not detrimental to finer details in the reconstruction.

The encoder blocks are series of Convolutional layer, Batch Normalization and ReLU activation layers. Between each pair of blocks, a maxpooling layer is present to reduce the layer size. The decoder blocks have the same configuration, with unpooling layers instead of maxpooling.

The SR network has an additional upsampling step appended to its input. This is to maintain the same depth-size relation across the tasks. It has also been noted that preprocessing has the potential of greatly improving results, as the network needs to learn the residual of the underlying physics.

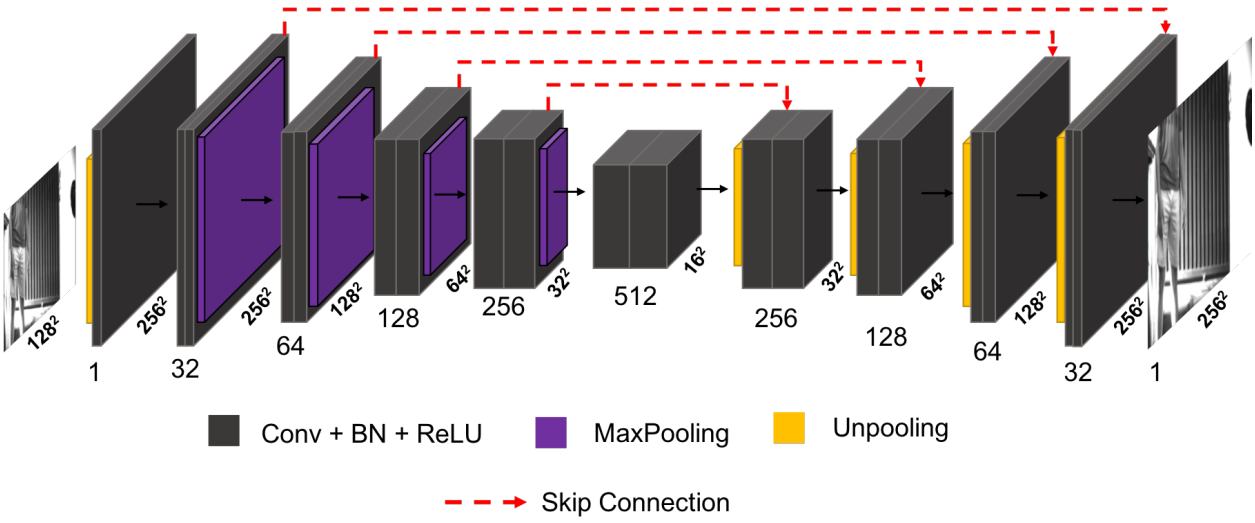

Figure 12: CNN architecture for MTL training scheme. In the first phase, the individual task networks are trained. The encoder layers are shared between the networks. Skip connections connect the output from each block, to the input of the connected block

3.2.2 Combined Tasks

The encoders from the individual tasks are shared with the combined task. This allows information from both tasks to be presented for the combined tasks at the different depth and sampling levels. Some MTL approaches emphasize a joint-task approach, where only few layers are shared. We did not chose this method as the burden on the decoder with mixed losses will only be exacerbated with such a scheme.

The whole network is trained end-to-end, with the learned features from the encoders held constant. The decoder learns the mapping from these learned features to the final HR-HDR image.

The advantage of such a scheme is that the combined network benefits form the filters learned in the individual tasks, and learns the best method for blending information passed from both. We are also free to pick different loss function from the three tasks (SR, DR and MT), allowing for even more diversity and accuracy in the feature extraction layers.

3.2.3 Mixed Losses

We experimented with mixed losses in a non-conventional scheme. Typically, mixed losses are implemented as weighted combined losses, as showing in Eqn. 9. The weight terms λ_i control the relative importance of the different metrics ψ_i. In the case of MTL, there would be a separate metric for each SR, ITM and decoder reconstruction.

$$\hat{\mathbf{f}} = \underset{f}{argmin} \sum_{i=1}^{N} \lambda_i \psi_i \{H_i(\mathbf{f}), \mathbf{g}_i, \Theta(\mathbf{f})\} \tag{9}$$

We implement mixed losses by training the different tasks with different losses, and then training a decoder to mix the contributions of these losses. In a sense, we are reframing the optimization problem into:

$$\hat{\mathbf{f}} = \underset{f}{argmin}\, \psi \{H(\mathbf{f}, H_1, ..., H_N), \mathbf{g}, \Theta(\mathbf{f})\} \tag{10}$$

The decoder in our architecture learns to translate between the different loss functions. Though this increases the complexity of the task of the decoder, there are several benefits. Firstly, the losses of the individual tasks may be selected to optimize the individual tasks. This increases the accuracy of the individual tasks. Secondly, mixed losses have the potential to improve results from the combined scheme. It increases the diversity of the features extracted and removes the additional parameter of relative strength of weights.

