ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

JOINT CALIBRATION AND RECONSTRUCTION FOR FOCAL PLANE ARRAY IMAGING

M.Sc. THESIS

Muhammet Umut BAHÇECİ

Department of Electronics and Communication Engineering

Electronics Engineering Programme

JUNE 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

JOINT CALIBRATION AND RECONSTRUCTION FOR FOCAL PLANE ARRAY IMAGING

M.Sc. THESIS

Muhammet Umut BAHÇECİ
(504211227)

Department of Electronics and Communication Engineering

Electronics Engineering Programme

Thesis Advisor: Prof. Dr. Ender Mete EKŞİOĞLU

JUNE 2023

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

ODAK DÜZLEMİ DİZİSİ GÖRÜNTÜLEME İÇİN BİRLEŞİK KALİBRASYON VE GERİÇATIM

YÜKSEK LİSANS TEZİ

Muhammet Umut BAHÇECİ
(504211227)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Ender Mete EKŞİOĞLU

HAZİRAN 2023

Muhammet Umut BAHÇECİ, a M.Sc. student of ITU Graduate School student ID 504211227 successfully defended the thesis entitled "JOINT CALIBRATION AND RECONSTRUCTION FOR FOCAL PLANE ARRAY IMAGING", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Ender Mete EKŞİOĞLU** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Işın ERER** Istanbul Technical University	.....
	Prof. Dr. Koray KAYABOL Gebze Technical University	.....

Date of Submission : **04 May 2023**
Date of Defense : **14 June 2023**

To my family,

FOREWORD

I would like to express my sincere gratitude to my thesis advisor, Prof. Dr. Ender Mete Ekşioğlu, for his invaluable guidance and support throughout my research. His expertise and encouragement have been instrumental in helping me to navigate the complexities of my thesis work.

I am grateful to my thesis committee members, Prof. Dr. Işın Erer and Prof. Dr. Koray Kayabol, for their valuable feedback and expertise in reviewing my thesis. I appreciate their commitment to this process and their efforts to help me succeed in my academic pursuits.

I would like to extend my thanks to Aselsan, my employer, for providing me with the opportunity to conduct my thesis work in the company. Their support and resources have been invaluable in facilitating my research, and I am grateful for the chance to apply my academic knowledge in a professional setting. Additionally, I want to thank Dr. Can Barış Top, my manager, for his continuous support and guidance throughout this process. I also want to express my gratitude to Dr. Alper Güngör, my team leader and project manager, for leading my work in the company and providing valuable feedback. Finally, I would like to thank Berkan Kılıç, Sefa Karaca, Evren Uysal, Burak Alptuğ Yılmaz, Ali Alper Özaslan, and Dr. Çağrı Çetintepe for their support and for creating a friendly work environment in the office.

Lastly, I would like to thank my family for their unwavering support throughout my life. I am grateful for the love and encouragement provided by my parents and sisters, who have always been there for me. Their constant belief in me has been a source of motivation and strength. I would also like to express my gratitude to my cat, Herkül, who has been a constant source of joy and companionship during my long hours of work. His presence has always lifted my spirits and helped me stay focused on my goals.

June 2023

Muhammet Umut BAHÇECİ
(Research Engineer)

TABLE OF CONTENTS

ABBREVIATIONS

ADC : Analog-to-Digital Converter
ADMM : Alternating Direction Method of Multipliers
ADR : Airy Disc Radius
CCD : Charge-Coupled Device
CMOS : Complementary Metal-Oxide Semiconductor
CNN : Convolutional Neural Network
CS : Compressive Sensing
DCT : Discrete Cosine Transform
DL : Deep Learning
DMD : Digital Micromirror Device
DSR : Down-Sample Ratio
E2E : End-to-End
FPA : Focal Plane Array
HR : High Resolution
JCnR : Joint Calibration and Reconstruction
LC-SLM : Liquid Crystal Spatial Light Modulator
LED : Light Emitting Diode
LIP : Linear Inverse Problem
LPIPS : Learned Perceptual Image Patch Similarity
LR : Low Resolution
LS : Least Squares
MAP : Maximum-A-Posteriori
MLP : Multilayer Perceptron
MSE : Mean Square Error
nF : Number of Features
nL : Number of Layers
PDF : Probability Density Function
PnP : Plug-and-Play
PSF : Point Spread Function
pSNR : peak Signal-to-Noise Ratio
ReLU : Rectified Linear Units
RL : Richardson-Lucy
SLM : Spatial Light Modulator
SPI : Single Pixel Imaging
SSIM : Structural Similarity
SVD : Singular Value Decomposition
SWIR : Short-Wave Infrared

SYMBOLS

Ge **:** Germanium
InGaAs **:** Indium Gallium Arsenic

LIST OF TABLES

LIST OF FIGURES

JOINT CALIBRATION AND RECONSTRUCTION FOR FOCAL PLANE ARRAY IMAGING

SUMMARY

Single Pixel Imaging (SPI) is a technique in which a single photo-detector captures a scene. Modern camera systems usually utilize millions of silicon-based detectors to capture a scene. These silicon-based sensors have high sensitivity only in the visible band. Outside the visible band, sensors must be produced with more exotic materials. Specifically, for larger wavelengths, germanium (Ge) and indium gallium arsenide (InGaAs) can be shown as examples of these rare materials. These materials are highly expensive and hard to produce as a focal plane array (FPA), which is a 2-dimensional sensor array and is similar to modern camera system's sensor array. Therefore, decreasing the number of sensors is important for non-visible band systems, and the SPI technique provides a system that only utilizes a single detector. Moreover, this system is adopted by many research fields such as terahertz imaging, remote sensing, and medical imaging.

The reconstruction process of the SPI system is an inverse problem and requires the solution of a highly ill-posed inverse problem. To get a unique solution for such an underdetermined system, a regularization term is needed. Compressive sensing (CS) is one of the techniques that use regularization as a prior information function. Moreover, for SPI systems, the CS technique is highly adopted since it can find a solution for a system that has less number of measurements than the number of unknown pixels. The reason is that the CS promotes sparsity in a domain that is incoherent with the measurement domain. Therefore, it provides a nonadaptive sampling procedure under some conditions. These conditions are related to the design process of measurement matrix design and the linear measurement process. If the restricted isometry property (RIP) is met, the entire system has an exact recovery guarantee. In general, it is proven that random measurement matrices are highly probable to satisfy the RIP.

For an SPI system, a spatial light modulator (SLM) encodes a scene with different patterns. As a result, SLM filters can provide random sampling for the SPI system. The spatial resolution of an SPI system is high whereas the temporal resolution is low. This is because a single detector must obtain samples for each pixel, but control units of SLM limit the performance of the system. To get faster measurement rates, an FPA system can be used. In contrast to modern camera sensor arrays, the number of sensors in FPA for the SPI field is very few, but these sensors can provide parallel measurements. As a result, the temporal resolution of these FPA systems is higher than the temporal resolution of an SPI system including only a single photo-detector, and this creates a trade-off between measurement speed and the cost of the system. However, improvements in temporal resolution tolerate the cost disadvantage, and the FPA system is used by various applications.

For an FPA system, a lens placed between the SLM encoder and sensor array becomes more effective in low-resolution FPA measurements. These effects are called airy disc effects and blurry FPA measurements are obtained. The airy disc effects happen because the airy disc does not fit into a single detector pixel, and other detectors around the detector make undesirable measurements and the FPA measurement becomes blurred. This effect arises due to the optical structure of the lens and requires calibration to reduce its effects. In this thesis, an online calibration technique is proposed to reduce the blur in FPA measurements. In the literature, there are proposed offline calibration techniques. One of them relies on getting a measurement for each pixel of a scene. This makes it a long and difficult process to form a calibration matrix. The reason is that even small temperature changes affect noise in detectors, and some pixels may be needed to be repeated to get healthy measurements. The other technique relies on CS, but it also needs to form a large calibration matrix after some repetitive measurements. There is also a deconvolution technique with an exact point spread function (PSF) information used to deblur FPA images. The PSF information cannot be easily obtained and the technique requires iterative calculations to find calibrated results. To overcome these limitations, an online deep learning-based (DL) design is proposed and it does not require long operations to define a calibration system and to implement it in an FPA system.

The recovery process usually involves the solution of an inverse problem through optimization-based approaches. These approaches improve image quality by iteratively utilizing prior information on the image while enforcing data consistency. While various prior information terms such as sparsity in some transformation domains can be used for improved image quality, recent approaches adopt learned prior information terms by changing the related step in the optimization approach with a pre-trained DL-based denoiser which is called the plug-and-play (PnP)-based approach. Within the context of this thesis, we opted to use an alternating direction method of multipliers (ADMM)-based approach since it is a popular choice for CS reconstruction. In contrast to previous techniques that decouple the prior information from the problem model, this thesis involves the integration of the conventional iterative ADMM-based method into a DL framework, which is known as algorithm unrolling. Consequently, a convolutional neural network (CNN) is trained to acquire prior information. The utilization of the unrolling technique enables the acquisition of images in fewer amount of iterations when compared with the conventional iterative approaches. The thesis entails the development of models for varying numbers of measurement scenarios, followed by a comparative analysis of image quality metrics and visuals for both PnP ADMM and unrolling-based techniques.

Finally, the thesis presents a joint design where the online calibration model and the unrolling ADMM module are combined and used together. In this joint model, the parameters of the prior information and calibration model are learned using a unified error function obtained by designing a joint objective function between both corrected FPA measurements and high-resolution model outputs. Different numbers of measurement scenarios are examined for each defined airy disc range. As a result, it is shown that the joint design both reduces the airy disc effects and obtains well-reconstructed images.

Three distinct methodologies have been employed to conduct numerical investigations. The first part solely consisted of comparisons with alternative calibration techniques for the calibration model. To do this, the average peak signal-to-noise ratio (pSNR) is compared for each defined airy disc radius range and the performance improvement brought by the model taking the SLM filter information as input is shown. In the second part, airy disc effects are disregarded and only detector noise is taken into account. The study demonstrates that the unrolling-based algorithm obtains comparable performance to conventional PnP ADMM methods across various image quality metrics, including pSNR, structural similarity index (SSIM), and learned perceptual image patch similarity (LPIPS). In addition, execution time analyses showed that the unrolling-based ADMM method achieved the same quantitative performance in much smaller time frames. In the final section, comparisons are made for the joint model, a PnP ADMM technique, and an unrolling-based reconstruction algorithm. Both the PnP ADMM and the unrolling-based reconstruction algorithm take calibrated FPA measurements from the calibration model. For each airy disc range, the joint model achieves the best metric results. Moreover, the differences between the joint model and others are increased as the effects of the airy disc are increased.

ODAK DÜZLEMİ DİZİSİ GÖRÜNTÜLEME İÇİN BİRLEŞİK KALİBRASYON VE GERİÇATIM

ÖZET

Tek piksel görüntüleme (SPI), örnekleme kısmında sadece bir piksel dedektörün kullanıldığı bir yöntemdir. Bu yöntemde, günümüz modern kamera sistemlerinden farklı olarak sahnenin her bir pikseli için ayrı ayrı ölçümler alınmaktadır. Son yıllarda popülerlik kazanmasıyla yüksek çözünürlüklü görüntü elde etmek için iki boyutlu sensör dizileri yerine kullanmaya başlanmıştır. Bunun nedeni, daha hızlı ve daha az donanım gerektirmesidir. Ayrıca, optik sistemin boyutunun küçülmesini sağlarken tek bir sensörden oluştuğu için de daha az enerji tüketmekte ve daha düşük maliyetli sistemlerin oluşturulmasına imkan vermektedir.

Günümüz sistemleri görünür bant için silikon tabanlı çözümleri temel olarak kullanmaktadır. Silikon endüstrisinde yaşanan bu gelişmelerle beraber yüksek çözünürlüklü görüntüleri milyonlarca dedektörün bulunduğu yapılarla ucuz ve hızlı bir şekilde elde edilebilmektedir. Ancak görünür bandın dışında silikon tabanlı çözümlerin duyarlılığı çok düşük seviyelere inmektedir. Bu duyarlılık, dalgaboyuna göre bir maddenin vermiş olduğu değişimi ifade etmektedir ve görünür bant dışında yüksek dalgaboylarında silikon yerine daha egzotik malzemelere ihtiyaç duyulmaktadır. Bu malzemelerden, germanyum (Ge) ve Endiyum Gallium Arsenid (InGaAs) gibi yarı iletken bileşik yapılarına ihtiyaç duyulmaktadır. Bu yapıların duyarlılığı, görünür bandın üzerindeki dalgaboylarında silikon tabanlı yöntemlerden çok daha yüksektir ve kızılötesi çalışmalarında kullanılabilmektedir. Ancak bahsedilen materyaller, nadir şekilde bulunduğundan ve üretiminin de zor olmasından dolayı günümüz kamera sistemlerindeki gibi milyonlarca dedektörün bir arada kullanıldığı gibi kullanılamazlar. Bu nedenle de SPI sisteminde olduğu gibi sadece bir dedektörün bulunduğu bir görüntüleme sistemi büyük önem taşımaktadır. Bu avantajı sebebiyle SPI sistemi, tıbbi görüntüleme, uzaktan algılama, endüstriyel görüntüleme ve diğer pek çok alanda kullanılmaktadır.

Elde edilen görüntülerden yüksek-çözünürlüklü görüntünün elde edilmesi ters problemlerle mümkündür. Genellikle bu yöntemler, kötü tanımlı problemler olmaktadır. Kötü tanımlı problem, sonuca ulaşmak için gerekli olan bilgilerin yetersizliğini ifade etmek için kullanılır. Bu tür problemlerde, sonuca ulaşmak için bazı varsayımların ve kısıtların yapılması gerekmektedir. Böylece eksik bilginin olması halinde oluşan sonsuz sayıdaki çözümden tek bir benzersiz sonucun elde edilebilmesi mümkün hale gelmektedir. Bir düzenlileştirme terimiyle eksik veya yetersiz bilgi açığı kapatılabilir. Bayesian yaklaşımına göre de bu eksik bilgi orijinal sinyalin bir ön bilgisi şeklinde ifade edilebilir.

Düzenlileştirme terimi için ihtiyaç duyulan bilgi, ön bilgi şeklinde ifade edilebilir. Ancak bu bilginin matematiksel olarak ifade edilmesi her zaman mümkün olmayabilir.

Bu tarz kötü tanımlı problemler için önerilen yöntemlerden biri sıkıştırılmış algılama (CS) tekniğidir. Bu yöntemde, az tanımlı (underdetermined) bir sistem için bulunan ölçüm sayısının bilinmeyen sayısından küçük olmasından dolayı kötü tanımlı problem olarak ifade edilmektedir. Bu yöntem ile bir uzaya göre seyrek çözüm aranmaktadır ve ihtiyaç duyulan bilginin daha az sayıda ölçüm ile tam bir şekilde geri kazanımı sağlanmaktadır. Diğer sinyal işleme yöntemlerinden farklı olarak CS, sinyale göre bir ölçüm modeli izlememektedir. Bu özelliği nedeniyle de ölçüm sistemi doğrusal olmayan bir yapıya sahiptir. Ölçüm sistemi sonucunda boyut indirgeme işlemi de yaşanmaktadır. Ancak, CS yöntemi ölçümlerden kendi ölçüm matrisi sayesinde bilgi kaybetmeksizin bir seyrek çözümün bulunabileceğini ifade etmektedir.

Bir SPI sisteminde yer alan uzamsal ışık modülatörü (SLM) yapılarıyla sahne ile filtrelerin eleman bazlı çarpımı gerçekleştirilerek sahneler farklı SLM filtreleri için kodlanabilmektedir. SLM filtreleri, elektronik veya fiziksel şekilde olabilmektedir. Bu filtrelerin sahneyi geçirip geçirmemelerine göre sahneyi kodlaması mümkün olduğu gibi üzerlerindeki açıklığın oranı ise elde edilecek ölçümün sinyal gücüne etki etmektedir ve bu açıklı oranı da SPI görüntüleme problemi için bir değişkenlik getirebilmektedir. SLM filtrelerinin hızlı değişebilmesi sayesinde farklı ölçümler alınabilmekte ve SLM filtreleri, rastgele dağılımlar halinde tanımlabilmektedir. Böylece, CS tekniğinde ölçüm matrisi, sınırlı izometri özelliğini (RIP) doğal olarak sağlaması mümkün hale gelmektedir. Ölçüm matrisin bu özelliği sağlamasıyla, orijinal sinyalin kayıpsız bir şekilde sıkıştırılması ve düşük boyutlu bir uzaya taşınmasına imkan verir.

Bir dedektörün bulunduğu sistemde uzamsal olarak çözünürlük yeterli olsa da zamansal çözünürlük açısından kısıtlara takılmaktadır. Burada, sistemde kullanılan SLM filtre değiştirmesinde kullanılan kontrolcünün hızı ölçüm hızını kısıtlamaktadır. Bu sorunu ortadan kaldırmak için iki boyutlu bir sensör dizisi (FPA) kullanımına geçilebileceği söylenmiştir. Klasik modern görüntüleme sistemlerinin aksine burada kullanılan sensör sayıları, sahne çözünürlüğüne göre çok düşük seviyelerdedir ve sahnenin farklı yerlerinden aynı anda birden çok ölçüm alınması sağlanarak ölçüm hızı arttırılabilir. Böylece, zamansal çözünürlük iyileştirilebilir. Bu FPA görüntüleme sistemiyle maliyetlerde artışlar gözlenmektedir. Ancak getirdiği zamansal çözünürlük açısından iyileştirmeler bu dezavantajı kapatmaktadır.

FPA sisteminde, birden çok dedektör kullanılmasından ötürü SLM filtreleriyle kodlanmış görüntünün bir lens vasıtasıyla iki boyutlu sensör dizisine düşürülmesinde lens etkileri öne çıkar. Bu etkiler, airy disk etkileri olarak bilinen bir bulanıklaşmaya neden olur. Tek dedektör sisteminin aksine bu sistemde airy disk, tek bir dedektör pikseline sığmamasıyla beraber dedektörün çevresinde bulunan diğer dedektörler istenmeyen ölçümler yapar ve görüntünün bulanıklaştığı gözlenir. Bu etki, lensin optik yapısı nedeniyle ortaya çıkmaktadır ve etkilerinin azaltılması için kalibrasyon işlemine ihtiyaç duymaktadır. Bu tez kapsamında, bahsedilen etkiyi kaldırmak için bir çevrimiçi kalibrasyon tekniği geliştirilmiştir. Literatürde yer alan diğer teknikler çevrimdışı bir yapıda kalibrasyon şemasına sahiptir. Çevrimdışı kalibrasyon işlemlerinde her sahne pikseli için ölçüm alınması gerekliliği ortaya çıkmaktadır. Bu durumda, kalibrasyon matrisini elde etmek uzun ve zor bir sürece neden olmaktadır. Bunun nedeni, ufak sıcaklık değişimleri bile dedektörler üzerindeki gürültünün değişimine

neden olabilmektedir ve böylece bazı ölçümlerin tekrar tekrar alınması gerekliliğini doğurmaktadır. Bir başka teknikte CS teknikleri kullanarak ihtiyaç duyulan ölçüm sayısı azaltılmış olsa da sürecin zorluğundan tekrar tekrar ölçümlerin alınması ve elde edilen kalibrasyon matrisinin büyük boyutlarda olması nedeniyle karmaşık bir süreç olduğu söylenebilir. Bu yöntemlerin dışında kesin nokta yayılım fonksiyonu (PSF) kullanıldığı bir ters çözümleme (deconvolution) problemine de yer verilmiştir. Önerilen çevrimiçi derin öğrenme (DL) tabanlı yöntemin uzun kalibrasyon işlemleri gerektirmeden farklı büyüklüklerdeki airy disk etkilerini ortadan kaldırabileceği gösterilmiştir.