3.3 Data Collection

We designed and fabricated a semi-autonomous wheeled robot for the collection of images for the multi-task learning. The semi-autonomous robot is shown in Fig. 13.The robot involved a camera on a raised platform that collected data with set parameters and exposure times. It run semi-autonomously, moving through hallways in order to capture different scenes. Its inputs were ultrasonic sensors oriented radially outward to avoid obstacles. It was controlled through specialized firmware programmed into a microcontroller.

Figure 13: Autonomous robot for image acquisition.

3.3.1 Mechanical Design

The robot base comprised of two motorized wheels and a castor wheel. The wheels were actuated by two PWM motors, individual controlled. Five ultrasonic sensor were placed at regular intervals along the front facing semicircle. An Arduino microcontroller was programmed with firmware to allow for semi-autonomous operation. The robot could travelled down long hallways nigh-autonomously.

Two platforms were built unto the base of the robot. The first platform rose 7 inches above the base, supporting the laptop and any additional storage devices. The second platform rose 18 inches above the previous. It held the camera mount, and additional imaging equipment needed to collect data.

3.3.2 Camera Operation

The Thorlabs DCC1240M camera was used to capture image data. Software run on a computer controlled the camera and microcontroller. The microcontroller was instructed to move the robot for a set duration. Then when at full rest, the camera was instructed to capture data. Then, the cycle repeats, allowing for the robot to autonomously travel while collecting data.

At each of the scene capturing step, the camera collected five images of the same scene at different exposure levels. In order they were: 140 milliseconds, 70 milliseconds, 35 milliseconds, 17 milliseconds and 8 milliseconds. The pixel clock was set to 13 MHz, and the image size was 1280 by 1024. The images were 8 bit monochrome format.

To ensure a good variety of images and scenes, images were captured along different paths, and at different times of the day. It was observed that data collection at mornings and evenings were most successful as the scenes were largely static, and were varied in brightness. The images were then manually curated, removing any unusable images from the set. Fig. 14 shows sample images collected by the autonomous robot.

Figure 14: Sample images collected with the autonomous robot. In order, the exposure times are a) 140 milliseconds, b) 70 milliseconds, c) 35 milliseconds, d) 17 milliseconds and e) 8 milliseconds. Images were taken at different times of the day, and along different paths.

3.4 Experimentation

3.4.1 Dataset Synthesis

From the collected images, we synthesized data for the training scheme. From each set of five images, with corresponding exposure times, we generate an HDR image, using the MATLAB

"makehdr" implementation [37]. For each set, the middle exposure image is used as the SDR image, as it balances high and low intensity information. From the generated HDR and SDR images, low resolution version are created. This is down by downsampling the images by a factor of 2.