Geriçatım tarafında FPA için tak ve kullan (PnP) yöntemini içeren alternatif yönlü çoklu çarpanlar (ADMM) tekniği, CS geriçatımında popülerlik kazanmıştır. Bu yöntemlerde, ön bilgi yerine farklı PnP yöntemleriyle hızlı ve doğru sonuçlar elde edilebilmektedir. Bu tezde, klasik iteratif ADMM tekniği açılarak (unrolling) bir DL uygulaması halinde gerçeklenmiştir. Böylece, ön bilgi, bir konvolüsyonel sinir ağı (CNN) ile öğrenilmiştir. Bu unrolling yöntemiyle, klasik şekilde gerçeklenen iteratif yöntemin aksine daha az sayıda iterasyonda sonuca gidebilmek mümkündür. Yöntemin çıktısı, gerçek sahne görüntüsüyle kıyaslanarak bir hata hesabı yapılır ve geriye doğru model katsayılarını güncellenir. Unrolling model çıktısı, ön bilginin tanımlanmasında kullanılmaktadır. Tezde farklı sayıda ölçüm sayısı senaryoları için modeller oluşturulmuş ve bilinen PnP ADMM yöntemleriyle görüntü kalitesi metrikleri ve görselleri üzerinden karşılaştırmalarına yer verilmiştir.

Son olarak tezde, çevrimiçi kalibrasyon modelini ve unrolling ADMM modulünün uçuca birleştirilip kullanıldığı bir tasarıma yer verilmiştir. Bu birleşik (joint) modelde; airy disk etkileriyle bozulmuş ölçümler, kullanılan SLM filtreleri, ileri yönlü matris gösterimi ve veri doğruluğu (data fidelity) ifadesi için gerekli kısıt ifadeleri giriş değerleri olarak alınmaktadır. Teknikte, hem düzeltilmiş düşük çözünürlüklü ölçümlerin hem de yüksek çözünürlüklü model çıktısı arasında bir optimizasyon problemi yazılarak elde edilen birleşik hata fonksiyonuyla geriye doğru ön bilginin ve kalibrasyon modelinin parametreleri öğretilmektedir. Tanımlanmış her airy disk aralığı için farklı sayıda ölçüm senaryoları incelenmiş ve eğitilmiştir. Böylece, airy disk etkilerini azaltan hızlı bir geriçatım modeli elde edilmiştir.

Numerik çalışmalar üç farklı şekilde ele alınmıştır. İlk başlıkta, sadece kalibrasyon modelinin diğer kalibrasyon yöntemleriyle kıyaslamalarına yer verilmiştir. Burada, her bir tanımlı airy disk yarıçap aralığı için elde edilen ortalama tepe sinyal-gürültü oranı (pSNR) karşılaştırmaları yapılmış ve modelin SLM filtre bilgisini girişinde almasının getirdiği performans artışı resmedilmiştir. Ayrıca, modellerin uyumsuz airy disk yarıçap aralıkları dışında kullanılmasıyla gözlenebilecek performans değişimlerine yer verilmiştir. İkinci başlıkta, airy disk etkileri ihmal edilerek sadece dedektör gürültülerinin bulunduğu durum için geriçatım performansları farklı görüntü boyutları ve farklı ölçüm sayıları için karşılaştırılmıştır. Unrolling tabanlı ADMM yönteminin en az klasik PnP ADMM yöntemleri kadar performanslar elde ettiği farklı görüntü kalitesi metrikleriyle; pSNR, yapısal benzerlik endeksi (SSIM) ve öğrenilmiş algısal görüntü parça benzerliği (LPIPS) gösterilmiştir. Ayrıca, yapılan yürütme süresi analizleriyle unrolling tabanlı ADMM yöntemin bu performanslara çok daha küçük sürelerde ulaştığı gösterilmiştir. Son başlıkta, airy disk etkileri de dahil edilerek

birleşik modelin, kalibrasyon modeliyle kalibre edilmiş girdiler alan bir PnP ADMM ve unrolling geriçatım algoritmasıyla performans kıyasları yapılmıştır. Tanımlı her bir airy disk aralığı için yürütülen çalışmalarda birleşik modelin en iyi metrik sonuçlarına eriştiği ve airy disk etkilerinin büyüdüğü koşullarda performans farkları oluştuğu görülmüştür.

1. INTRODUCTION

The technique of Single Pixel Imaging (SPI) has gained considerable attention in recent times owing to its capacity to acquire high-resolution images utilizing only a single detector element, such as a photodiode [1]. In contrast to conventional imaging techniques that necessitate extensive pixel arrays for image formation, SPI employs a configurable spatial light modulator (SLM) that encodes a scene with a sequence of structured illumination patterns. Subsequently, the aforementioned patterns are quantified by a single detector component, and by means of a sequence of mathematical calculations, a representation of the entity is reconstructed.

Silicon-based sensors such as charge-coupled devices (CCD) and complementary metal-oxide semiconductors (CMOS) are highly used in modern imaging systems, which have plenty of detectors to get high-resolution (HR) images. Since the developments in the silicon industry, these devices bring cheap and high-quality solutions for visible band systems. On the other hand, for the outside of the visible band, silicon-based systems do not work as in the visible band. The responsivity of silicon is lower than some other exotic materials such as indium gallium arsenic (InGaAs) and germanium (Ge). Especially, for the short-wave infrared (SWIR) band, these rare materials are critical to capture images [2]. Moreover, the SWIR spectrum is crucial to penetrate fog or smoke scatterings [3]. For another application, gas leaks can be detected by using SWIR band [4]. However, these rare materials are highly expensive even for use in military applications. In fact, getting a reasonable resolution with these materials are several order magnitudes more expensive than silicon-based systems.

It might seem ineffective to only have one pixel when modern cameras have millions of them. However, using an SPI system has some advantages. The system can respond the light intensity changes even if the change is so small [5]. The overall system requires less-demanding storage and processing units since a single detector is used to capture

the data. This brings some advantages to systems that need to work at low data rates such as remote sensing systems [6]. A single detector can not be as expensive as a sensor array although the detector is made of exotic materials for non-visible bands. Therefore, for infrared imaging, a low-cost system is possible [7]. The SPI system is also used for different areas such as 3-dimensional imaging [8], object tracking [9], and terahertz imaging [10].

With the development of compressive sensing (CS) [11,12], SPI systems have obtained a nonadaptive sampling strategy. In this strategy, the measurement process is not related to the desired signal. Therefore, the measurement system design is independent of the original signal. Furthermore, the number of measurements is less than the number of pixels, and the utilization of a random measurement system results in an underdetermined system. However, the promotion of sparsity in this field allows for the finding of a distinct solution with a reduced number of measurements [13]. Therefore, an SPI system with a CS reconstruction process can get a better mean square error (MSE) than other scanning methods such as raster scan and basic scan in the same amount of time [14].

The SPI system can be extended into a small sensor array, which is known as a focal plane array (FPA). The main idea of using a single detector is to reduce the cost of systems for the outside of the visible band. The size of an FPA system is still dramatically lower than a modern camera detector system. In addition, this sensor array increases the resolution of the reconstruction while it increases a bit the cost and complexity of the system. However, outcomes from the system can tolerate these drawbacks. The reason is that an FPA system can get higher measurement rates than a single detector [15]. As a result, in the same amount of time, obtainable resolution increases.

Improvements in temporal resolution bring some difficulties to the physical model of the FPA system. The main problem is lens effects on the FPA detectors due to the lens' physical nature. This effect causes an adjacent detector reads an unintended measure due to a focusing problem [16]. Since focusing on the FPA detectors are not good, measurements are not well-focused results, and performances obtained

using super-resolution techniques are drastically decreased. In literature, calibration systems are proposed to reduce the effects of optical aberrations. One of the known techniques requires full-size matrix and matrix-matrix production. This technique is known as point-scan calibration [17]. For a scene with the size of $M \times N$, it is needed to repeat for MN times. As a result, the point-scan calibration is time-consuming, slow, and vulnerable to temperature changes when measurements are obtained. Since it is an offline technique, for each new system, this process must be repeated to get a calibration matrix. The second technique requires fewer measurements than the point-scan calibration technique. This technique uses a CS method to reduce the number of needed measurements [18]. As such, the time to build the calibration matrix is shortened. However, it has no other advantage compared to the point-scan calibration method. In this design, as in the point-scan calibration technique, the pixels of the SLM must be controllable, which is also called an electronically controllable modulator. It also does not allow changing calibration matrices online. Therefore the calibration matrix still needs large matrix multiplications, and for different system setups, the calibration procedure must be repeated.

In this thesis, first, an online calibration technique is proposed to calibrate lens-corrupted FPA measurements. The proposed design is a faster method than other techniques in both the generation of the calibration system and the assembly of the calibration function. Moreover, it does not require electronic control for the SLM part since other techniques rely on electronic modulators. The second part of the thesis is related to a super-resolution algorithm that relies on the combination of CS and learning-based models. For this technique, a conventional CS reconstruction algorithm is unrolled, and prior information in the optimization problem is not defined as a hand-crafted function or a function that can be modeled as a mathematical representation. By using the unrolled network, a loss function is obtained to improve the performance of the prior function since it is defined as a convolutional neural network (CNN). As a result, the model also learns a prior function while it produces a reconstruction result for FPA measurements in the training step. After that, the best model is investigated after each epoch. This unrolled network combines both a classical CS application and deep learning (DL)-based approach to get a more powerful

and faster technique than well-known classical plug-and-play (PnP) systems. Lastly, both calibration and unrolling-based reconstruction process is integrated to get a bigger joint model. This joint method uses both low-resolution (LR) calibrated images and HR reconstructed images to compute a combined loss, which provides the model to learn the path of both the calibration and reconstruction processes. Therefore, the joint model can generate better reconstruction results in terms of image quality metrics when lens effects are seen on FPA measurements.

The rest of this thesis is organized as follows. Chapter 2 expresses the basics of an FPA imaging system. To do this, first, the signal model from the scene to measurement is explained. Then, image reconstruction techniques are investigated. In this part, The general optimization problem is explained in the Bayesian approach. After that, the CS technique is given for an iterative method called as Alternating Direction Method of Multipliers (ADMM). It is also shown how a PnP image denoiser can be used in ADMM steps. Chapter 3 explains the DL-based online calibration technique. To make results concrete, the model outputs are compared with various models and a deconvolution method. In Chapter 4, the unrolling-based technique is introduced. Then, the joint model is constructed as a general model. Model performances are analyzed in image quality metrics and timing analysis. Chapter 5 ends with some final thoughts and a discussion of possible directions for the future.

2. BASICS OF FOCAL PLANE ARRAY IMAGING

2.1 Signal Model

SPI is achieved through the use of a specialized optical computer architecture that consists of several key components, including an SLM, two lenses, a single photon detector, and an analog-to-digital converter (ADC). The first lens is responsible for projecting a scene onto the SLM. Then, the SLM performs random linear measurements of the scene, which allows the system to capture compressed measurements of the object or scene. These compressed measurements can then be used to reconstruct an image of the object or scene using algorithms based on CS. The second lens is used to focus encoded HR images into a single photon detector. After that, the detector measures the scattered light, and the ADC converts these signals into a digital system that can be processed by digital systems. Overall, SPI provides an effective way to capture and process visual information using only a single detector element [14].

In contrast to SPI, FPA imaging requires an array of individual detectors to capture visual information. While FPA imaging systems can provide higher spatial and temporal resolutions, they also tend to be more expensive to design and implement due to the need for multiple detector elements [15]. Figure 2.1 visualizes a single-pixel detector and an array of single-pixel detectors for FPA. The FPA system can be defined by a reference detector like a reference sensor assumption in a phased array system. Since many single-pixel detectors are gathered to make a 2-dimensional array, the design cost is also increased. However, this structure enables the system to run at high measurement rates due to taking several measurements at the same time.

The FPA imaging system can be represented in Figure 2.2.

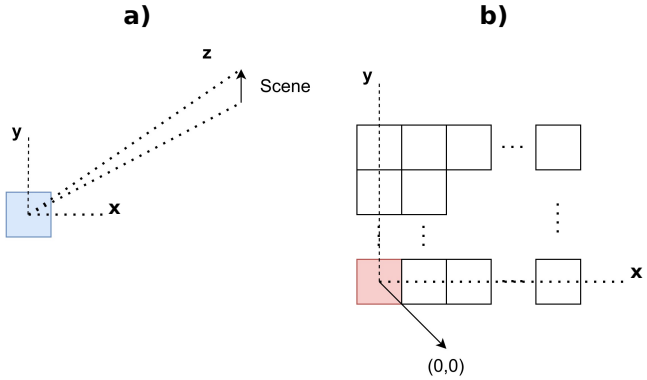

Figure 2.1 : Representations of SPI and FPA detectors. **a)** A single-pixel detector is for the SPI system. **b)** A 2-dimensional array of single-pixel detectors is for the FPA system.

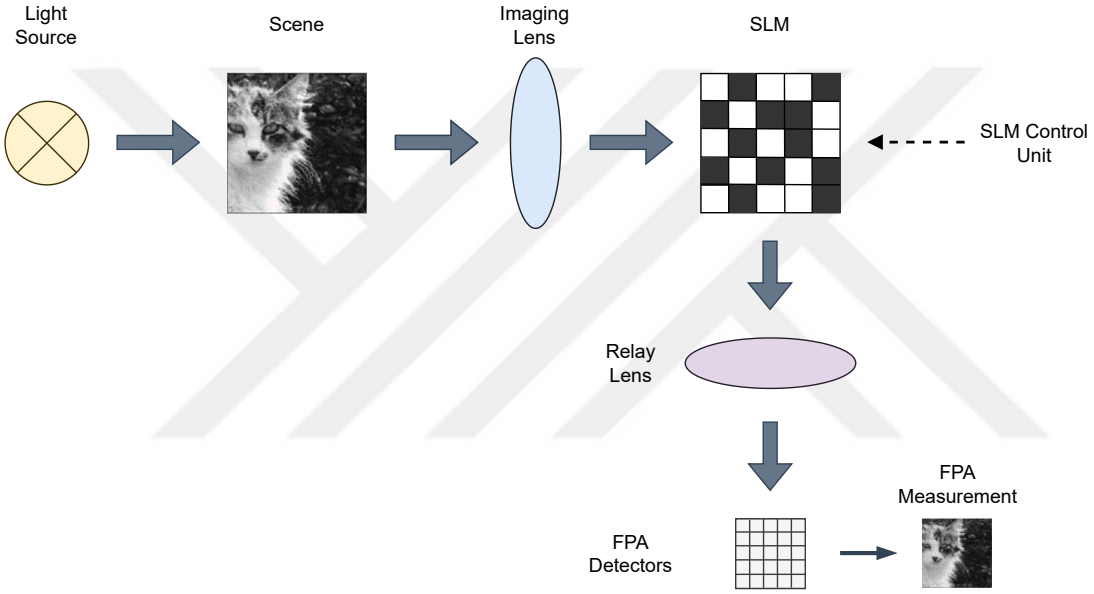

Figure 2.2 : Diagram of FPA imaging system.

The light source illuminates the scene. Then, a lens called an imaging lens is used to focus on the SLM. The SLM modulates the scene, and this results in an encoded version of the scene. The SLM Control Unit allows for the acquisition of various scene versions. These versions define different snapshots of the same scene. For SPI systems, SLM is usually an electronically controllable device, which is known as Digital Micromirror Device (DMD) [19]. This device has multiple micro-sized mirrors, and each mirror can be positioned. This process can be handled quickly. In fact, its speed can reach approximately 20 kHz. The device has three states, which are $+12^\circ$ for the on position, -12° for the off position, and 0° for the powered down

position. By changing the positions of mirrors in DMD, different DMD modulation scenes can be created.

For modulation schemes, there are also other designs. One of them is pseudothermal modulation. In this modulation, a light source produces a laser beam, and the beam passes through a rotating glass diffuser [20]. In another technique, a liquid crystal SLM (LC-SLM) is used with random phase masks to generate pseudothermal light beams [21]. For high-speed demands, a technique includes light-emitting diode (LED) arrays and is accomplished by combining the symmetry of the utilized Hadamard basis set with the very quick switching time of the LEDs [22]. For another modulation design, a physical modulation filter is designed on the glass, then the glass is motioned by a piezo-stage device. This device has high accuracy for dimensions, and with a small change, the scene is modulated by a new pattern [23].

After the modulation step in Figure 2.2, another lens known as the relay lens is used. This lens focuses the modulated image onto FPA detectors. Before sampling and quantizations, the last modulated HR scene is seen at the relay lens. Then, FPA detectors sample the light intensity that is observed. After all, for a snapshot, a 2-dimensional measurement is obtained from the ADC. Figure 2.3 shows how a detector in FPA measures.

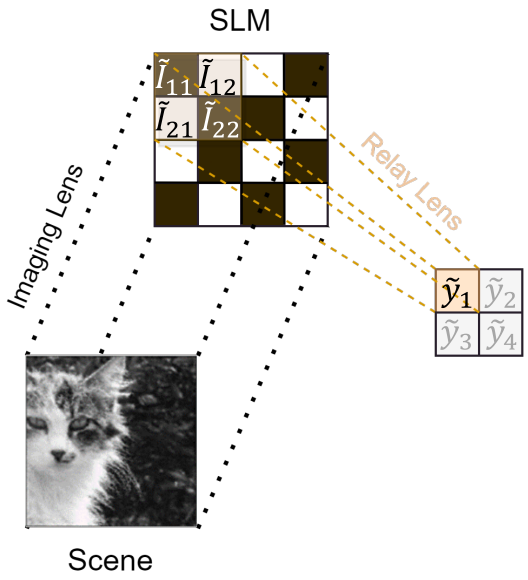

Figure 2.3 : Measurement of a detector in FPA.

The scene is focused onto the SLM by an imaging lens. Since the example includes 4×4 HR image and SLM, and an FPA system with the size of 2×2, each 2×2 pixels area is used to find an average result for a detector in FPA. The size of this area is defined by a down-sample ratio, and both scene and FPA detector size determine the ratio. In noise-free assumption, the detector in FPA samples $\tilde{\boldsymbol{y}}_1$, and it can be formulated as:

$$\tilde{\boldsymbol{y}}_1 = \frac{1}{4} \sum_{m=1}^{2} \sum_{n=1}^{2} \tilde{\boldsymbol{I}}_{mn}, \tag{2.1}$$

where $\tilde{\boldsymbol{I}}_{mn}$ defines the multiplication of pixels mn of both the scene and SLM. In other words, the scene and SLM are element-wise multiplied. Then, the pixel mn is chosen for $\tilde{\boldsymbol{I}}_{mn}$. In more general form of Eq. (2.1), $\tilde{\boldsymbol{y}}_1$ includes also multiplicative noise [24], and it can be rewritten as:

$$\tilde{\boldsymbol{y}}_1 = \frac{1}{4} \sum_{m=1}^{2} \sum_{n=1}^{2} \tilde{\boldsymbol{I}}_{mn} + \tilde{\boldsymbol{n}}_1, \tag{2.2}$$

where $\tilde{\boldsymbol{n}}_1$ can be assumed that noise on the measurement detector is dominant. With the reading of capacitive load, the FPA measurement gets also an undesired signal that is somehow multiplied version of the original signal.

In Figure 2.4, the scenario given in Figure 2.3 is used, and it shows how subareas are shaped as vectors.

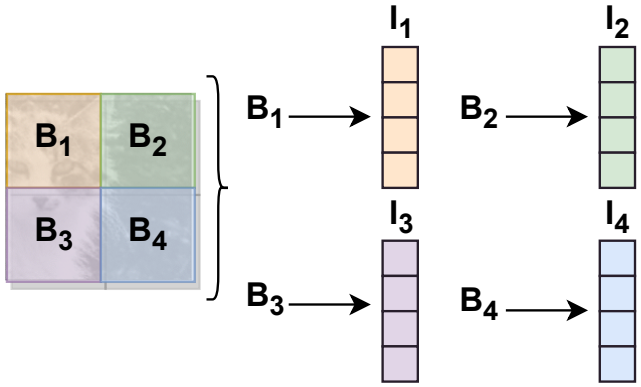

Figure 2.4 : A vector form representation for 4×4 scene and a 2×2 down-sample ratio.

The number of subareas for a scene is directly related to the FPA detector size. Each subarea is a sub-image of the scene and is represented by $\boldsymbol{B}_i$. Then, each $\boldsymbol{B}_i$ is reshaped as a vector $\boldsymbol{I}_i \in \mathbb{R}^{N_p \times 1}$, where N_p represents the number of pixels in down-sampled

area. For a 2×2 sub-image, the vector becomes 4×1. This is a reshaping procedure before using several snapshots. In Figure 2.5, a multiple snapshot experiment is represented for 3 samples.

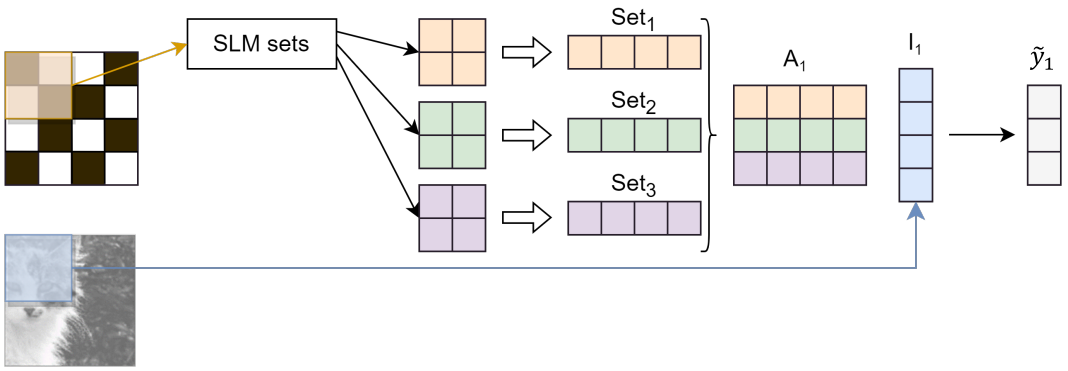

Figure 2.5 : A multiple measurement experiment of the FPA system for 3 samples.

Similar to the vectorization of sub-images, each SLM for the same subarea is also reshaped, but it is shaped as a raw vector. Then, all snapshots are sorted as row vectors of a matrix $\boldsymbol{A} \in \mathbb{R}^{N_s \times N_p}$. For the example in Figure 2.5, a down-sampled area is 2×2 and $N_s = 3$. As a result, $\boldsymbol{A}$ is the element of $\mathbb{R}^{3\times 4}$. Figure 2.5 refers $\boldsymbol{A}$ as $\boldsymbol{A}_1$ to point out that this matrix is shaped for the first sub-image. In Figure 2.4, different sub-images are given, and the number of sub-images is N_f and is determined by the number of FPA detectors. The overall system including modulation and down-sampling operations for a detector in FPA can be rewritten as:

$$\tilde{\boldsymbol{y}}_1 = \boldsymbol{A}_1 \boldsymbol{I}_1 + \boldsymbol{n}_1, \tag{2.3}$$

where $\tilde{\boldsymbol{y}}_1 \in \mathbb{R}^{N_s \times 1}$. Thus, a general formulation for given example in Figure 2.5 can be written as:

$$\tilde{\boldsymbol{y}}_i = \boldsymbol{A}_i \boldsymbol{I}_i + \boldsymbol{n}_i, \; i = 1, ..., N_f. \tag{2.4}$$

When all sub-images are written as in Eq. (2.4), a block diagonal representation is obtained as [25]:

$$\begin{bmatrix} \tilde{\boldsymbol{y}}_1 \\ \tilde{\boldsymbol{y}}_2 \\ \vdots \\ \tilde{\boldsymbol{y}}_{N_f} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{A}_{N_f} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_1 \\ \boldsymbol{I}_2 \\ \vdots \\ \boldsymbol{I}_{N_f} \end{bmatrix} + \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2 \\ \vdots \\ \boldsymbol{n}_{N_f} \end{bmatrix}. \tag{2.5}$$

The block diagonal part in Eq. (2.5) can also be represented by $\boldsymbol{A}_B$, and whole equation can be simplified as:

$$\boldsymbol{y}_c = \boldsymbol{A}_B \boldsymbol{x}_c + \boldsymbol{n}_c, \tag{2.6}$$

where $\boldsymbol{y}_c \in \mathbb{R}^{N_s N_f \times 1}$ and $\boldsymbol{A}_B \in \mathbb{R}^{N_s N_f \times N_p N_f}$. The desired $\boldsymbol{x}_c$ is in a vector form with the size of $\mathbf{R}^{H_{HR} W_{HR}} \times 1$. For reconstruction processes, $\boldsymbol{x}_c$ is shaped into a matrix form with the size of $H_{HR} \times W_{HR}$ by a function. The inverse of this function can be used to change the matrix form into a vector form because it is also invertible. If a single snapshot is used, the output of the FPA sensors can be named as $\boldsymbol{y}_i$ for snapshot i.

It is assumed that lens effects are perfect for the purposes of the signal model representations shown above. Yet, due to the presence of lenses in the system, optical aberrations occur, causing objects to appear blurry. Diffraction can occur in an optical system regardless of how well it is constructed for the object's radiant energy distribution and image size. This is because of the inherent nature of the system. When light travels through a space that is too narrow for its wavelength, a phenomenon known as interference of light waves takes place. It results in a dulling of the image's sharpness. While traveling through a big aperture, light beams almost travel in plate waves. On the other hand, if the aperture is quite narrow, the light beams will interact with one another. In addition, at certain locations either they cancel each other out or they might increase the amount of light that is present in certain areas [26]. This results in the formation of a diffraction pattern, which, when viewed in a two-dimensional representation, seems to be a collection of rings and is referred to as the "airy disc effect" [27]. Nevertheless, these effects can be controlled by proper designs. It is not possible to see the airy disc if its size is less than a pixel. The test image for this case has a high quality in terms of sharpness. When the airy disc continues to expand its size, it will, at some point, also cover the pixels that are near it. As a result of this circumstance, the sharpness is diminished.

The primary objective of the first lens, shown as the imaging lens in Figure 2.2, is to bring the scene into sharp focus on the SLM. If the scene is represented by $\boldsymbol{x}$, and an SLM filter for snapshot i is represented by $\boldsymbol{\Lambda}_i$, the focusing process can be expressed as:

$$\boldsymbol{x}_{encoded,i} = (\boldsymbol{h}_1 \circledast \boldsymbol{x}) \odot \boldsymbol{\Lambda}_i, \tag{2.7}$$

where $\boldsymbol{h}_1$ represents the imaging lens effects. Then, a convolution operation implies that the lens effects are observed on the scene when it is focused on $\boldsymbol{\Lambda}_i$. The $\boldsymbol{x}_{encoded,i}$ represents the output after the SLM operation. Then, the second lens, shown as the relay lens in Figure 2.2, is used. The aim of using this lens is to focus an encoded image on the FPA sensors. Therefore, it affects directly the FPA measurement performance. If down-sampling operation is represented by a function $\mathrm{D}(.)$, the FPA measurement for snapshot i can be written as:

$$\boldsymbol{y}_i = \mathrm{D}(\boldsymbol{h}_2 \circledast \boldsymbol{x}_{encoded,i}) + \boldsymbol{n}_i, \tag{2.8}$$

where $\boldsymbol{h}_2$ represents the effects of the relay lens. Since the relay lens has more effect on the FPA measurement than the imaging lens [28], the effects of the imaging lens can be neglected. Therefore, Eq. (2.8) can be simplified as:

$$\boldsymbol{y}_i = \mathrm{D}(\boldsymbol{h} \circledast (\boldsymbol{x} \odot \boldsymbol{\Lambda}_i)) + \boldsymbol{n}_i, \tag{2.9}$$

where $\boldsymbol{h}$ is the $\boldsymbol{h}_2$ for the relay lens effect. The relay lens creates an airy disc effect on the FPA sensors. An airy disc radius (ADR) represents the radius of the central disc in the airy disc effect, and it can be formulated as:

$$ADR = 1.22\frac{\lambda f}{D}, \tag{2.10}$$

where λ is the wavelength of the operated light, and f is the focal length of the lens. The focal length is a way to measure how strongly the system brings a light together or spreads it out. Also, D stands for the size of the lens's aperture. Since f/D describes the *f-number*, Eq. (2.10) can also be thought of as a function of the *f-number*. When the *f-number* is low, the aperture widens, and the lens lets in more light. On the other hand, if the *f-number* is high, the aperture is small and the lens's diffraction effects are more noticeable.

When focusing the image, the relay lens generates blurring due to its small aperture and circular shape. The reason for this is that the airy disc does not fit in an FPA resolution pixel. In a design sense, the maximum resolution is reached at the last place where the airy disc fits. Nevertheless, ADR varies and influences the size of the airy disc at different wavelengths. As a result, if airy disc effects are not wanted, each application can operate at a distinct *f-number* for different wavelengths.

This blurring is often modeled as a jinc function [29]. The jinc function causes circular effects in the frequency space and acts as a low-frequency filter. The jinc function is often used interchangeably with the airy disc function because of its shape. As a result, the airy disc effect causes the focused energy to be spread to adjacent detectors, and unwanted data to be read in them. This effect can be modeled as point spread functions (PSF), and in Figure 2.6, multiple ADRs are plotted by using [30]. A PSF is found for each ADR. After that, the logarithm of the PSF is plotted. To demonstrate variations across PSFs, they are clipped in the same range, which is $[-6, 0]$ in logarithms.

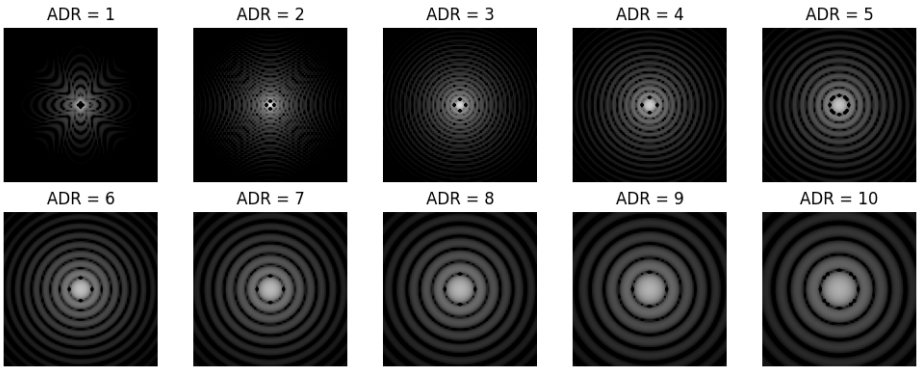

Figure 2.6 : Relation between ADR and PSF.

In Eq. (2.10), focal length and lens diameter are parameters of the physical imaging system and are not changed after a setup of a system. The wavelength may also vary depending on the sensitivity of the lens and the detector and can be accepted as a nearly fixed value in the working environment. For this reason, the $\boldsymbol{h}$ for PSF can be modeled as an airy disc under operating conditions and its behavior does not change largely.

2.2 Image Reconstruction

The concepts of image reconstruction and inverse problems are closely related in signal processing and image analysis [31]. Image reconstruction involves generating a digital image from a set of measurements or observations obtained from a physical system, while inverse problems refer to estimating the input of a system from its output [32]. Image reconstruction can be viewed as a type of inverse problem, where the goal is to recover an image from a set of measurements related to the image with a mathematical model. In various fields, such as medical imaging [33] and remote sensing [34],

inverse problems and image reconstruction are fundamental. It is therefore crucial to understand the fundamental principles of inverse problems and their relevance in image reconstruction.

Typical image reconstruction problems can be modeled as Eq. (2.6). In here, noisy measurement is represented as $\boldsymbol{y}_c$, desired data $\boldsymbol{x}_c$, the matrix $\boldsymbol{A}_B$ of the forward model, and $\boldsymbol{n}_c$ for noise. The picture $\boldsymbol{x}$ is converted into a vector $\boldsymbol{x}_c$ for common representation, and it includes all pixels of the image $\boldsymbol{x}$ in vector form. In general, the matrix $\boldsymbol{A}_B$ represents linear operations and is the expression of acquiring measurements from a system.

The challenge of estimating the desired unknown from noisy observations is known as the linear inverse problem (LIP) since these mathematical expressions are linear equations [35]. Usually, this estimation problem is an ill-posed problem. Ill-posed linear inverse problems are a type of mathematical problem that arises when attempting to recover an unknown quantity from a set of measurements that are related to it by a linear operator. These problems can be difficult to solve because of the way the measurements are related to the unknown quantity. The inverse problem can be underdetermined because the measurements may be noisy, incomplete, or indirect. This indicates that there are alternative solutions to the problem that could fit the data equally well. Solving these problems can be difficult because of these factors. Finding a solution that is stable and dependable while also being able to take into account the various sources of uncertainty is the primary goal in the process of solving an ill-posed LIP.

Finding the least squares (LS) regression is the typical strategy for inverse problems [36]. It is, however, ill-posed when the number of variables exceeds the number of observations and has an infinite number of solutions. In other words, there is no unique solution to the problem. To overcome this, a regularization term is required in order to obtain a unique solution. This is also known as the prior model of the desired data.

One of the inverse problem techniques that utilize prior information for underdetermined systems is CS. The goal of CS is to reconstruct a signal from a minimal number

of linear measurements [12]. The fundamental idea underlying CS is that many signals in nature are sparse, which means they may be represented by only a few significant coefficients on a suitable basis. For a traditional technique, measurements are obtained according to the Nyquist rate. Then, a large number of samples are compressed to get small useful amounts. In other words, most of the data is redundant. For example, in JPEG [37], Discrete Cosine Transform (DCT) is used to find coefficients. Then, a small significant portion of these coefficients is necessarily used.

CS, on the other hand, has a unique sampling approach that combines both sampling and data acquisition steps. In more general, it applies a measurement process that reduces dimensions for measurements. Since the dimension of measurements is less than the original data, some information is lost. Moreover, this situation leads to the possibility of infinite solutions. The reason is that the number of equations or measurements is less than the number of unknown variables. The measurement matrix, however, is designed to get a sparse solution. The measurement matrix design is a crucial step in CS recovery. In general, using random matrices guarantee a high probability recovery of a sparse signal for the problem [13]. Since random measurements are reconstructed with a high probability, for an FPA system, using random filter designs in the SLM guarantees the recovery of the scene with a fewer number of measurements. The reason is that, unlike LS, CS theory says that with at least M measurements, which is proportional to $K \times \log(N/K)$, where K is the sparsity factor, and N is the number of pixels in desired data, K-sparse signal can be estimated.

CS technique requires prior information for the model. A variety of research areas utilizes different prior information such as ℓ_1-norm for the direction of arrival estimations [38], TV-norm for image reconstructions [39], and some neural network-based priors for the most recent applications [40]. Since the CS technique has a model-based approach, the Bayesian perspective can be investigated to see the necessity of prior information in the optimization problem. For Eq. (2.6), Bayesian solution can be formulated as:

$$\hat{\boldsymbol{x}}_c = \arg\max_{\boldsymbol{x}_c} \; p(\boldsymbol{x}_c|\boldsymbol{y}_c), \tag{2.11}$$

where $p(\boldsymbol{x}_c|\boldsymbol{y}_c)$ is the posterior probability density function (PDF). To find a solution for $\boldsymbol{x}_c$ is equal to maximum a posteriori probability (MAP) estimation problem [41]. With using the Bayes rule, Eq. (2.11) can be written as:

$$\hat{\boldsymbol{x}}_c = \arg\max_{\boldsymbol{x}_c} \frac{p(\boldsymbol{y}_c|\boldsymbol{x}_c)p(\boldsymbol{x}_c)}{p(\boldsymbol{y}_c)}. \tag{2.12}$$

Since the equation searches a solution $\boldsymbol{x}_c$ to maximize it, $p(\boldsymbol{y}_c)$ part can be ignored, and Eq. (2.12) becomes:

$$\hat{\boldsymbol{x}}_c = \arg\max_{\boldsymbol{x}_c} p(\boldsymbol{y}_c|\boldsymbol{x}_c)p(\boldsymbol{x}_c). \tag{2.13}$$

Then, a logarithm operation is applied to Eq. (2.13) to eliminate multiplication. By using the logarithm property, multiplication in a logarithm operation can be written as a summation of distinct logarithm operations as:

$$\hat{\boldsymbol{x}}_c = \arg\max_{\boldsymbol{x}_c} \log p(\boldsymbol{y}_c|\boldsymbol{x}_c) + \log p(\boldsymbol{x}_c). \tag{2.14}$$

The maximization problem can also be changed into a minimization problem:

$$\hat{\boldsymbol{x}}_c = \arg\min_{\boldsymbol{x}_c} -\log p(\boldsymbol{y}_c|\boldsymbol{x}_c) - \log p(\boldsymbol{x}_c), \tag{2.15}$$

where $p(\boldsymbol{y}_c|\boldsymbol{x}_c)$ is the conditional pdf of measurement $\boldsymbol{y}_c$ given $\boldsymbol{x}_c$, is also known as likelihood function. This function is related to the forward model. On the other hand, $p(\boldsymbol{x}_c)$ is related to the prior information. Equation (2.15) can be rewritten as:

$$\hat{\boldsymbol{x}}_c = \arg\min_{\boldsymbol{x}_c} w(\boldsymbol{y}_c;\boldsymbol{x}_c) + \beta s(\boldsymbol{x}_c), \tag{2.16}$$

where $w(\boldsymbol{y}_c;\boldsymbol{x}_c) = -\log p(\boldsymbol{y}_c|\boldsymbol{x}_c)$ and $s(\boldsymbol{x}_c) = -\log p(\boldsymbol{x}_c)$. There is also an additional regularization coefficient to control the importance of the prior model against the forward model. If $\beta = 1$, Eq. (2.16) is the same as a MAP estimation problem, but with distinct β selections, the problem can be applied in different ways.