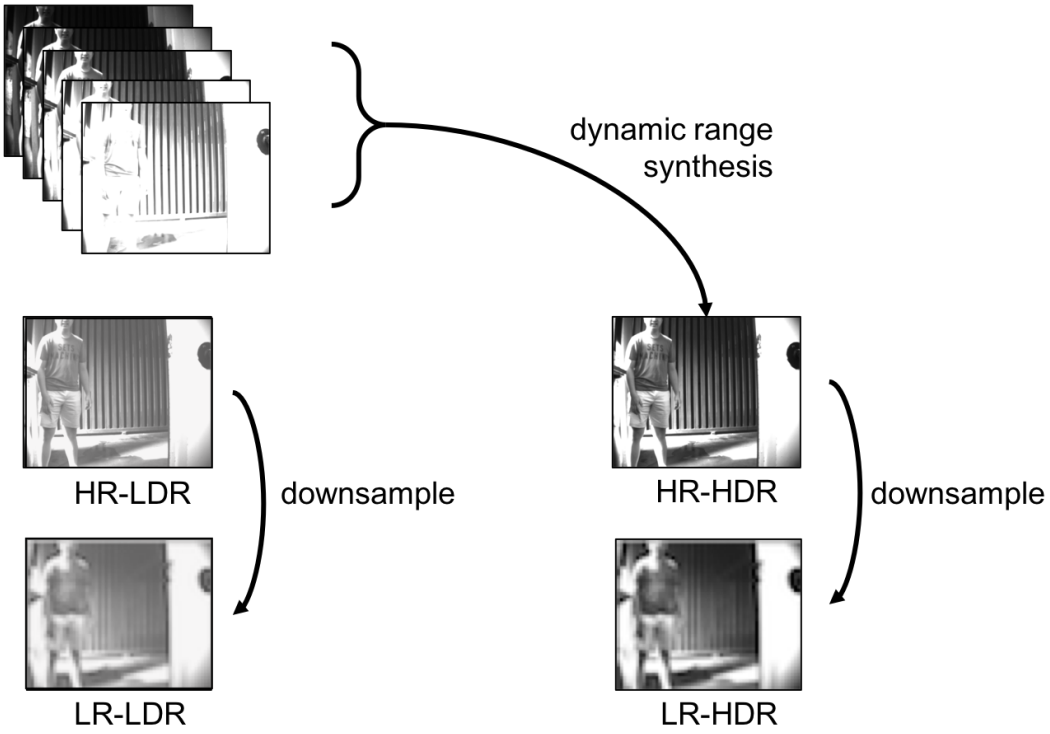

Figure 15: Dataset generation process flow. An HDR image is generated from a set of SDR images. The high resolution HDR-SDR pair is then downsampled to generate their low resolution counterparts.

3.4.2 Training

We trained the network on an Nvidia Titan X GPU. Since the network was fully convolutional, it was invariant to input size greater than 2^4. This is governed by the number of MaxPooling layers. As such, we cropped the images to 256 x 256, allowing us to generate 10K training examples from the collected dataset. The networks were trained until reaching reasonable convergence.

We investigated three loss functions: MAE, NPCC and structural similarity index (SSIM)

[38]. We first trained the networks to complete the singular tasks of SR and ITM, independently. Following this, the trained encoders for SR and ITM were then coupled with a decoder. The decoder was trained to complete the MT of generating HR-HDR images from the inputs of the two encoders. We tested the use of the three aforementioned loss functions in this case as well.

We also compared this MT network to two other network schemes: networks in serial, and end-to-end network. To ensure a fair comparison, the size of the end-to-end network was comparable to the size of the MT network *i.e.* the number of encoders in the end-to-end network is doubled to match that of the MT network.

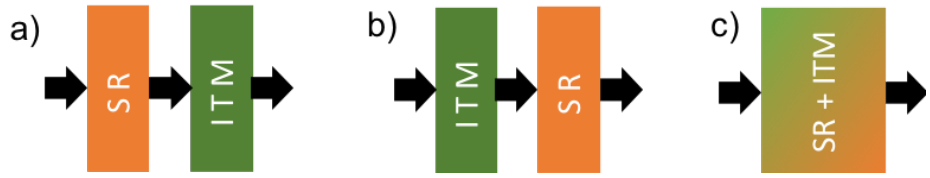

Figure 16: Other deep neural network configuration for the MT. Networks may be trained separately, and used in serial configuration: a) SR followed by ITM networks b) ITM followed by SR networks c) single network trained end-to-end.

3.5 Results

3.5.1 Singular Tasks

The performance of the networks trained with different loss function is summarized in Table 2 for SR, and Table 3 for ITM. For the task of SR, MAE and NPCC were the best performing loss functions. While MAE focuses more on per-pixel error, NPCC seeks to reduce overall image dissimilarity. MAE is thus much more quantitative, while NPCC is a much more qualitative, visual metric. For the task of ITM, SSIM performed best as the training loss function.

Loss Function	PSNR(dB)	PCC	SSIM
MAE	31.2	0.98	0.95
NPCC	39.9	0.99	0.98
SSIM	25.3	0.99	0.93
MSE	33.7	0.97	0.90

Table 2: Performance of the SR network. To correct the scale invariance, the test reconstructions were normalized by the validation set.

Loss Function	PSNR(dB)	PCC	SSIM
MAE	26.6	0.87	0.97
NPCC	10.8	0.91	0.82
SSIM	32.3	0.93	0.98
MSE	13.9	0.77	0.94

Table 3: Performance of the ITM network. To correct the scale invariance, the test reconstructions were normalized by the validation set.

3.5.2 Multi-Task with Serial Networks

Table 4 summarizes the performance of the two serial approaches. It is clear that the SR-ITM serial network outperforms the ITM-SR serial network. ITM is inherently more difficult a task, compared to SR. This is most likely due to saturation in bright pixels and effects of dark pixels. When used as the first network, teconstruction error generated in the ITM-network, is passed on to the SR-network. This increases the overall error of the combined network.

3.5.3 Multi-Task with Mixed Loss Decoders

With the trained encoders from the SR and ITM-networks, we created the MTL network, training a decoder to generate the final HR-HDR image from information passed by the two encoders. The results from the MT network is summarized quantitatively in Table 4. The MT Network showed improved results over the end-to-end and serial approaches. By utilizing our knowledge of how the task is composed, we were able to design our network to incorporate this prior knowledge into its learning scheme. The MTL approach deconvolves

that tasks, allowing the effects of ITM and SR to be efficiently and independently learned.

	PSNR	PCC	SSIM
MTL	28.2	0.89	0.98
End-to-end	27.1	0.85	0.98
SR-ITM	19.0	0.85	0.94
HDR-SR	16.8	0.81	0.94

Table 4: Summary of the performance of different algorithms on the MT.

3.6 Mixed Losses and Reconstruction Analysis

The results of our experimentation with mixed losses is summarized in Fig 17. For the MTL scheme, any encoder-encoder-decoder configuration trained with a single loss function, can be outperformed by some encoder-encoder-decoder configuration with mixed loss functions. This suggests that the improvement in performance created by the MTL approach is more likely due to the diversity in feature maps as a result of the mixture of loss functions, rather than any single particular loss function. Another keep observation is that, for any decoder trained with a particular TLF, the best encoders to use are those trained with a different TLF.

Fig 18 shows the cross-section of various reconstructions. From the cross sections, it is evident that the serial approaches are unable to converge to the ground truth. This is likely due to the aforementioned error accumulation. The MTL and end-to-end reconstructions both show proper convergence towards the ground truth. The performance of these two methods are comparable, though MTL shows better performance in majority of the images.

Fig. 19 shows the absolute error, with respect to pixel brightness of sample reconstructions of both the MTL and end-to-end neural networks. Both networks display an instability, whenever images on contain strong edge effects. This could be an artifact of the Unpooling layers of the network, diluting the the sharp edges when upscaling the activations. This could simply be a limitation of the deep learning approaches, as pixels along the sharp edges

a)

Encoder (SR) \ Decoder (MAE)	Encoder (HDR) MAE	Encoder (HDR) SSIM
MAE		25.1
SSIM	25.0	27.3

Encoder (SR) \ Decoder (SSIM)	Encoder (ITM) MAE	Encoder (ITM) SSIM
MAE	24.7	21.9
SSIM	28.2	24.1

Encoder (SR) \ Decoder (MAE)	Encoder (ITM) MAE	Encoder (ITM) NPCC
MAE	19.2	26.8
NPCC	25.9	27.4

Encoder (SR) \ Decoder (NPCC)	Encoder (ITM) MAE	Encoder (ITM) NPCC
MAE	14.7	15.9
NPCC	12.1	12.7

b)

Encoder (SR) \ Decoder (MAE)	Encoder (ITM) MAE	Encoder (ITM) SSIM
MAE		0.86
SSIM	0.85	0.88

Encoder (SR) \ Decoder (SSIM)	Encoder (ITM) MAE	Encoder (ITM) SSIM
MAE	0.90	0.91
SSIM	0.89	0.89

Encoder (SR) \ Decoder (MAE)	Encoder (ITM) MAE	Encoder (ITM) NPCC
MAE	0.86	0.90
NPCC	0.86	0.85

Encoder (SR) \ Decoder (NPCC)	Encoder (ITM) MAE	Encoder (ITM) NPCC
MAE	0.91	0.91
NPCC	0.91	0.90

c)

Encoder (SR) \ Decoder (MAE)	Encoder (ITM) MAE	Encoder (ITM) SSIM
MAE		0.97
SSIM	0.96	0.98

Encoder (SR) \ Decoder (SSIM)	Encoder (ITM) MAE	Encoder (ITM) SSIM
MAE	0.98	0.98
SSIM	0.98	0.98

Encoder (SR) \ Decoder (MAE)	Encoder (ITM) MAE	Encoder (ITM) NPCC
MAE	0.97	0.99
NPCC	0.97	0.98

Encoder (SR) \ Decoder (NPCC)	Encoder (ITM) MAE	Encoder (ITM) NPCC
MAE	0.90	0.92
NPCC	0.85	0.86

Figure 17: Performance of various loss functions evaluated using (a) PSNR (b) PCC (c) SSIM metrics. In most configurations, varied loss functions (highlighted green) outperforms a single loss function (highlighted red) in the MTL scheme.