For the CS problem, $w(\boldsymbol{y}_c;\boldsymbol{x}_c)$ corresponds to the data fidelity term and is directly related to the representation of the forward model. Furthermore, any prior information can be represented by $s(\boldsymbol{x}_c)$. By using regularization parameters like β, the reconstruction output can be changed.

2.2.1 Alternating direction method of multipliers-based reconstruction

ADMM is an optimization technique that decomposes a convex optimization problem with linear constraints into smaller subproblems [42]. The algorithm then iteratively solves these subproblems by alternating between optimizing the primal variables and the dual variables while ensuring that they remain consistent through a penalty parameter. By breaking down the problem into smaller components, ADMM converges faster than other optimization methods [43]. Furthermore, ADMM is a versatile approach that can handle various types of optimization problems, including those with non-convex or non-differentiable objective functions [44].

The general ADMM optimization problem can be written as in [45]:

$$\arg\min_{\dot{\boldsymbol{u}},\dot{\boldsymbol{v}}} \; f_1(\dot{\boldsymbol{u}}) + f_2(\dot{\boldsymbol{v}}) \text{ subject to } \boldsymbol{G}\dot{\boldsymbol{u}} = \dot{\boldsymbol{v}}, \tag{2.17}$$

where $\dot{\boldsymbol{u}} = \boldsymbol{x}_c$, and in order to separate the forward model and prior information, a new variable $\dot{\boldsymbol{v}}$ is also defined as $\boldsymbol{G}\dot{\boldsymbol{u}}$. The matrix $\boldsymbol{G} = [\boldsymbol{I}\ \boldsymbol{A}_B]^T$ is for $\dot{\boldsymbol{v}} = [\dot{\boldsymbol{v}}_1\ \dot{\boldsymbol{v}}_2]$. As a result, initially, $\dot{\boldsymbol{v}}_1$ is $\boldsymbol{x}_c$, and $\dot{\boldsymbol{v}}_2 = \boldsymbol{A}_B\boldsymbol{x}_c$. Equation (2.17) can be written as an unconstrained augmented lagrangian form:

$$L_\rho(\dot{\boldsymbol{u}}, \dot{\boldsymbol{v}}; \dot{\boldsymbol{d}}) = f_1(\dot{\boldsymbol{u}}) + f_2(\dot{\boldsymbol{v}}) + \dot{\boldsymbol{d}}^T(\dot{\boldsymbol{v}} - \boldsymbol{G}\dot{\boldsymbol{u}}) + \frac{\rho}{2}\|\boldsymbol{G}\dot{\boldsymbol{u}} - \dot{\boldsymbol{v}}\|_2^2, \tag{2.18}$$

where $\dot{\boldsymbol{d}}$ is a dual variable, and ρ is used for penalty term to penalize the problem if the equality constraint in Eq. (2.17) is not satisfied. The penalty term also designates how much penalization is applied to the optimization problem. For each variable, a local optimum is found. This process is applied by minimization operations. Then, to get a general solution, a global optimum is calculated. ADMM solves the following local optimization problems:

$$\dot{\boldsymbol{u}}^{k+1} = \arg\min_{\dot{\boldsymbol{u}}} \; f_1(\dot{\boldsymbol{u}}) + \frac{\rho}{2}\|\boldsymbol{G}\dot{\boldsymbol{u}} - \dot{\boldsymbol{v}}^k - \dot{\boldsymbol{d}}^k\|_2^2, \tag{2.19}$$

$$\dot{\boldsymbol{v}}^{k+1} = \arg\min_{\dot{\boldsymbol{v}}} \; f_2(\dot{\boldsymbol{v}}) + \frac{\rho}{2}\|\boldsymbol{G}\dot{\boldsymbol{u}}^{k+1} - \dot{\boldsymbol{v}} - \dot{\boldsymbol{d}}^k\|_2^2, \tag{2.20}$$

$$\dot{\boldsymbol{d}}^{k+1} = \dot{\boldsymbol{d}}^k - (\boldsymbol{G}\dot{\boldsymbol{u}}^{k+1} - \dot{\boldsymbol{v}}^{k+1}), \tag{2.21}$$

where the superscript k represents the iteration number k. For Eq. (2.19), $f_1(\dot{\boldsymbol{u}})$ is assumed as 0. For Eq. (2.20), $f_2(\dot{\boldsymbol{v}})$ is $PF(\dot{\boldsymbol{v}}_1) + IF(\dot{\boldsymbol{v}}_2)$, where $PF(.)$ is the prior function and $IF(.)$ is the indicator function [45]. The indicator function takes $\dot{\boldsymbol{v}}_2$, $\boldsymbol{y}_c$,

and eps as inputs. If the condition of $\dot{\boldsymbol{v}}_2 \| \dot{\boldsymbol{v}}_2 - \boldsymbol{y}_c \|_2 \leq eps$ is not satisfied in the indicator function, it produces ∞. If the condition is satisfied, it produces 0. After that, Eq. (2.20) can be also split into two parts:

$$\dot{\boldsymbol{v}}_2^{k+1} = \arg\min_{\dot{\boldsymbol{v}}_2} \; IF(\dot{\boldsymbol{v}}_2) + \frac{\rho}{2} \| \boldsymbol{A}_B \dot{\boldsymbol{u}}^{k+1} - \dot{\boldsymbol{v}}_2 - \dot{\boldsymbol{d}}_2^k \|_2^2, \tag{2.22}$$

$$\dot{\boldsymbol{v}}_1^{k+1} = \arg\min_{\dot{\boldsymbol{v}}_1} \; PF(\dot{\boldsymbol{v}}_1) + \frac{\rho}{2} \| \dot{\boldsymbol{u}}^{k+1} - \dot{\boldsymbol{v}}_1 - \dot{\boldsymbol{d}}_1^k \|_2^2, \tag{2.23}$$

where $\dot{\boldsymbol{d}} = [\dot{\boldsymbol{d}}_1 \; \dot{\boldsymbol{d}}_2]^T$. Then, dual variable updates also can be written as:

$$\dot{\boldsymbol{d}}_1^{k+1} = \dot{\boldsymbol{d}}_1^k - \dot{\boldsymbol{u}}^{k+1} + \dot{\boldsymbol{v}}_1^{k+1}, \tag{2.24}$$

$$\dot{\boldsymbol{d}}_2^{k+1} = \dot{\boldsymbol{d}}_2^k - \boldsymbol{A}_B \dot{\boldsymbol{u}}^{k+1} + \dot{\boldsymbol{v}}_2^{k+1}. \tag{2.25}$$

To work with image-sized results, $\dot{\boldsymbol{u}}$, $\dot{\boldsymbol{v}}_1$, and $\dot{\boldsymbol{d}}_1$ can be reshaped as $\boldsymbol{u}$, $\boldsymbol{v}_1$, and $\boldsymbol{d}_1$, respectively. For this purpose, a function named $P(.) : \mathbb{R}^{HW \times 1} \rightarrow \mathbb{R}^{H \times W}$ is used to reshape from a vector to a matrix. For example, if the input of the function is $\dot{\boldsymbol{v}}_1$, the output of the function is $\boldsymbol{v}_1$. This function is also invertible. Thus, $P'(.) : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{HW \times 1}$ reshapes a matrix into a vector. The other variables such as $\dot{\boldsymbol{v}}_2$, $\dot{\boldsymbol{d}}_2$ are the same as $\boldsymbol{v}_2$ and $\boldsymbol{d}_2$, respectively.

For each iteration update, the local solutions can be formulated as:

$$\boldsymbol{u}^{k+1} = P((\boldsymbol{I} + \boldsymbol{A}_B)^{-1}(P'(\boldsymbol{v}_1^k + \boldsymbol{d}_1^k) + \boldsymbol{A}_B^T(\boldsymbol{v}_2^k + \boldsymbol{d}_2^k))), \tag{2.26}$$

$$\boldsymbol{v}_1^{k+1} = \arg\min_{\boldsymbol{v}_1} \; PF(\boldsymbol{v}_1) + \frac{\rho}{2} \| P'(\boldsymbol{v}_1 - (\boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k)) \|_2^2, \tag{2.27}$$

$$\boldsymbol{v}_2^{k+1} = \arg\min_{\boldsymbol{v}_2} \; \frac{\rho}{2} \| \boldsymbol{A}_B P'(\boldsymbol{u}^{k+1}) - \boldsymbol{v}_2 - \boldsymbol{d}_2^k \|_2^2 \text{ subject to } \| \boldsymbol{v}_2 - \boldsymbol{y}_c \|_2 \leq eps, \tag{2.28}$$

$$\boldsymbol{d}_1^{k+1} = \boldsymbol{d}_1^k - (\boldsymbol{u}^{k+1} - \boldsymbol{v}_1^{k+1}), \tag{2.29}$$

$$\boldsymbol{d}_2^{k+1} = \boldsymbol{d}_2^k - (\boldsymbol{A}_B P'(\boldsymbol{u}^{k+1}) - \boldsymbol{v}_2^{k+1}). \tag{2.30}$$

With using any prior functions, Eq. (2.17) can be solved. In fact, it is the same as Eq. (2.16). The ADMM algorithm is capable of yielding a solution for the MAP estimation problem with rapid convergence. Moreover, all updating steps for $\boldsymbol{u}$, $\boldsymbol{v}_1$, and $\boldsymbol{d}_1$ are used as in image-shaped sizes. For updating $\boldsymbol{u}$ in Eq. (2.26), first, image-sized input $\boldsymbol{v}_1^k + \boldsymbol{d}_1^k$ is stacked into a vector form by the inverse of the function $P(.)$. Then, $\dot{\boldsymbol{u}}^{k+1}$ is obtained as in a vector form. For the last step, it is converted to a matrix form by the function $P(.)$. For updating $\boldsymbol{v}_1$, image-sized input $\boldsymbol{v}_1 - (\boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k)$ is reshaped as a

vector by the inverse function of $P(.)$. Then, to update $\boldsymbol{v}_2$ and $\boldsymbol{d}_2$, image-sized $\boldsymbol{u}^{k+1}$ is reshaped to the vector form of it.

2.2.2 Plug-and-play algorithms for compressive sensing

PnP ADMM is a framework that has recently sparked substantial scientific attention for image restoration [46,47]. This method uses denoising and deblurring algorithms to perform image restoration tasks such as super-resolution, inpainting, and compressed sensing [48]. One of the main advantages of this method is that it allows the use of pre-existing denoisers and deblurring algorithms to build a strong picture restoration pipeline. This not only speeds the whole operation but also produces excellent results across a wide range of restoration applications [49,50].

For any type of prior functions, Eq. (2.27) can be used as a general optimization problem. If $\boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k$ is thought as a noisy image, Eq. (2.27) can be rewritten as:

$$\boldsymbol{v}_1^{k+1} = \arg\min_{\boldsymbol{v}_1} \lambda PF(\boldsymbol{v}_1) + \frac{\rho}{2}\|\boldsymbol{v}_1 - \tilde{\boldsymbol{v}}_1^{k+1}\|_F^2, \tag{2.31}$$

where $\tilde{\boldsymbol{v}}_1^{k+1}$ represents the noisy input image for iteration $k+1$, and with λ, the problem controls the effect of the prior model on the solution. It can be given some examples for the selection of a prior function. Besides, since image-shaped variables are used, Frobenius norm,$\|.\|_F$, is used instead of $\|.\|_2$. If the prior is selected as $\|P'(\boldsymbol{v}_1)\|_1$, the problem searches for a sparse solution. Another example of prior selection is $\|\boldsymbol{v}_1\|_{TV}$ to remove artifacts in pictures or movies in this particular case. It can also be used in a combination of multiple priors as a single prior in the problem. For example, both TV-norm and ℓ_1-norm can be used as the prior functions like $PF(\boldsymbol{v}_1) = \alpha_1\|\boldsymbol{v}_1\|_{TV} + \alpha_2\|P'(\boldsymbol{v}_1)\|_1$. Finding suitable α_1 and α_2 values is another optimization-required difficulty in this scenario.

Any proximal operator (or mapping function) can be written as:

$$\text{prox}_{f(.)}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2, \tag{2.32}$$

where $\boldsymbol{z}$ is the input, and the solution $\hat{\boldsymbol{x}}$ is searched. This function is designed to locate the point within the domain of a differentiable function that is in the closest proximity to a specified point within the domain of a non-differentiable function,

based on a certain distance metric. Typically, non-differentiable functions are used to promote sparsity or smoothness such as ℓ_1-norm, TV-norm. If Eq. (2.32) is compared to Eq. (2.31), it is seen that they are the same. Moreover, Eq. (2.31) only depends on the choice of prior and can be interpreted as a denoising operation. The reason is that the difference between $\boldsymbol{v}_1$ and $\tilde{\boldsymbol{v}}_1^{k+1}$ is minimized while prior model is shaped as $PF(.)$ with a regularization parameter λ. As a result, Eq. (2.31) can be interpreted as a denoising problem without considering differentiability, and a denoiser can be selected for the prior function. This leads that any off-the-shelf denoisers can be used as a prior function in Eq. (2.31). If $\sigma = \sqrt{\lambda/\rho}$ is defined, Eq. (2.31) becomes:

$$\boldsymbol{v}_1^{k+1} = \arg\min_{\boldsymbol{v}_1} \; PF(\boldsymbol{v}_1) + \frac{1}{2\sigma^2}\|\boldsymbol{v}_1 - \tilde{\boldsymbol{v}}_1^{k+1}\|_F^2. \tag{2.33}$$

Therefore, Eq. (2.33) can be simply replaced with an off-the-shelf denoiser $D_\sigma(\tilde{\boldsymbol{v}}_1^{k+1})$ to find $\boldsymbol{v}_1^{k+1}$. The distance metric in the problem provides the proximal mapping to get a suitable result for off-the-shelf denoisers.

3. DEEP LEARNING-BASED ONLINE CALIBRATION

3.1 Introduction

An online calibration technique is explained for two DL-based models. Both models take three inputs; FPA measurements corrupted by both detector noise and airy disc effects, high-resolution SLM encoder patterns, and a one-hot input. For the output of each model, corrected FPA measurements are produced.

The method is a flexible calibration technique that can be used offline without requiring FPA detectors to be calibrated. Additionally, the method enables online calibration, which is faster than existing methods and does not require electronic control of SLM filters. This makes the calibration process more efficient and effective, allowing for easier creation and application of the calibration process.

The contents of this chapter rely on [51,52]. In Chapter 3.2, offline calibration techniques are explained as relative works. In Chapter 3.3, the theory of online calibration is given and two methods are introduced. Chapter 3.4 includes the creation of datasets, training procedure, and implementation steps, and a competing method relying on exact PSF knowledge of lens behavior. Chapter 3.5 gives ablation studies and simulation results.

3.2 Related Works

The deviation from the actual system makes the measurement matrix inaccurate and decreases the imaging quality. To overcome this problem, conventional calibration techniques depend on an offline calibration measurement process.

The first conventional technique depends on the creation of a full calibration matrix [17]. In this technique, first of all, the background is made white for calibration and the $\boldsymbol{x}$ image is filled with the value 1. As the size of SLM filter and FPA detectors are $M \times N$ and $P \times Q$, respectively, each SLM pixel is opened and open pixels are

represented by 1 and the rest of them are 0. For each pixel, constructed SLM filter is saved as $\boldsymbol{\Lambda}_i$. In this form, the FPA measurement of the image $\boldsymbol{x}$ is obtained as $\boldsymbol{y}_i$ with the size of $P \times Q$. Then, after reshaping of each $\boldsymbol{y}_i$ to a column vector, a matrix $\boldsymbol{C}$ is obtained as a full calibration matrix. The size of the full calibration matrix is $PQ \times MN$. Since $\boldsymbol{\Lambda}_i$ and $\boldsymbol{x}$ have the same size, $(\boldsymbol{\Lambda}_i \odot \boldsymbol{x})$ can be reshaped as $\boldsymbol{u}_i \in \mathbf{R}^{MN \times 1}$, and the full matrix calibration can be expressed as:

$$\boldsymbol{y}_i = \boldsymbol{C}\boldsymbol{u}_i + \boldsymbol{n}_i,\ i = 1, \ldots, MN, \tag{3.1}$$

where $\boldsymbol{C}$ is used for both calibration and down-sample operation. This process produces a full calibration matrix, but it is time-consuming. For a large image with a size of 1920×1080, the number of required measurements to generate the full calibration matrix is 2457600. Moreover, this technique requires the use of electronically controllable SLM filters like DMD [53]. The reason is that the offline calibration process requires MN repeated measurements. As a result, for high-order demands, DMDs are necessary. The other problem of this technique is related to repetitive measurements. Since many measurements are needed, it requires a long time to form the calibration matrix and temperature change leads to noise fluctuations on FPA detectors. Thus, the measurement process should be repeated for the wrong measurements and it causes the procedure to be difficult and time-consuming.

The second offline calibration technique is based on the CS approach [18]. In this technique, m DMD filters are produced by random binary masks with the size of $M \times N$. A single blank paper is used as the image and the FPA measurements are obtained by detectors. As a result, Eq. (3.1) can be rewritten as:

$$\boldsymbol{y}_i = \boldsymbol{C}diag(\boldsymbol{\Lambda}_i)\boldsymbol{u}_i + \boldsymbol{n}_i,\ i = 1, \ldots, m, \tag{3.2}$$

where $diag(.)$ diagonalizes $\boldsymbol{\Lambda}_i$. Instead of using MN measurements, with using only $m << MN$ measurements, a full calibration matrix can be estimated by CS algorithms. The problem of the CS-based full matrix calibration technique is similar to the previous technique. Temperature changes lead to some measurements being repeated again and DMD is a necessary component of the technique.

3.3 Theory

The proposed online calibration technique does not require any calibration measurement system and it is constructed from the forward model equation. In Chapter 2.1, airy disc effects are given and it is represented by $\boldsymbol{h}$. With this technique, airy disc effects are reduced by model-based DL algorithms. To do this, Eq. (2.9) can be rewritten as:

$$\boldsymbol{y}_i = M(\boldsymbol{h}_B * \boldsymbol{h} * (\boldsymbol{\Lambda}_i \odot \boldsymbol{x}))) + \boldsymbol{n}_i,\ i = 1,..,N_s, \tag{3.3}$$

where $\boldsymbol{h}_B$ represents a moving average filter and is applied on an airy disc-corrupted SLM-encoded image by a convolution operation. Then, the $M(.)$ operation is applied to select the top left pixel of the image that has the same size as true image $\boldsymbol{x}$. Thus, down-sampled measurement $\boldsymbol{y}_i$ is obtained. With using of the commutativity property of the convolution operation, Eq. (3.3) can be written as:

$$\boldsymbol{y}_i = M(\boldsymbol{h} * \boldsymbol{h}_B * (\boldsymbol{\Lambda}_i \odot \boldsymbol{x}))) + \boldsymbol{n}_i,\ i = 1,..,N_s. \tag{3.4}$$

The $M(.)$ operation can be formulated in a convolution operation like $*_{strided}$. Equation (3.4) can also be written as:

$$\boldsymbol{y}_i = \boldsymbol{h} *_{strided} (\boldsymbol{h}_B * (\boldsymbol{\Lambda}_i \odot \boldsymbol{x})) + \boldsymbol{n}_i,\ i = 1,..,N_s. \tag{3.5}$$

If airy disc effects are not seen in measurements and detector noise is neglected, the ideal FPA measurements are obtained and can be formulated as:

$$\boldsymbol{y}_{ideal,i} = \boldsymbol{\delta} *_{strided} (\boldsymbol{h}_B * (\boldsymbol{\Lambda}_i \odot \boldsymbol{x})),\ i = 1,..,N_s, \tag{3.6}$$

where $\boldsymbol{\delta}$ is the delta-dirac impulse function. The main goal of the calibration process is obtaining the $\boldsymbol{y}_{ideal,i}$ for each measurement i. To achieve this, a deconvolution operation is required to remove the effects of $\boldsymbol{h}$. However, deconvolution operation is an ill-posed inverse problem. As a result, to deconvolve airy disc-corrupted image, using of prior information is important [54]. The optimization problem can be formulated as:

$$\arg\min_{\boldsymbol{z}_i}\ f(M(\boldsymbol{z}_i)) \text{ subject to } \|M(\boldsymbol{h} * \boldsymbol{z}_i) - \boldsymbol{y}_i\|_2 \leq \varepsilon,\ i = 1,...,N_s, \tag{3.7}$$

where $\boldsymbol{z}_i = \boldsymbol{h}_B * (\boldsymbol{\Lambda}_i \odot x)$ and $M(\boldsymbol{z}_i)$ is the interested output of the problem for measurement i. In Eq. (3.7), $f(.)$ indicates the prior information and $\boldsymbol{\Lambda}$ is compulsory input of it. For this reason, a calibration technique needs SLM filters as input.

3.3.1 Strided SLM calibration network

Corrupted FPA measurements, corresponding SLM filters, and one-hot encoded ADR range values are used as network inputs. ADR ranges vary for each simulation and effects the performance of the network. When the ADR range is increased, the airy disk effect becomes stronger, and FPA measurements are seen as more blurry. Using all ADR ranges in a model loss results in a dominance of higher ADR range losses, while obtained test performances suffer for lower ADR ranges. As a result, ADR ranges can be encoded like the one in Table 3.1. Each ADR range is simulated for a corresponding uniform distribution.

Table 3.1 : One-hot values for ADR ranges.

ADR Range	One-hot Code
1.5-2.5	100000000
2.5-3.5	010000000
3.5-4.5	001000000
4.5-5.5	000100000
5.5-6.5	000010000
6.5-7.5	000001000
7.5-8.5	000000100
8.5-9.5	000000010
9.5-10.5	000000001

Figure 3.1 shows the architecture of strided SLM network architecture. The ADR path represents ADR range selection and is taken as a one-hot code. According to Table 3.1, the ADR path takes 9-width signal and creates a signal that is as long as the length of combined channels after both the SLM path and the FPA path. Since the number of channels is represented as N_c for both SLM and FPA paths, the ADR path produces a signal of length $2N_c$. This process can be formulated as:

$$\boldsymbol{o}_{ADR,i} = \boldsymbol{M}_{ADR}\bar{\boldsymbol{o}}_i, \tag{3.8}$$

where $\boldsymbol{M}_{ADR} \in \mathbb{R}^{2N_c \times 9}$ represents the Multilayer Perceptron (MLP) structure and $\bar{\boldsymbol{o}}_i \in \mathbb{R}^{9\times 1}$ is a one-hot encoded input of the network. Hence, $\boldsymbol{o}_{ADR,i}$ is the output

of the ADR path for measurement i. This signal is used to divide each corresponding channel of the concatenated outputs of both SLM and FPA routes.

FPA path takes an input signal with the size of $1 \times H_{LR} \times W_{LR}$. The single channel input has the same size as the down-sampled true image and $H_{LR} \times W_{LR}$ defines it. A single FPA measurement is convoluted with 3×3 kernels, and a batch normalization operation is applied to get a more stable training process [55]. Then an activation function is applied to learn complex problems and it is selected as Rectified Linear Units (ReLU) in this design [56]. A block of operations including convolution kernels, batch normalization, and ReLU can be seen as the default block operation with 3×3 kernels and 1×1 stride. When $\boldsymbol{y}_i$ is the input of the default block operation of the FPA path for measurement i, the output of the default block operation can be written as:

$$\bar{\boldsymbol{y}}_{i,1} = \sigma(\mathrm{BN}(H_{1,1}(\boldsymbol{y}_i))), \tag{3.9}$$

where $H_{1,1}(.)$ represents CNN features of the first default block operation of the FPA path, $\mathrm{BN}(.)$ is for batch normalization, and $\sigma(.)$ is the activation function. The size of the output in Eq. (3.9) is $N_c \times H_{LR} \times W_{LR}$. The number of channels is increased by using N_c CNN features with 1×1 stride in the default block operation. Before the concatenation process, this default block operation is repeated as:

$$\bar{\boldsymbol{y}}_{i,2} = \sigma(\mathrm{BN}(H_{1,2}(\bar{\boldsymbol{y}}_{i,1}))), \tag{3.10}$$

where $H_{1,2}(.)$ represents N_c CNN features for the second default block operation of the FPA path.

In the SLM path, a single channel encoder filter is used, and it has the same size as the HR image, which is $H_{HR} \times W_{HR}$. The output of the first default block operation of the SLM path can be written as:

$$\bar{\boldsymbol{\Lambda}}_{i,1} = \sigma(\mathrm{BN}(H_{2,1}(\boldsymbol{\Lambda}_i))), \tag{3.11}$$

where $H_{2,1}(.)$ represents N_c CNN features for the first default block operation of the SLM path. Thus, the size of $\bar{\boldsymbol{\Lambda}}_{i,1}$ is $N_c \times H_{HR} \times W_{HR}$. For the second default block operation in the SLM path, the image size should be reduced to the size of the LR image. If the down-sample ratio in both dimensions is defined as $\frac{H_{HR}}{H_{LR}} \times \frac{W_{HR}}{W_{LR}}$, the size

of stride operation is directly chosen as down-sample ratio. The output of the SLM path can be written as:

$$\bar{\mathbf{\Lambda}}_{i,2} = \sigma(\mathrm{BN}(H_{2,2}(\bar{\mathbf{\Lambda}}_{i,1}))). \tag{3.12}$$

The size of $\bar{\mathbf{\Lambda}}_{i,2}$ becomes $N_c \times H_{LR} \times W_{LR}$ after N_c CNN features in the second default block operation of the SLM path.

After the concatenation of both the SLM and the FPA path outputs, the ADR path normalizes this concatenated output in the channel direction as:

$$\bar{\boldsymbol{c}}_{i,1} = \mathrm{CONCAT}(\bar{\boldsymbol{y}}_{i,2}, \bar{\mathbf{\Lambda}}_{i,2}) / \boldsymbol{o}_{ADR,i}, \tag{3.13}$$

where $\mathrm{CONCAT}(.)$ represents the concatenation operation in channel direction, and $\boldsymbol{o}_{ADR,i} \in \mathbb{R}^{2N_c}$ is the output of the ADR path. The output $\bar{\boldsymbol{c}}_{i,1}$ is the input of the Recursive path. The first default block operation of the Recursive path reduces the number of channels to half. This process can be formulated as:

$$\bar{\boldsymbol{c}}_{i,2} = \sigma(\mathrm{BN}(H_{3,1}(\bar{\boldsymbol{c}}_{i,1}))), \tag{3.14}$$

where $H_{3,1}(.)$ represents N_c CNN features for the first default block operation of the Recursive path and it reduces the number of channels from $2N_c$ to N_c. $\bar{\boldsymbol{c}}_{i,2}$ is the input of the repeated $N_l - 2$ default block operations. Before the last default block operation, the outputs of this process can be formulated as:

$$\bar{\boldsymbol{c}}_{i,k+1} = \sigma(\mathrm{BN}(H_{3,k}(\bar{\boldsymbol{c}}_{i,k}))),\ k = 2, ..., N_l - 1. \tag{3.15}$$

For the last default block operation of the Recursive path, the number of channels is reduced from N_c to 1. Since the last output of the repeated $N_l - 2$ default block operations is $\bar{\boldsymbol{c}}_{i,N_l}$, the output of the Recursive path can be written as:

$$\bar{\boldsymbol{c}}_{i,N_l+1} = H_{3,N_l}(\bar{\boldsymbol{c}}_{i,N_l}). \tag{3.16}$$

For the last step for the Strided SLM calibration network, the output $\bar{\boldsymbol{c}}_{i,N_l+1}$ and the first corrupted FPA measurement $\boldsymbol{y}_i$ are added together. The output of the network can be written as:

$$\boldsymbol{out}_{strided,i} = \bar{\boldsymbol{c}}_{i,N_l+1} + \boldsymbol{y}_i. \tag{3.17}$$

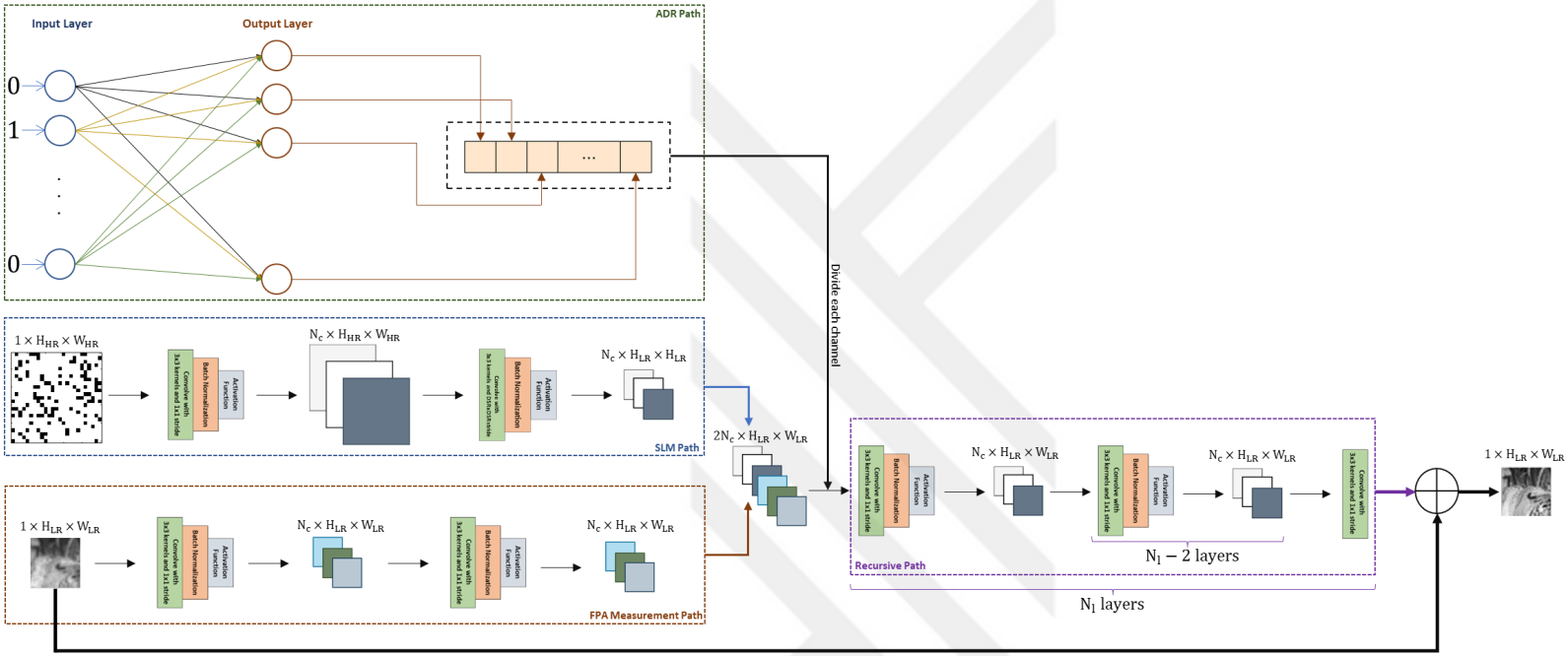

Figure 3.1 : Strided SLM calibration network architecture.

3.3.2 Unshuffled SLM calibration network

This calibration network is similar to the network given in Chapter 3.3.1. Similarly, SLM filters, corrupted FPA measurements, and one-hot encoded ADR range inputs are taken as network inputs. The only difference between Strided SLM and Unshuffled SLM is in the SLM path. The other paths are the same.

The SLM path of the network only includes 1×1 stride operations. Figure 3.2 illustrates the SLM path of the network:

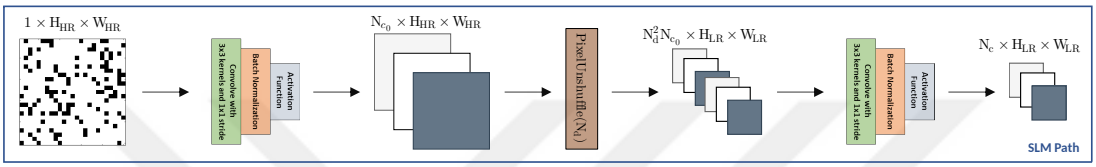

Figure 3.2 : Unshuffled SLM path architecture.

The SLM path takes a single channel SLM encoder filter with the size of $H_{HR} \times W_{HR}$ as input. The first default block operation is applied as in Eq. (3.11). However, the number of features in the first default block operation is N_{c_0}. Consequently, the size of the output of the first default block operation is $N_{c_0} \times H_{HR} \times W_{HR}$. Then, instead of using stride operation as down-sampling process, PixelUnshuffle$(\mathrm{N_d})$ is used. PixelUnshuffle$(\mathrm{N_d})$ is used to increase the number of channels N_d^2 times when both width and height of input images are divided by N_d [57]. For PixelUnshuffle$(\mathrm{N_d})$, this process can be formulated as:

$$\bar{\mathbf{\Lambda}}_{i,PU} = \mathrm{PixelUnshuffle}(\bar{\mathbf{\Lambda}}_{i,1}), \tag{3.18}$$

where $\bar{\mathbf{\Lambda}}_{i,PU}$ defines the output of the PixelUnshuffle$(\mathrm{N_d})$. Since the number of channels for the input of PixelUnshuffle is N_{c_0}, the number of channels of $\bar{\mathbf{\Lambda}}_{i,PU}$ is increased to $N_d^2 N_{c_0}$ while both height and width of the output are diminished to H_{LR} and W_{LR}, respectively. The output of the last default block operation in the SLM path can be written as:

$$\bar{\mathbf{\Lambda}}_{i,3} = \sigma(\mathrm{BN}(H_{2,2}(\bar{\mathbf{\Lambda}}_{i,PU}))), \tag{3.19}$$

where $\bar{\mathbf{\Lambda}}_{i,3}$ gets the same dimensions as in Eq. (3.12) before the concatenation operation is applied.

3.4 Methods

3.4.1 Dataset

A diverse dataset is needed to train both models. For this purpose, DIV2K dataset is used [58]. First of all, all images in the dataset are converted to grayscale images to work with only single-channel inputs. Then, all images are normalized by 255 to acquire normalized images. Since DIV2K contains 900 images, the dataset is divided into 3 parts. The first part has 80% of the dataset, which is equal to 720 images. This part is called training HR images. The second part has 10% of the dataset, which is equal to 90 images. The second part is called validation HR images. The last part has also 10% of the dataset, which is equal to 90 images. The last part is called test HR images.

As the DIV2K is a HR dataset, to get more samples for training and validation processes, the data augmentation step is applied [59]. For this reason, the size of images in both the training and the validation dataset is chosen as 150×150, and HR images are clipped randomly in their dataset. After the clipping procedure, the number of images is increased. However, to eliminate some images, some constraints are defined. The first constraint is formed on the maximum and minimum values of images. If $\max(\text{Image}_\text{k}) - \min(\text{Image}_\text{k}) > 204$, the Image_k satisfies the first constraint. The second constraint is derived from the average values of images. If $\text{avg}(\text{Image}_\text{k}) > 127$, the Image_k satisfies the second constraint. The last constraint is raised for the standard deviation of images. If $\text{std}(\text{Image}_\text{k}) > 51$, the Image_k satisfies the last constraint. If all three conditions are satisfied, the Image_k is selected for the corresponding dataset. Under these constraints, the training dataset includes 7270 images, and the validation dataset includes 981 images.

The data augmentation process is also applied to the test HR images. In contrast to both training and validation HR images, random clipping is applied to create 400×400 images. After the clipping procedure, some constraints are also implemented. The

first constraint of the test dataset is the same as the first constraint of both the training and validation datasets. The second constraint is $\text{avg}(\text{Image}_\text{k}) > 63$ and is a more relaxing condition than the others. The last condition is $\text{std}(\text{Image}_\text{k}) > 63$ and is a harder condition than the others. If all three conditions are satisfied, the Image_k is selected for the test dataset.

3.4.2 Training procedure

Both networks are trained for a 0.8 SLM open rate. The SLM open rate corresponds to the ratio of pixel values other than 0 to the total number of pixels in an HR image. The SLM open rate affects the peak Signal-to-Noise Ratio (pSNR) of the measurements on the FPA detectors. If the SLM open rate is close to 0, measurements have bad pSNR levels. On the other hand, if it is close to 1, the pSNR level is also higher. However, this rate also affects the uniqueness of a measurement. As a result, the 0.8 SLM open rate is critical in maintaining measurement uniqueness while maintaining acceptable pSNR levels.

The Down-Sample Ratio (DSR in Figure 3.1) is chosen as 5 for both dimensions. Since both the training and validation datasets contain 150×150 images, FPA measurements are obtained as 30×30 images. As a result, both networks take 150×150 and 30×30 images for SLM filters and FPA measurements, respectively.

The airy disc kernels are generated for 9 ranges given in Table 3.1. A fixed number of ADRs are produced randomly for each uniformly distributed ADR range. Then, for each item in a batch, ADR is chosen at random while keeping the true ADR range for one-hot encoded input. The size of airy disc kernels is 81×81 due to the suggestion of a kernel with the size of at least $8 \times ADR + 1$ for a simulated ADR value [30]. Since the biggest range for the ADR range is $9.5 - 10.5$, the suggested kernel size is 81×81 for the mean of the biggest ADR range.

The number of epochs for the training of the model is 1000. For each epoch, both the training and validation datasets are shuffled to get a more general solution. The batch size, N_b, is 128, and for each batch, pSNR levels and the model losses are calculated. The learning rate is started at 0.001. This value is multiplied by 0.999 after each epoch

and is forced to take smaller steps than previous epochs. Both training and validation datasets have the same size images. The error function is frequently used in regression problems. In the absence of airy disk effects and noise-induced disturbances in the detector, Eq. (2.9) can be written as:

$$\boldsymbol{y}_{ideal,i} = D(\boldsymbol{\Lambda_i} \odot \boldsymbol{x}),\ i = 1,...,N_s, \tag{3.20}$$

where $\boldsymbol{y}_{ideal,i}$ is the same as the output of Eq. (3.6).