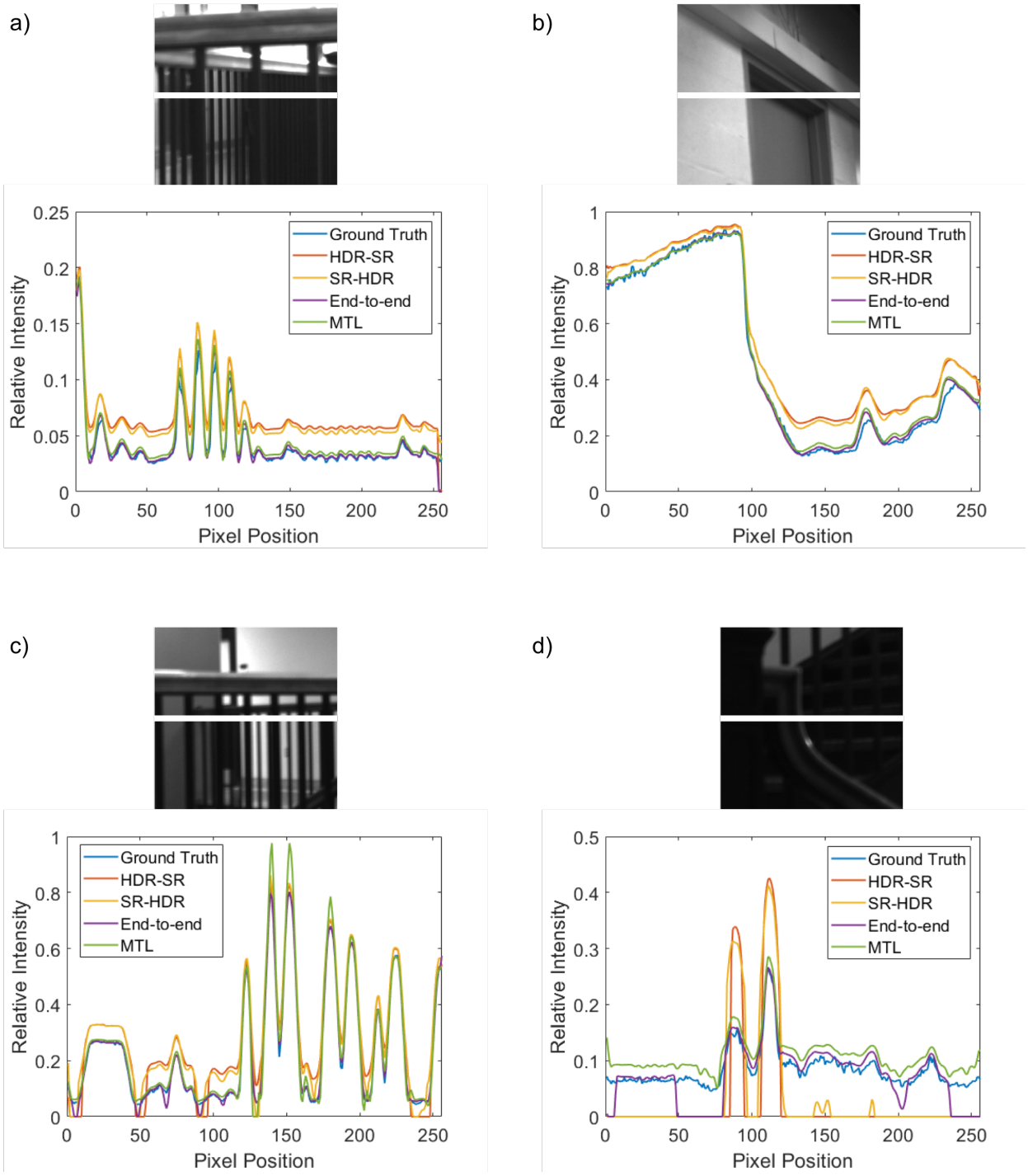

Figure 18: Cross-section analysis of reconstructions produced from the various algorithms. (a)-(d) show different HR-HDR images. The white line shown is the corresponding cross-section being analyzed. Both the MTL and end-to-end track the pixel values closely, with MTL performing better in many cases.

are no longer strongly correlated with their all their neighboring pixels.

Future work can be focused on decomposing the issues of ITM separately. In bright pixels, the main concern is saturation, and the network must use adjacent pixels and the illumination itself to reconstruct lost structure. In dark pixels, the low SNR is the main concern. An MTL approach can be envisioned to tackle these issues separately, and then combined by a decoder network.

3.7 MTL for Training Reduction

For deep learning, particularly in the case of computer vision, the size of datasets required is often prohibitive. The networks are sized for the complexity of the task to be learned. A large network would require a large dataset to train, as enough instances are needed to optimize the several thousands to millions of weights typically involved. This makes large networks more difficult to train, as it requires more time or computation power to converge. There is also the risk of memorizing: if the the dataset is not large and diverse enough, the network, through the course of training, might simply memorize some of the more dominant patterns and examples. MTL reduces the risk of all these issues. It ensures that the tasks are manageable, and that a smaller number of weights are being trained at any stage of the algorithm. It ensure diversity in features being learned, and thus helps in dissuades networks from simply memorizing.

3.8 My Contributions

I led the work in this chapter. Alexander Ivanov and Matthew Yuan constructed and programmed the robot. I contributed in developing the concept, generating and training the networks, and analyzing the data and results.

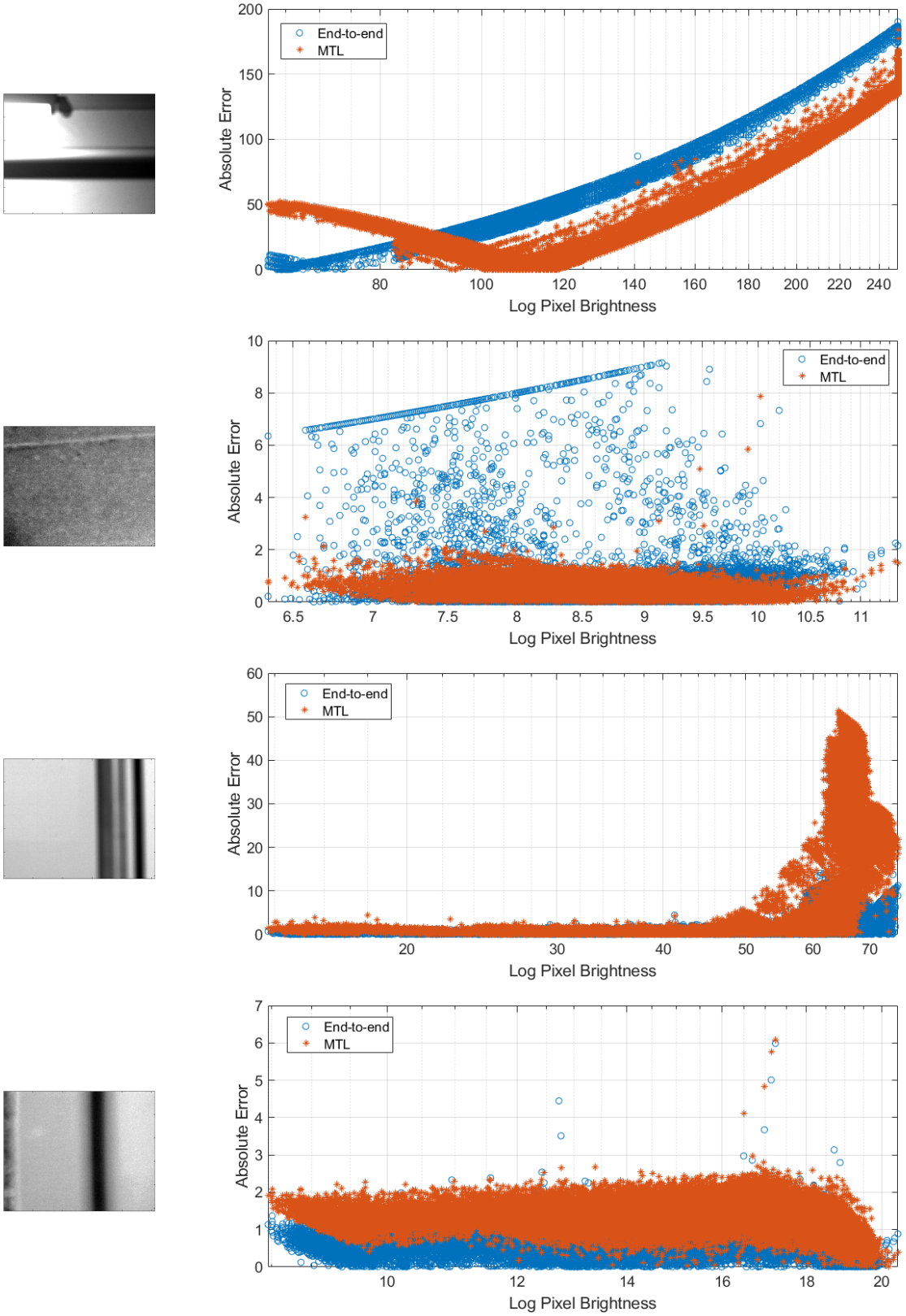

Figure 19: Absolute error with respect to pixel brightness of reconstructions from both MTL and End-to-end schemes. The presence of sharp edges causes higher error in dark and moderate pixels.

4 Conclusion and Future Work

The thesis has studied the use of prior knowledge in the deep learning (DL) algorithms for solving computational imaging (CI) problems. The benefits of DL are various, and it has seen increased widespread use. Yet, the image of DL algorithms as black-box, exposes an underlying view of the methods as not being fully rigorous or fully understood. The main contributions of thesis are summarized as follows:

- Proposed and experimentally demonstrated (to our knowledge) the first convolutional neural network architecture for quantitative phase imaging in extremely low light conditions.
- Investigated the multi-task learning (MTL) approach for image enhancement in the case of super-resolution and inverse tone mapping.
- Investigated the use of different loss functions for sub-tasks in the MTL training scheme.

Beyond the scope of this work, there are several problems that remain open and are worthy of future investigation.