Error operations are applied by taking ℓ_1-norm differences between the network outputs and ideal case measurements given in Eq. (3.20). As a result, if hyper-parameters and model output for a measurement i are represented by θ and $g_\theta(\boldsymbol{\Lambda}_i, \boldsymbol{y}_i, \bar{\boldsymbol{o}}_i)$, respectively, both models try to minimize the optimization problem expressed as:

$$\arg\min_{\theta} \|P'(g_\theta(\boldsymbol{\Lambda}_i, \boldsymbol{y}_i, \bar{\boldsymbol{o}}_i) - \boldsymbol{y}_{ideal,i})\|_1. \tag{3.21}$$

For the model selection criterion, the maximum average validation pSNR level is selected. The average validation pSNR level is calculated as:

$$pSNR_{avg} = \frac{1}{9}\sum_{r=2}^{10} pSNR_{val}(r). \tag{3.22}$$

Validation pSNR levels are obtained for all ADR ranges and the ADR range is represented by r in Eq. (3.22). Each r indicates the mean of each ADR range. As an example, if r is selected as 6, the corresponding ADR range is 5.5-6.5.

3.4.3 Competing method

The Richardson-Lucy (RL) deconvolution technique is an effective method for restoring blurred images [60,61]. The technique estimates an image that closely resembles the observed image when the image is convolved with a PSF. The observed image is a combination of the desired image and the PSF that is convolved with noise. The technique starts with an assumption that an estimated version of the PSF is known. Then, the technique initializes an estimate of the original image, which can be the observed image or a 50% grey image. At each iteration, the RL computes a correction factor for each pixel. These correction factors are the ratio between the observed image and a blurred estimate of the current image. When the correction factors converge to

1, the technique gets a more desired image. Moreover, These correction factors can be formed as an image since they are computed for each pixel and can be computed using image processing techniques.

The RL is sensitive to noise and rapid changes in noise levels lead to artifacts on the deconvolved output image. Hence, the technique is usually used with other regularization methods for more robust results.

3.5 Results

3.5.1 Ablation studies

In this section, three different models are compared to measure the contribution of model changes. The first model does not include any SLM filters. Still, Eq. (3.7) indicates that prior information needs SLM filters to resolve airy disc effects. This model is called *Model w/o SLM* in simulations. The second model takes SLM filters as input. Since the SLM path includes down-sampling with stride operations, this model is called *Model w/ Strided SLM* in simulations. The last model also uses SLM filters and is called *Model w/ Unshuffled SLM*. In this model, the SLM path includes PixelUnshuffle($\mathrm{N_d}$). This block rearranges elements of an input tensor with a shape of $N_c \times H \times W$ to an output tensor with a shape of $N_d^2 N_c \times \frac{H}{N_d} \times \frac{W}{N_d}$. As a result, when the input image is down-sampled, the channel size is increased by N_d^2 and SLM filters can be learned better than directly using *Model w/ Strided SLM* model.

Figure 3.3 demonstrates the average pSNR levels defined in Eq. (3.22) for all three models. As it is seen, using SLM filters as input variable improves the model performances for all simulated model configurations. Simulated model configurations having from the smallest number of model parameters to the largest are $3/32$, $5/32$, $3/48$, $7/32$, $5/48$, $3/64$, $7/48$, $5/64$, $7/64$, and each one of them represents a pair of the number of layers (nL) and the number of features in each layer (nF), respectively. Increasing the number of model parameters for *Model w/o SLM* does not enhance the performances of models significantly. Respectively, SLM filters are crucial to reducing the airy disc effects. On the other hand, *Model w/ Unshuffled SLM* model slightly uses more parameters than *Model w/ Strided SLM* and obtains better results.

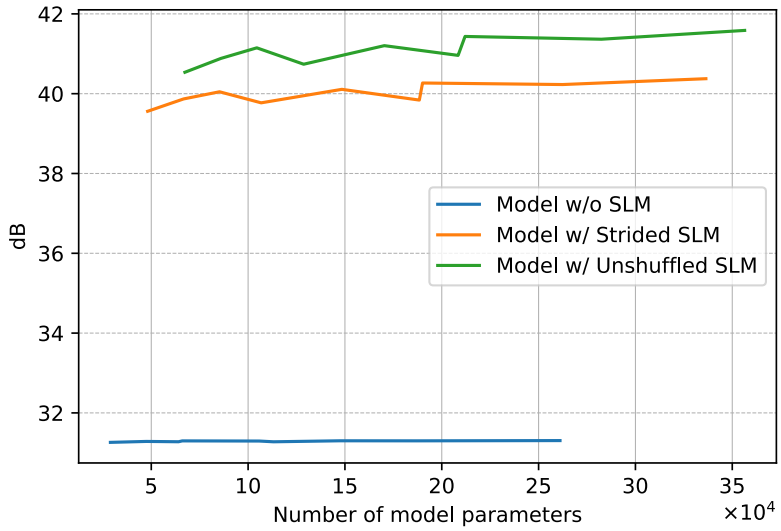

Figure 3.3 : The average pSNRs for all calibration model configurations.

When model configurations are compared for *Model w/ Unshuffled SLM*, $7/32$ and $5/48$ configurations have similar pSNR levels and have 104497 and 170353 parameters, respectively. As a result, $7/32$ is the most convenient model in terms of both pSNR and the number of parameters.

In Figure 3.3, simulated ADR and one-hot encoded inputs of networks are exactly matched. If networks have mismatched one-hot encoded input for true ADR range, the pSNR levels are degraded. Using the validation dataset, Figure 3.4 demonstrates mismatched ADR range inputs for true ADR ranges. As expected, for all scenarios, networks including SLM filters have better performances than *Model w/o SLM*. Even if slightly mismatched ADR range inputs are given, using SLM filters for calibration gives benefits for model outputs. Further, in networks using SLM filters, *Model w/ Unshuffled SLM* is the best model in all cases.

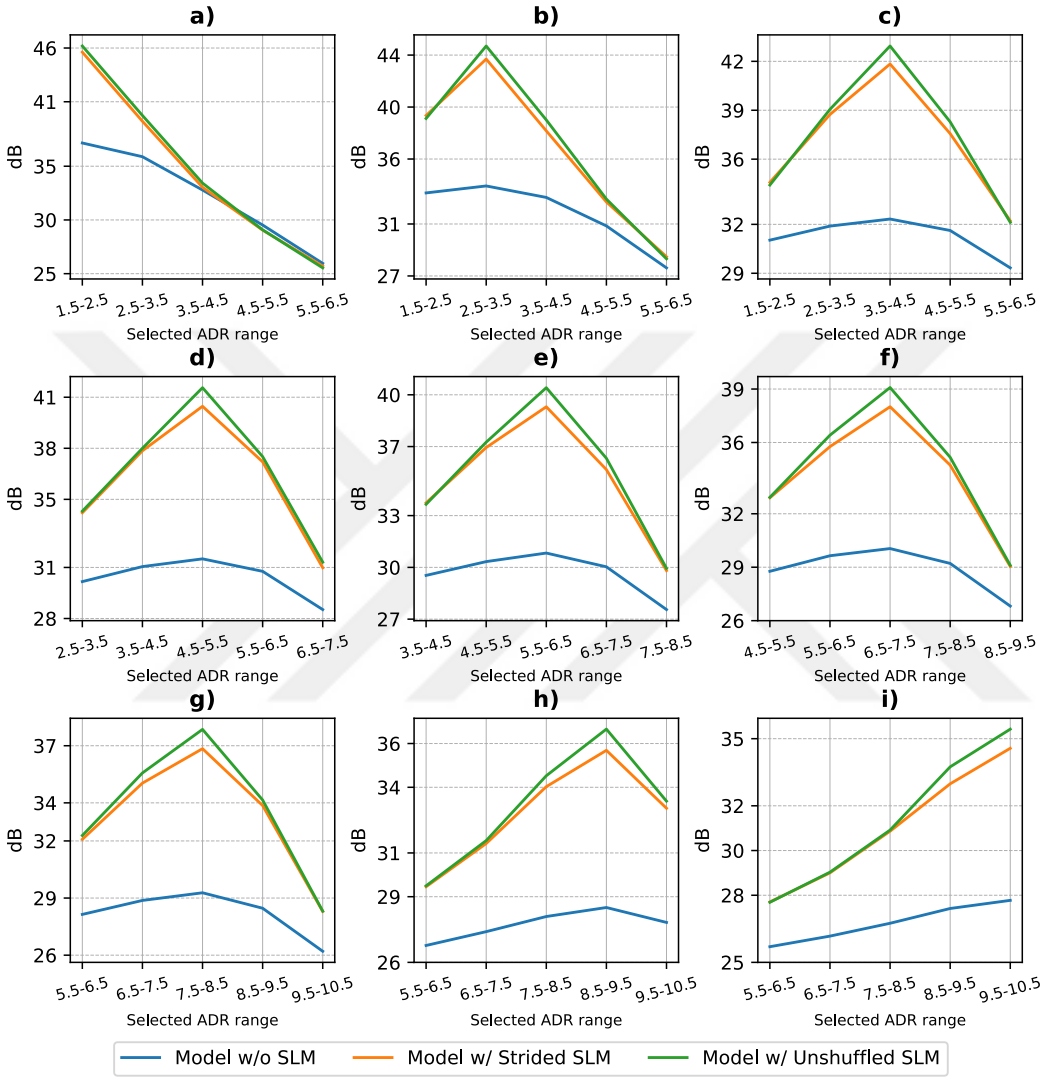

Figure 3.4 : Mismatced ADR range inputs for true ADR ranges, which are **a)** $1.5-2.5$, **b)** $2.5-3.5$, **c)** $3.5-4.5$, **d)** $4.5-5.5$, **e)** $5.5-6.5$, **f)** $6.5-7.5$, **g)** $7.5-8.5$, **h)** $8.5-9.5$, and **i)** $9.5-10.5$.

3.5.2 Simulated results

In this section, all three models have the same configuration, which is $7/32$. In addition to network models, RL deconvolution and raw input are given in simulations. Raw input does not need to be calibrated, and it can be used to compare the effects of different deconvolution processes. On the other hand, the RL deconvolution method needs exact PSFs, and airy disc kernels are used to get exact PSFs in simulations.

Figure 3.5 illustrates a single image comparison for all methods in the case of 60 dB. The first row in the figure from left to right illustrates a true image with the size of 400×400, an encoded image with the size of 400×400, and the ideal FPA image with the size of 80×80 when DSR is chosen 5×5. The FPA image in the first row is the ideal LR image due to the absence of airy disc effects and noise. The second row includes three cases, which are raw inputs when ADR is 2.77, 5.85, and 9.2, from left to right. The other rows represent different deconvolution techniques. As expected, when ADR is increased, FPA images become more blurry and pSNR levels are impaired. Besides, model deblurring performances are diminished due to strengthening airy disc effects. For deconvolution techniques in b), c), d), e) rows, the ADR range is selected $2.5-3.5$, $5.5-6.5$, and $8.5-9.5$ in each column, from left to right. RL deconvolution cannot compete with DL-based methods with the configuration of $7/32$. Using SLM filters in models improves model outputs, and adding more information by SLM filters provides the model to resolve blurry effects in a better way. Therefore, *Model w/ Unshuffled SLM* obtains the best results in all deconvolution methods.

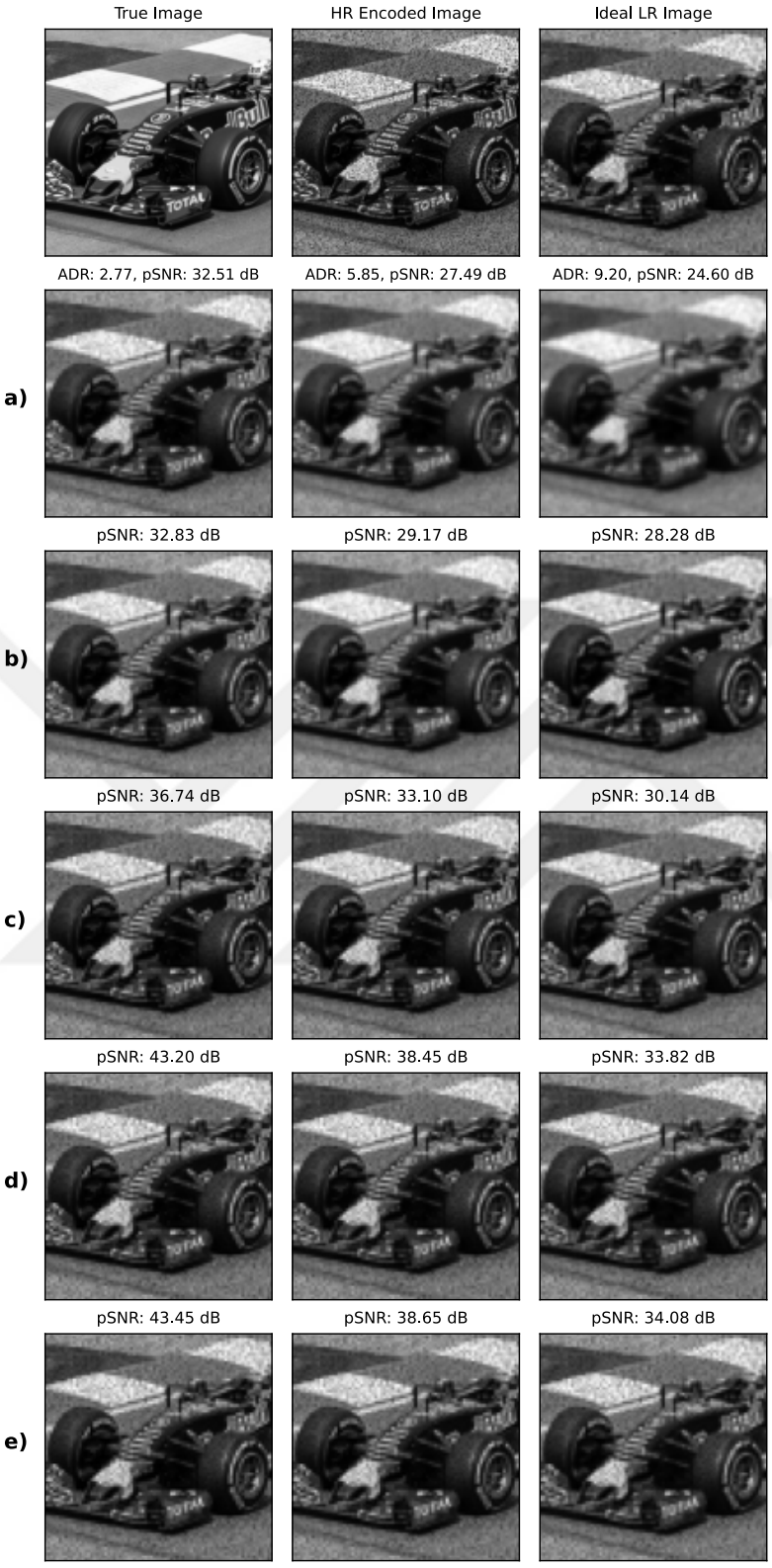

Figure 3.5 : A single image comparison of all calibration methods, which are **a)** Raw inputs with ADRs, **b)** RL deconvolution, **c)** Model w/o SLM, **d)** Model w/ Strided SLM, and **e)** Model w/ Unshuffled SLM. The noise level is computed for 60 dB.

When all simulated ADRs and selected ADR ranges for one-hot encoded inputs are exactly matched, DL-based calibration models with the configuration of $7/32$ achieve the best pSNR performances for them. Furthermore, RL deconvolution needs exact PSFs to resolve deblurring effects. For the validation dataset containing 150×150 images, the average pSNRs can be obtained for each ADR range given in Table 3.1. Figure 3.6 demonstrates that DL-based models have correct ADR ranges for simulated ADR ranges, whereas RL deconvolution employs precise airy disc kernels to use exact PSFs.

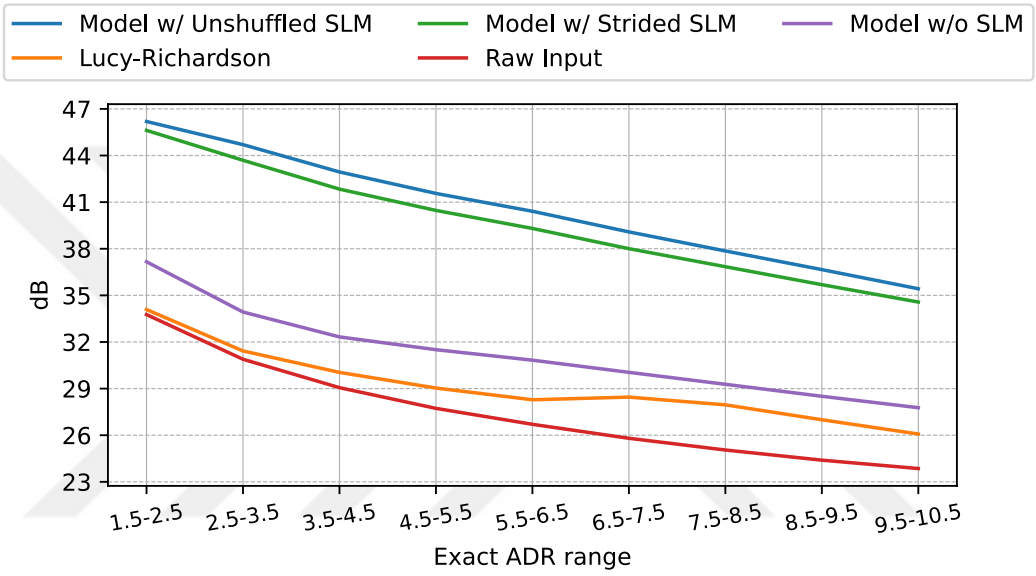

Figure 3.6 : The average pSNRs of all calibration methods in exact ADR ranges.

For all ADR ranges, DL-based models perform better than the RL deconvolution method. The *Model w/ Unshuffled SLM* network gets nearly 13 dB higher pSNR performances than raw inputs for each ADR range. Contrarily, when compared to raw inputs, RL deconvolution only slightly improves corrupted FPA measurements and results in small pSNR gain.

4. UNROLLING-BASED RECONSTRUCTION

4.1 Introduction

A mathematical formulation is transformed into a DL model using a process known as algorithm unrolling [62]. The method can be useful when it is difficult to access or define a prior function in an optimization problem. The technique provides a DL model to be more understandable due to unrolling of each layer operation in the model. In literature, unrolling-based models are adapted for various domains such as image denoising [63,64], image segmentation [65,66], and speech processing [67].

The use of this method has some benefits and drawbacks raised from both End-to-End (E2E) networks and traditional iterative algorithms. In contrast to traditional iterative algorithms that use hand-crafted or physically modeled priors, data-driven priors are necessary for an E2E network, and their representations are difficult. Furthermore, E2E networks lack sufficient training samples to generalize priors for problems. One illustration of a constrained dataset for E2E networks is the field of medical imaging [68]. As a result, an E2E network suffers to both drive and comprehend prior data from a small dataset. On the other hand, DL-based techniques have an advantage over traditional iterative algorithms in terms of computational efficiency. Convolution operations are handled effectively in modern computation systems because they are optimized for a batch of operations in models. However, traditional iterative algorithms need numerous iterations to converge a desired condition for a problem. Additionally, for each iteration in a traditional algorithm, more complex operations are required. For signal and image processing applications, DL-based techniques are preferred because they require shorter execution times for layer operations in DL applications than iteration operations in conventional iterative algorithms [62].

In general, unrolling-based models can represent a model without losing generality, which is known as the overfitting problem for DL techniques [69]. The reason is that

iterative algorithms DL-based architectures are combined in unrolling-based models. As a result, unrolling-based models can effectively learn domain knowledge without needing a large dataset or a lot of processing time.

In Chapter 4.2, some related FPA imaging reconstruction methods are explained. Chapter 4.3 explains the theory of unrolling-based models and different model architectures are given. Then, Chapter 4.4 includes a dataset, training procedure details, and a competing method using the PnP structure with an off-the-shelf denoiser. Chapter 4.5 illustrates simulation results with and without airy disc effects.

4.2 Related Works

In literature, several CS-based approaches have previously been proposed for the reconstruction of FPA imaging.

An approach uses a CS framework and is called the FPA-CS approach. It reconstructs an image from a reduced set of measurements while taking advantage of the sparsity of natural scenes in a transform domain [15]. The FPA-CS approach is divided into two basic steps: first, the FPA sensor gathers random projections of the scene, which are later sent and compressed for use in image reconstruction by a system. Second, a CS-based technique that makes use of the scene's sparsity in a transform domain is utilized to reconstruct the image using compressed measurements. With this technique, a full matrix $\boldsymbol{A}_B$ given in Eq. (2.5) is used. Using a full matrix for the CS technique increases computation times and memory requirements.

For the work given in [70], a matrix-free form reconstruction technique is proposed to lower computation requirements. In an experimental setup, full matrix-based and matrix-free-based convergence times are given. The authors also compare to TVAL3 [71] technique in terms of convergence times. PnP algorithms are also used for the reconstruction of FPA imaging [72].

In [73], images are reconstructed from SPI data by using pre-trained denoisers. The denoisers, which are trained on a different image set, can successfully eliminate noise and other artifacts from SPI data. Using both simulated and actual SPI data, the authors assess the performance of several PnP methods. The outcomes demonstrate

that the PnP algorithms may greatly enhance the reconstruction quality of SPI images, particularly in low-light situations. The possible drawbacks of the PnP strategy, including the necessity for a substantial amount of training data and the computing complexity of the denoising methods, are also discussed in the study. One of the PnP techniques in [73] is PnP-GAP SPI algorithm [74], which is suitable for noiseless scenarios. The other technique in [73] is the PnP ADMM algorithm, which can handle noise in FPA measurements.

In a recent work [75], measurement matrix size is decreased, and each part of an image is encoded with a small-sized measurement matrix. Then, down-sampled images are obtained as images with the size of $HR/B \times HR/B$, where HR denotes the true image size and B is the size of the small measurement matrix. After simulating CS measurements, they are used as inputs in an E2E network to improve CNN model accuracy.

4.3 Theory

In this section, two different ADMM-based methods are introduced for unrolled networks. Later, a Joint Calibration and Reconstruction (JCnR) network architecture is explained. All methods contain an iterative unrolled network for their reconstruction steps. Iterative unrolled networks are developed on ADMM unrolling, which means that networks have the same steps as a classic iterative ADMM algorithm. Figure 4.1 shows the general architecture of an ADMM-based unrolling model. The model takes four inputs, which are combined LR measurements $\boldsymbol{y}_c$, block-diagonal measurement matrix $\boldsymbol{A}_B$, epsilon eps, and initial $\boldsymbol{v}$ as $\boldsymbol{v}^0$. Since the network is based on ADMM steps, there is also an eps input for the data fidelity term. The model has N_{it} iterations to produce a reconstructed HR image. Each iteration step takes the outputs of the previous step as its inputs. These inputs are $\boldsymbol{u}^k$,$\boldsymbol{v}^k$ and $\boldsymbol{d}^k$ for iteration $k+1$. Then, the outputs of iteration $k+1$ are used in the subsequent step, which is iteration $k+2$. At the end of iterations, the model produces an HR image that is reconstructed by a model that is the combination of ADMM and a DL-based network.

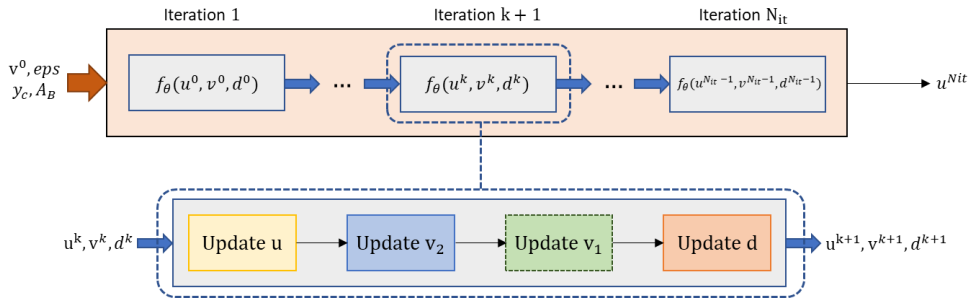

Figure 4.1 : General ADMM-based unrolled network architecture.

From the beginning of the network, only $\boldsymbol{v}^0$ is taken as input to start the model in a better spot. Moreover, $\boldsymbol{v}^0$ can be written as a concatenation of $\dot{\boldsymbol{v}}_1$ and $\boldsymbol{v}_2$, which have the same shape of $P'(\boldsymbol{u})$ and $\boldsymbol{A}_B P'(\boldsymbol{u})$, respectively. For initial $P'(\boldsymbol{v}_1) = \dot{\boldsymbol{v}}_1$, an optimization problem in Eq. (4.1) can be written as:

$$\arg\min_{\dot{\boldsymbol{v}}_1} \|\dot{\boldsymbol{v}}_1\|_2^2 \text{ subject to } \|\boldsymbol{y}_c - \boldsymbol{A}_B\dot{\boldsymbol{v}}_1\|_2^2 \leq \gamma, \tag{4.1}$$

where γ is aimed to be small. Equation (4.2) is obtained by taking the derivative with respect to $\dot{\boldsymbol{v}}_1$ and setting it equal to 0:

$$\frac{\partial}{\partial \dot{\boldsymbol{v}}_1}(\dot{\boldsymbol{v}}_1^H\dot{\boldsymbol{v}}_1 + \lambda(-\boldsymbol{y}_c^H\boldsymbol{A}_B\dot{\boldsymbol{v}}_1 - \dot{\boldsymbol{v}}_1^H\boldsymbol{A}_B^H\boldsymbol{y}_c + \dot{\boldsymbol{v}}_1^H\boldsymbol{A}_B^H\boldsymbol{A}_B\dot{\boldsymbol{v}}_1)) = 0. \tag{4.2}$$

After that, $\dot{\boldsymbol{v}}_1$ can be found as:

$$\dot{\boldsymbol{v}}_1 = \lambda(I + \lambda\boldsymbol{A}_B^H\boldsymbol{A}_B)^{-1}\boldsymbol{A}_B^H\boldsymbol{y}_c. \tag{4.3}$$

If Singular Value Decomposition (SVD) form of $\boldsymbol{A}_B$ is written as $\boldsymbol{A}_B = \boldsymbol{U}_B\boldsymbol{S}_B\boldsymbol{V}_B^H$, after simplifications, $\boldsymbol{v}_1$ can be found as:

$$\boldsymbol{v}_1 = P\left(\boldsymbol{V}_B^H\frac{\lambda\boldsymbol{S}_B^H}{I + \lambda|\boldsymbol{S}_B|^2}\boldsymbol{U}_B^H\boldsymbol{y}_c\right), \tag{4.4}$$

where λ is a variable to be determined by N_s. If N_s is smaller than DSR^2, λ can be chosen lower than 1. On the other hand, if $N_s \geq DSR^2$, λ can be chosen greater than 1. Equation (4.4) produces similar output to the LS solution. The key difference between them is that, while working with undersampled cases, using an optimization problem in Eq. (4.1) brings more generality. For the last step, initial $\boldsymbol{v}_1$ is clipped into a range $[0, 1]$ to eliminate outliers.

For initial $\boldsymbol{v}_2$, the computed initial $\dot{\boldsymbol{v}}_1$ can be used as:

$$\boldsymbol{v}_2 = \boldsymbol{A}_B\dot{\boldsymbol{v}}_1. \tag{4.5}$$

After all, initial $\boldsymbol{v}^0$ is given at the beginning of the network.

The input eps is related to noise levels on FPA measurements. For an ideal FPA image i, the pSNR level can be found as:

$$pSNR = 20 \times log_{10}\left(\frac{max(\boldsymbol{y}_{ideal,i})}{\sigma_i}\right), \tag{4.6}$$

where σ_i represents the standard deviation of noise computed on ideal FPA measurement i. If σ_i is found by a given pSNR level, Eq. (4.7) can be obtained as:

$$\sigma_i = 10^{-pSNR/20} \times max(\boldsymbol{y}_{ideal,i}). \tag{4.7}$$

The data fidelity term is written for multiple snapshots. Therefore, eps is formed by both N_s and LR image size and can be obtained as:

$$eps = \sigma_i \times \sqrt{N_s \times W_{LR} \times H_{LR}}. \tag{4.8}$$

In Figure 4.1, iteration $k+1$ includes several steps. For the first step, $\boldsymbol{u}$ is updated by the optimization problem given in Eq. (2.26). For the second step, $\boldsymbol{v}_2$ is updated by the optimization problem given in Eq. (2.28). For the third step, $\boldsymbol{v}_1$ is updated by the optimization problem given in Eq. (2.27). In classical CS iterative algorithms, usually, hand-crafted priors or pre-trained denoisers are used for updating $\boldsymbol{v}_1$. On the contrary, an unrolling-based network contains a network for its denoiser step, which is constructed by CNNs. This step allows the network to learn a rich parameterization of the prior of a given dataset. Consequently, the overall network learns a prior from its dataset in the unrolling-based model reconstruction process. Lastly, in iteration $k+1$, dual variable updates are run as in both Eq. (2.29) and Eq. (2.30).

4.3.1 Base unrolled network

This section explains a CNN model architecture named as *Base Unrolled Net* for the $\boldsymbol{v}_1$ updating stage. The overall model is the same as the model given in Figure 4.1. Similarly, it takes $\boldsymbol{y}_c$, $\boldsymbol{A}_B$, $\boldsymbol{v}^0$, and eps as inputs for the model. For the input $\boldsymbol{v}^0$, the model uses findings in both Eq. (4.4) and Eq. (4.5). For the denoiser step, it takes the difference of the last updated $\boldsymbol{u}^{k+1}$ and $\boldsymbol{d}_1^k$ variables as inputs. In Figure 4.2, the input and output of the model are represented.

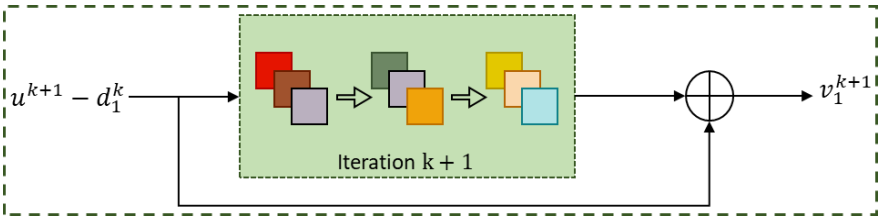

Figure 4.2 : Input and output of the denoiser of *Base Unrolled Net* at iteration $k+1$.

The model produces $\boldsymbol{v}_1^{k+1}$ at the output of the denoiser in iteration $k+1$. In general, model architecture can be represented in Figure 4.3.

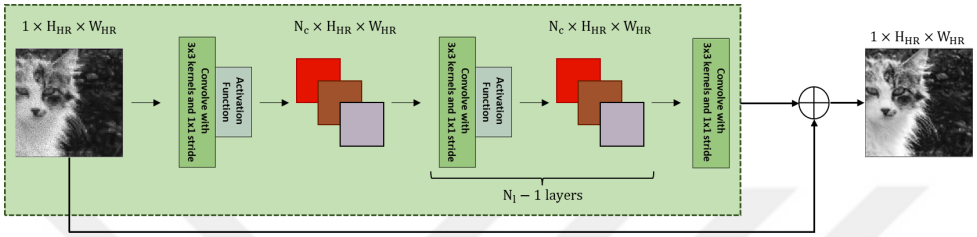

Figure 4.3 : Model architecture of the denoiser of *Base Unrolled Net*.

The model takes a single-channel input as $1 \times H_{HR} \times W_{HR}$. Then, the input image is passed in a convolution layer with N_c features. For the last step in the first layer operations, *LeakyReLU* is used for the activation function [76]. The output of the first layer operations at iteration $k+1$ can be formulated as:

$$\bar{\boldsymbol{v}}_{1,1}^{k+1} = \sigma(F_1(\boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k)), \tag{4.9}$$

where $F_1(.)$ represents N_c CNN kernels, and $\sigma(.)$ is for *LeakyReLU* activation function. The output of Eq. (4.9) indicates the output of the first layer operations for the model at iteration $k+1$. The next $N_l - 1$ layers are repeated as below:

$$\bar{\boldsymbol{v}}_{1,j+1}^{k+1} = \sigma(F_{j+1}(\bar{\boldsymbol{v}}_{1,j}^{k+1})),\ j = 1, ..., N_l - 1,\ N_l \geq 2, \tag{4.10}$$

where $F_{j+1}(.)$ represents N_c CNN kernels at the $(j+1)$-th layer of the model. In Eq. (4.10), the number of layers N_l is assumed to be greater or equal to 2 to satisfy the formulation. For the last layer, the output of CNN blocks can be formulated as:

$$\bar{\boldsymbol{v}}_{1,N_l+1}^{k+1} = F_{N_l+1}(\bar{\boldsymbol{v}}_{1,N_l}^{k+1}). \tag{4.11}$$

For the last step in the model at iteration $k+1$, CNN output and noisy input are added together as follows:

$$\boldsymbol{v}_1^{k+1} = \bar{\boldsymbol{v}}_{1,N_l+1}^{k+1} + (\boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k). \tag{4.12}$$

As a result, the model learns slight changes for input images instead of obtaining the whole image from a random start.

4.3.2 Residual unrolled network

This section is an improved version of the model given in the previous section, and this model is called *Residual Unrolled Net* (or shortly, *Res Unrolled Net*). The current model also has the same architecture as the model given in Figure 4.1 and takes four inputs, which are $\boldsymbol{y}_c$, $\boldsymbol{A}_B$, $\boldsymbol{v}^0$, and eps. For the initial $\boldsymbol{v}^0$, the method uses the outputs in both Eq. (4.4) and Eq. (4.5). The denoiser step at iteration $k+1$ takes two inputs, which are the two last noisy inputs computed in the current iteration and previous iteration k. Here, noisy inputs infer inputs of the denoiser step. In Figure 4.4, inputs and output of the model at iteration $k+1$ are demonstrated.

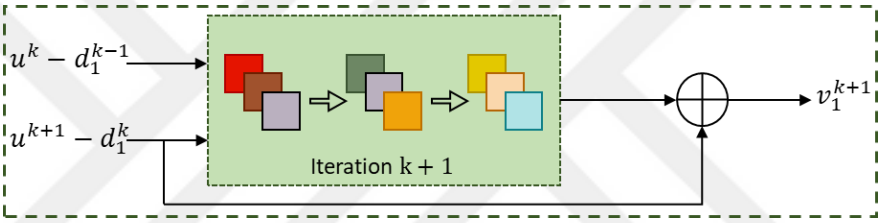

Figure 4.4 : Inputs and output of the denoiser of *Residual Unrolled Net* at iteration $k+1$.

The model architecture is illustrated in Fig 4.5.

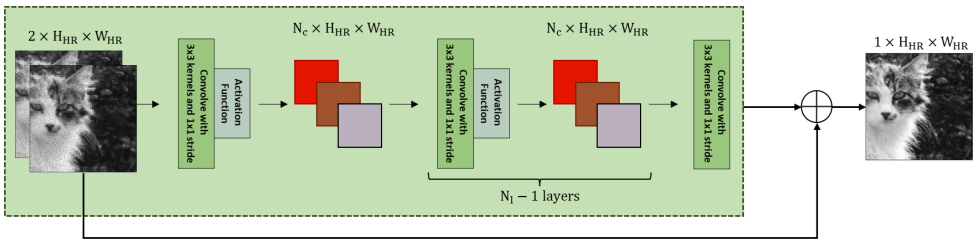

Figure 4.5 : Model architecture of the denoiser of *Residual Unrolled Net*.

It takes two channels input as $2 \times H_{HR} \times W_{HR}$. These two channels' input is routed through N_c CNN kernels before the *LeakyReLU* activation function is used. Equation (4.13) formulates the output of the first layer operations as:

$$\bar{\boldsymbol{v}}_{1,1}^{k+1} = \sigma(F_1([\boldsymbol{u}^k - \boldsymbol{d}_1^{k-1},\ \boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k)]), \tag{4.13}$$

where $[\boldsymbol{u}^k - \boldsymbol{d}_1^{k-1},\ \boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k)]$ represents two channels input in a concatenated input. The output $\bar{\boldsymbol{v}}_{1,1}^{k+1}$ has the size of $N_c \times H_{HR} \times W_{HR}$. As a result, instead of *Base Unrolled Net*, the current model maps two channels input to N_c channel output. Further, it uses more information to get a representation of a prior. For the next $N_l - 1$ layers, outputs can be found by using Eq. (4.10). Likewise, the output of CNN blocks can be found by using Eq. (4.11). The last step of the model is the addition of the noisy input of the current iteration $k+1$, which is $\boldsymbol{u}^{k+1} - \boldsymbol{d}_1^k$. As a result, the output of the model can be formed by using Eq. (4.12). In contrast to producing the entire image, the model converges quickly and learns little adjustments for each pixel, as in *Base Unrolled Net*.

4.3.3 Joint calibration and reconstruction network

The previous two sections focus only on the reconstruction of HR images from FPA measurements. For this purpose, a classical CS iterative technique and a DL-based model are combined. This technique is known as algorithm unrolling, and in previous sections, two methods are explained. The methods can be separated from each other when the denoiser steps are analyzed. In this section, not only the reconstruction process but also the DL-based calibration technique is included in the same model. Hence, this model is named *JCnR Net*. In Figure 4.6, a general network architecture is shown.

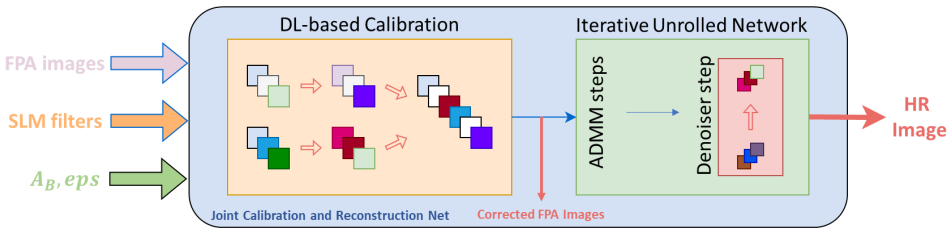

Figure 4.6 : General network architecture of *JCnR Net*.