- In this thesis, we focused on embedding the prior physical knowledge of the imaging system. We focused on knowledge of the forward operator, as well as the collection operator of the system. It is also possible to incorporate physical information of the object domain as well. Some algorithms are already being investigated such as spectral premodulation [7].
- In this thesis, we used the GS approximate as the physics embedding step. Other physical priors may be used to incorporate physics. We have shown that gradient-descent may be a better approximant in the case of tomography [10]
- In this thesis, we used CNN architectures for embedding prior knowledge in a supervised approach. Of course, there are other ML algorithms that can be used to operate

on object domain priors. Autoencoders focus greatly on compressed representation of object. Unsupervised learning may also be used to more directly focus on object domain features. All these methods are possible ways to encoder object or data priors, but were beyond the scope of this thesis work.

A Supplementary Materials of Physical Priors

Supplementary Materials for the Section 2 can be found here [9]. The supplementary materials include photon count calibration, SLM calibration, and simulation results of the phase retrieval problem.

B MTL Loss Function Parameters

Evaluation of Various Training Loss Functions. We analyzed the effectiveness of different loss functions. We selected three main loss functions for this case: mean absolute error (MAE), negative pearson correlation coefficient (NPCC), and structural similarity index SSIM). MAE is a classical TLF used in many optimization problems. We have used NPCC very successfully in the past [9,10], as we showed it to resolve salient visual features, better than MAE.

$$MAE(A,B) = \frac{1}{NM}\Sigma_i\Sigma_j|A_{ij} - B_{ij}| \tag{11}$$

$$NPCC(A,B) = \frac{-\Sigma_i(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\Sigma_i(A_i - \bar{A})^2(B_i - \bar{B})^2}} \tag{12}$$

$$SSIM(A,B) = [l(A,B)]^\alpha \times [c(A,B)]^\beta \times [s(A,B)]^\gamma \tag{13}$$

where

$$l(A,B) = \frac{2\mu_A\mu_B + C_1}{\mu_A^2 + \mu_B^2 + C_1}, c(A,B) = \frac{2\sigma_A\sigma_B + C_2}{\sigma_A^2 + \sigma_B^2 + C_2}, s(A,B) = \frac{\sigma_{AB} + C_3}{\sigma_A\sigma_B + C_3} \tag{14}$$

where μ_A, μ_B, σ_A, σ_B, and σ_{AB} are the local means, standard deviations, and cross-covariance for the images A and B. $C_3 = C_2/2$ and the exponents α, β and γ are chosen to be 1.

NPCC Rescaling. The reconstructions from the NPCC trained decoder are off by a scale factor. This is because the NPCC loss function is scale invariant, i.e. $NPCC(A,B) = NPCC(aA,bB)$ for some constants a, b. To correct this effect, we can use the validation data set to find the correction factor. We used the reconstructions from the validation set, and used a linear model to fit the reconstructions with the ground truth.

SSIM Constants. For this evaluation, we used a standard $L = 2^{10}$ for HDR images, and 2^8 for SR images. Naturally, the DNN output does not generate outputs with strict discretization, so the choice of this L may be derived more from empirical expectation, rather than some initial selection; $C_2 = (0.03 \times L)^2$ and $C_1 = (0.01 \times L)^2$.

	PSNR	PCC	SSIM
MTL (MAE-decoder)	26.8	**0.99**	**0.99**
End-to-end (MAE)	**27.1**	0.86	0.98
MTL (NPCC-decoder)	**15.9**	**0.91**	**0.92**
End-to-end (NPCC)	10.9	**0.91**	**0.92**
MTL (SSIM-decoder)	**28.2**	0.89	**0.98**
End-to-end (SSIM)	19.3	**0.90**	0.98

Table 5: Comparison of MTL decoders and end-to-end networks trained with MAE, NPCC, and SSIM loss functions.

References

[1] A. N. Tikhonov, "On the solution of ill-posed problems and the method of regularization," *Dokl. Akad. Nauk. SSSR*, vol. 151, 1963.

[2] A. N. Tikhonov, "On the regularization of ill-posed problems," *Dokl. Akad. Nauk. SSSR*, vol. 153, no. 1, 1963.

[3] M. Bertero and P. Boccacci, "Introduction to inverse problems in imaging," 1998.

[4] J. Mait, G. W. Euliss, and R. A. Athale, "Computational imaging," *Adv. Opt. Photon*, vol. 10, 2018.

[5] N. Wiener and E. Hopf, "Uber eine klasse singulaerer integralgeichungen," *Sitzungsber. Preuss. Akad. Math.-Phys. KI.*, vol. 31, 1931.

[6] S. Saha, "A comprehensive guide to convolutional neural networks the eli5 way," Dec 2018.

[7] M. Deng, S. Li, A. Goy, I. Kang, and G. Barbastathis, "Learning to synthesize: robust phase retrieval at low photon counts," *Light Sci Appl*, vol. 9, no. 36, 2020.

[8] Y. Rivenson, Y. Zhang, H. Günaydin, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," *Light: Sci. Appl.*, vol. 7, p. 17141, 2018.

[9] A. Goy, K. Arthur, Shuai Li, and G. Barbastathis, "Low photon count phase retrieval using deep learning," *Phys. Rev. Lett.*, vol. 121, no. 24, p. 243902, 2018.

[10] A. Goy, G. Rughoobur, Shuai Li, K. Arthur, A. Akinwande, and G. Barbastathis, "High-resolution limited-angle phase tomography of dense layered objects using deep neural networks," *Proc. Nat. Acad. Sci.*, (accepted) 2019.

[11] T. C. Nguyen, V. Bui, and G. Nehmetallah, "Computational optical tomography using 3-d deep convolutional neural networks," *Optical Engineering*, vol. 57, pp. 57 – 57 – 11, 2018.

[12] M. Lyu, W. Wang, H. Wang, H. Wang, G. Li, N. Chen, and G. Situ, "Deep-learning-based Ghost imaging," *Sci. Rep.*, vol. 7, p. 17865, dec 2017.

[13] S. Li, G. Barbastathis, and A. Goy, "Analysis of phase-extraction neural network (phenn) performance for lensless quantitative phase imaging," in *Quantitative Phase Imaging V*, vol. 10887, p. 108870T, International Society for Optics and Photonics, 2019.

[14] M. Lyu, H. Wang, G. Li, and G. Situ, "Learning-based lensless imaging through optically thick scattering media." in preparation, 2018.

[15] A. Sinha, Justin Lee, Shuai Li, and G. Barbastathis, "Lensless computational imaging through deep learning," *Optica*, vol. 4, pp. 1117–1125, 2017.

[16] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, "Diffusercam: lensless single-exposure 3d imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.

[17] G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," *Optica*, vol. 6, no. 8, pp. 921–943, 2019.

[18] S. Kojouharov *Becoming Human: Artificial Intelligence Magazine [Online]. Available: https://becominghuman.ai/*, 2017.

[19] N. Streibl, "Phase imaging by the transport equation of intensity," *Optics Communications*, vol. 49, no. 1, pp. 6 – 10, 1984.

[20] R. W. Gerchberg, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972.

[21] J. R. Fienup, "Reconstruction of an object from the modulus of its fourier transform," *Opt. Lett.*, vol. 3, pp. 27–29, Jul 1978.

[22] W. Xu, M. H. Jericho, I. A. Meinertzhagen, and H. J. Kreuzer, "Digital in-line holography for biological applications," *Proceedings of the National Academy of Sciences*, vol. 98, no. 20, pp. 11301–11305, 2001.

[23] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009.

[24] Y. LeCun, C. Cortes, and C. J. Burges, "MNIST handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[25] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[26] S. Li, M. Deng, J. Lee, A. Sinha, and G. Barbastathis, "Imaging through glass diffusers using densely connected convolutional networks," *Optica*, vol. 5, pp. 803–813, Jul 2018.

[27] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, (New York, NY, USA), p. 160–167, Association for Computing Machinery, 2008.

[28] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8599–8603, 2013.

[29] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively multitask networks for drug discovery," 2015.

[30] Chao Dong, Chen Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[31] J. Johnson, A. Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision (ECCV) / Lecture Notes on Computer Science* (B. Leide, J. Matas, N. Sebe, and M. Welling, eds.), vol. 9906, pp. 694–711, 2016.

[32] Chao Dong, Chen Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional neural network for image super-resolution," in *European Conference on Computer Vision (ECCV), Part IV / Lecture Notes on Computer Science*, vol. 8692, pp. 184–199, 2014.

[33] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang, "Single-image super-resolution: a benchmark," in *European Conference on Computer Vision (ECCV) / Lecture Notes on Computer Science* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), vol. 8692, pp. 372–386, 2014.

[34] Jianchao Yang, J. Wright, T. S. Huang, and Yi Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Proc.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[35] G. Eilertsen, J. Kronander, G. Denes, R. Mantiuk, and J. Unger, "Hdr image reconstruction from a single exposure using deep cnns," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, 2017.

[36] S. Y. Kim, J. Oh, and M. Kim, "Deep sr-itm: Joint learning of super-resolution and inverse tone-mapping for 4k uhd hdr applications," 2019.

[37] T. M. Inc., *makehdr, Create high dynamic range image.* Natick, Massachusetts, United States, 2020.

[38] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.