The model takes three types of inputs, which are FPA images, SLM filters, and $\boldsymbol{A}_B$ with eps for an iterative unrolled network. The first step of the network is obtaining corrected FPA measurement by using a DL-based calibration model in the network. The calibration model can be chosen either *Strided* or *Unshuffled SLM Calibration Net*. After the calibration process, corrected FPA images are obtained for an ADR range. Instead of using one-hot encoded inputs to make the calibration network work

for all ADR ranges, the *JCNR Net* does not have one-hot encoded inputs for ADR range selection. The reason is related to the generalization of the model loss, and details are explained in Chapter 4.4.2.

The second step of the network contains an ADMM-based unrolled network. This step can be realized by *Base* or *Residual Unrolled Nets*. Furthermore, the denoiser step of the ADMM-based unrolled network can be implemented by using different types of network architectures like U-Net [68]. In Chapter 4.5.1, the performance of *U-Net Unrolled Net* is compared with *Base* and *Residual Unrolled Nets*. The corrected FPA images are taken as inputs by a selected iterative unrolled network. Then, using $\boldsymbol{A}_B$ and eps, a reconstructed HR image is obtained as the output of the network given in Figure 4.6.

Figure 4.7 expresses detailed model architecture of the *JCnR Net*. The calibration process is handled image-to-image, which means that for an FPA measurement, a single-channel HR SLM filter is used. On the contrary, an iterative unrolled network is constructed on a CS-based ADMM algorithm. Additionally, this algorithm takes multiple snapshots to reconstruct an HR image. Because of this, the calibration network takes multiple snapshots in a batch process using a single input channel. In the same way, corresponding SLM filters are taken in the batch direction. Thus, when N_B indicates the number of batch inputs and N_s is for multiple snapshots, the size of FPA images is $N_B N_s \times 1 \times H_{LR} \times W_{LR}$ and the size of SLM filters is $N_B N_s \times 1 \times H_{HR} \times W_{HR}$. Correspondingly, the size of the output of the calibration process in the network is $N_B N_s \times 1 \times H_{LR} \times W_{LR}$. As expected, the iterative unrolled network takes these inputs and produces an output with the size of $N_B \times 1 \times H_{HR} \times W_{HR}$. For the reconstruction process, $\boldsymbol{A}_B$ and eps are used in ADMM steps. For a selected denoiser type, the number of channels of input images of the denoiser step changes. In Figure 4.7, curly brackets represent two different paths for denoiser architectures. For a selected denoiser, the input passes through only one path of them at a time, and at the end of the model, HR reconstructed image is obtained.

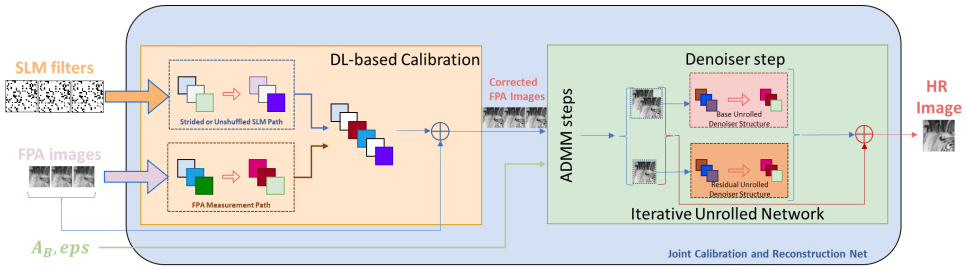

Figure 4.7 : Detailed network architecture of *JCnR Net*.

4.4 Methods

4.4.1 Dataset

The dataset is the same as the dataset mentioned in Chapter 3.4.1.

The dataset is divided into three categories, which are training, validation, and test sets. For all categories, image datasets are augmented by clipping. The clipping process for training and validation is executed to get a bunch of 150×150 HR images. Each image k satisfies $\max(\text{Image}_k) - \min(\text{Image}_k) > 204$, $\text{avg}(\text{Image}_k) > 127$, and $\text{std}(\text{Image}_k) > 51$ for both training and validation. Furthermore, the clipping process of the test set produces 400×400 images. For each image k in the test set, $\max(\text{Image}_k) - \min(\text{Image}_k) > 204$, $\text{avg}(\text{Image}_k) > 63$, and $\text{std}(\text{Image}_k) > 63$ conditions are satisfied. The test images are chosen bigger because reducing the size of the test images does not affect image properties, and simulations given in Chapter 4.5 have lower sizes than the size of images in the original test set.

4.4.2 Training procedure

All three networks given in Chapter 4.3 have 0.8 SLM open rate. Besides, all networks use DSR as 5, which means that they take 30×30 FPA images and 150×150 SLM filters for the defined training dataset. Noise realizations on FPA detectors are selected for the 60 dB scenario.

For both *Base* and *Res Unrolled Nets*, N_b is 32, and the number of epochs is selected as 500. The learning rate starts at 0.001 and after each epoch, it is multiplied by 0.998. The denoiser step in networks starts with an LS solution given in Eq. (4.4). The

number of layers, nL, is 5, and the number of features, nF, is 64 for both networks. Moreover, the number of iterations (N_{it}) is 8. The N_{it} signifies how many iterations are applied before computing a model loss. Figure 4.8 shows how model loss is obtained for each model run in the training step.

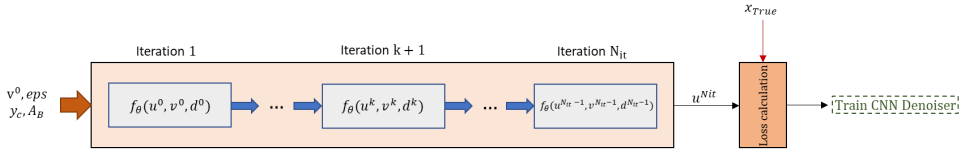

Figure 4.8 : Model loss for both *Base* and *Residual Unrolled Nets*.

At the end of an unrolled model, the output image has the same shape as the true HR image, $\boldsymbol{x}_{True}$. If the output of the general model in Figure 4.1 is represented by $g_\theta(\boldsymbol{y}_c, \boldsymbol{A}_B, \boldsymbol{v}^0, eps)$, model loss can be written as:

$$loss = \|P'(g_\theta(\boldsymbol{y}_c, \boldsymbol{A}_B, \boldsymbol{v}^0, eps) - \boldsymbol{x}_{True})\|_1. \tag{4.14}$$

As a result, the model loss is correlated with a function of FPA measurements, SLM filters, denoiser starting point, and epsilon of the ADMM procedure. Moreover, θ represents the model parameters and changes the function's behavior. Thus, different architectures obtain different losses. For the model, any *eps* value can be given, but in the training process, exact *eps* values are used. These values are calculated as in Eq. (4.8). Since the *eps* value is related to noise standard deviation, in the 60 dB case, for each image i, σ_i is found and used in the model. Afterward, the loss function in Eq. (4.14) depends on ℓ_1-norm. Both networks update the best models after each epoch if at the current epoch, validation pSNR performance is better than the last model's pSNR performance. As a result, the model selection is determined by the validation set performances.

For the *JCnR Net*, the number of epochs is selected as 250. The batch size, N_b is 32. The learning rate starts at 0.001 and after every three epochs, it is multiplied by 0.989. Like both unrolled networks, the denoiser takes an LS solution in Eq. (4.4) as the initial point. Since the current model includes both calibration and iterative unrolled networks, the number of layers in the calibration path (*cnl*) is 5, and the number of features in each calibration layer (cnf) is 48. For the denoiser in an iterative

unrolled path, the number of layers (dnl) and the number of features (dnf) are 5, and 48, respectively. The number of iterations, N_{it}, in unrolled path is 8. This method computes two different losses, and their weighted combination gives the model loss. In Figure 4.9, general loss contributions from calibration and iterative unrolled paths are illustrated. Both outputs from calibration and iterative unrolled network are used in model loss computation. The loss calculation is performed by the combined loss block.

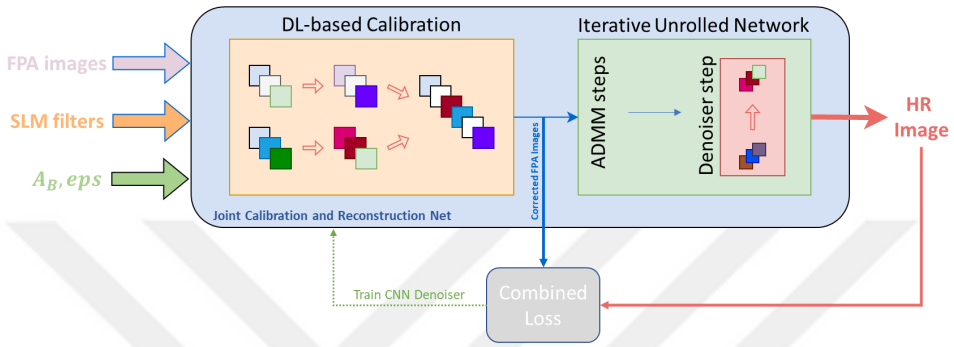

Figure 4.9 : Model loss for *JCnR Net*.

The first loss is LR loss, and if airy disc effects and noise are not seen on FPA measurements, ideal LR measurements are obtained as in Eq. (3.20). These ideal measurements can be achieved by using an average pooling operation on SLM-encoded HR images [77]. Thus, LR loss in a batch can be formulated as:

$$loss_{LR} = \sum_{i=1}^{N_s N_B} \|P'(g_{\theta_1}(\mathbf{\Lambda}_i, \mathbf{y}_i) - \mathbf{y}_{ideal,i})\|_1, \tag{4.15}$$

where $g_{\theta_1}(\mathbf{\Lambda}_i, \mathbf{y}_i)$ defines the i-th output of the calibration path. In Chapter 3.3, ADR range selection is implemented by a one-hot encoded input, and the model training is performed in the same model for all ADR ranges given in Table 3.1. However, here, the *JCnR Net* finds an output for the HR image, and the model generalization drastically affects the model performance. This is because the model loss is dominated by simulations affected by higher ADR ranges. In other words, from the first ADR range, $1.5-2.5$, to the last ADR range, $9.5-10.5$, loss contribution increases drastically. As a result, a generalized model gets lower pSNR performances than a network trained for a specific ADR range. Consequently, the model does not take any ADR range selection input, and for each ADR range, a *JCnR Net* is trained. Equation (4.15) also

shows that there are $N_s N_B$ images per batch operation. This is because the calibration path works as a single snapshot process, which means that for an HR SLM filter, an LR FPA measurement is used.

The iterative unrolled network takes corrected FPA measurements and attains HR images after 8 iterations. In this process, this part takes multiple snapshots and performs an ADMM-based unrolled algorithm. Hence, this part produces HR loss for weighted combined loss. The HR loss in a batch can be written as:

$$loss_{HR} = \sum_{i=1}^{N_B} \|P'(g_{\theta_2}(\boldsymbol{y}_{c,i}, \boldsymbol{A}_{B,i}, \boldsymbol{v}_i^0, eps_i) - \boldsymbol{x}_{True,i})\|_1, \tag{4.16}$$

where $g_{\theta_2}(\boldsymbol{y}_{c,i}, \boldsymbol{A}_{B,i}, \boldsymbol{v}_i^0, eps_i)$ is the output HR image of the iterative unrolled path for test image i. The output of the path and $\boldsymbol{x}_{True,i}$ have the size of $1 \times 1 \times H_{HR} \times W_{HR}$. Since a batch contains N_B images, $\|.\|_1$ operation is repeated N_B times for a batch. Then, these findings are summed up to define $loss_{HR}$ for a batch.

Both LR and HR losses are used to define the joint model loss that is given as:

$$loss_{joint} = \alpha \times loss_{HR} + (1-\alpha) \times loss_{LR},\ 0 < \alpha < 1, \tag{4.17}$$

where α defines the HR loss coefficient. In Figure 4.10, α is started at 0.45 and is linearly increased in the first half of epochs.

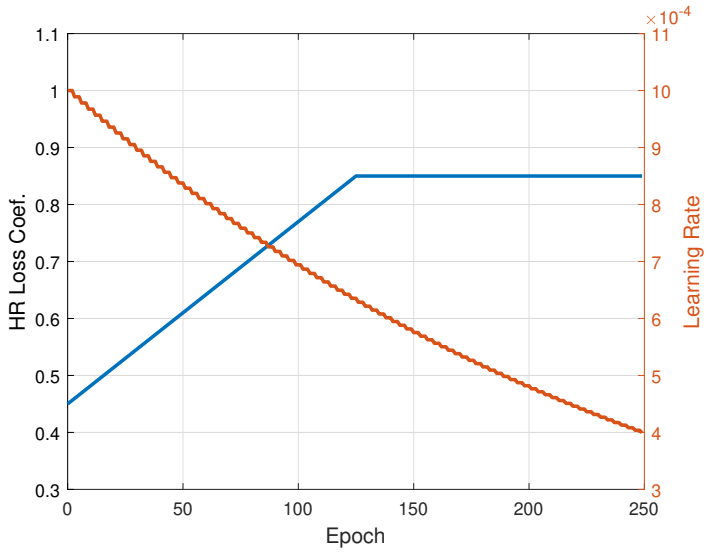

Figure 4.10 : HR loss coefficient and learning rate vs. epoch.

For the second half of epochs, α remains constant at 0.85. In the beginning, the model gives more importance to $loss_{LR}$. After 17 epochs, $loss_{HR}$ takes more attantion. Then,

it starts to become more effective with each epoch. With the second half of epochs, α is not changed, and the model accounts for 85% of contribution from HR image reconstruction. The change in learning rate has a staircase shape versus epochs. When the model reaches the half of epochs, the learning rate is 0.0006354. After each epoch, the model takes smaller steps than before. As a result, between α and the learning rate, there is a balance to hold the model that learns better.

The *JCnR Net* takes *eps* value for corrected FPA measurements. The model has trained under the 60 dB assumption, but FPA measurements are affected by airy disc effects. For each ADR range, expected validation pSNR performance can be used. For this purpose, Table 4.1 expresses pSNR levels are used in Eq. (4.7) to calculate *eps* values. It shows that slightly increased pSNR levels are used for each ADR range. This is because the network utilizes two weighted loss functions as the model loss, and small improvements in pSNR levels can be seen. Therefore, from $1.5-2.5$ to $9.5-10.5$, the model's *eps* is chosen for better pSNR levels than their counterparts.

Table 4.1 : Chosen pSNR levels in *eps* calculation when the calibration is *Unshuffled SLM Calibration Network*.

ADR Range	*Unshuffled SLM Calibration Network*	pSNR for *eps*
1.5-2.5	47.8	48.0
2.5-3.5	45.4	45.7
3.5-4.5	43.6	44.0
4.5-5.5	42.3	42.8
5.5-6.5	41.0	41.6
6.5-7.5	39.7	40.4
7.5-8.5	38.4	39.2
8.5-9.5	37.2	38.1
9.5-10.5	36.0	37.0

4.4.3 Competing methods

In this section, several reconstruction techniques are briefly explained. All techniques except *LS* and *U-Net Unrolled Net* exploit the PnP-ADMM structure.

The most basic reconstruction technique is *LS*, which is used to show the baseline for all methods. In this technique, Eq. (4.4) is applied to get the solution. This technique aims to get a solution that minimizes ℓ_2 difference between $\boldsymbol{y}_c$ and $\boldsymbol{A}_B P'(\boldsymbol{x})$. Here, $\boldsymbol{x}$ represents the desired HR image.

The second technique uses *Total Variation*-norm (*TV*-norm) as a prior model in ADMM structure [78]. The variation of adjacent pixels in an image is mathematically measured using the *TV*-norm. The sum of the absolute differences between adjacent pixels is used to calculate for updating of $\boldsymbol{v}_1$ step in Eq. (2.27). For $\boldsymbol{v}_1$, *TV*-norm can be formulated as:

$$\|\boldsymbol{v}_1\|_{TV} = \sum_j \sqrt{(\nabla_1|\boldsymbol{v}_1[j]|)^2 + (\nabla_2|\boldsymbol{v}_1[j]|)^2}, \tag{4.18}$$

where $\nabla_1(.)$ and $\nabla_2(.)$ represent horizontal and vertical derivatives, respectively. The *TV*-norm can be incorporated as a term in the optimization objective function, and in literature, many studies exploit to use of *TV* in CS problems [79,80].

The third technique combines both *TV* and ℓ_1-norm in the Fourier domain (*FL1*). The *FL1*-norm relies on soft-thresholding operations [45], and in [70], two priors are combined as:

$$Prior(\boldsymbol{v}_1) = \beta_1\|\boldsymbol{v}_1\|_{TV} + \beta_2\|P'(\boldsymbol{F}\boldsymbol{v}_1)\|_1, \tag{4.19}$$

where $\boldsymbol{F}$ defines a two-dimensional Fourier transform, and the regularization coefficients are labeled as β_1 and β_2. With the change of regularization coefficients, the prior of the problem can be defined in different ways.

The fourth technique is *Deep CNN* denoiser (*DnCNN*). In order to reduce image noise, the DnCNN image denoising model employs DL methods [81]. It is built using a CNN architecture that is intended for learning the mapping between noisy images and the equivalent true images. Several layers of CNNs are often used in the denoisers to identify and eliminate noise from various areas of the image. To understand the patterns and characteristics that distinguish between noise and signal in images, the model is trained using a sizable dataset of noisy and true images. In Figure 4.11, a CNN-based model structure is demonstrated.

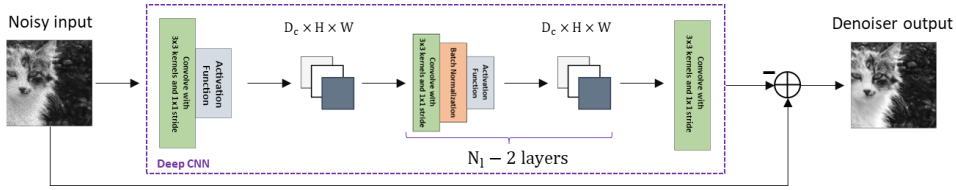

Figure 4.11 : *DnCNN* denoiser model structure.

The denoiser structure is similar to both *Base* and *Res. Unrolled Nets*. The network takes noisy input and learns how to remove noisy effects from true input for a selected sigma parameter. D_c represents the number of channels after a CNN block. It is determined by the denoiser number of features. The number of layers is expressed by N_l, and $N_l - 2$ defines the repeated part of the network. For the output, CNN blocks produce a residual image, which is not a whole image, and the model output is obtained by subtracting the residual image from the noisy input. This model can be used as a PnP model in the ADMM structure. For this purpose, a pre-trained model when the noise level is 25 is used in the evaluation mode [82].

The last technique is *U-Net Unrolled Net*, which is an unrolling network including U-Net type of a denoiser to update $\boldsymbol{v}_1$. In the U-Net design [68,83], wf defines the number of feature channels in the first layer as 2^{wf} and d defines the depth of the contracting path. For the contracting path, 3×3 kernels are used in feature maps. After each layer, the *ReLU* activation function is used. To lower the size of an image, a downsampling operation is applied by a stride with the size of 2×2. At each downsampling step, the number of feature channels is doubled. At the expansive path, the upsampling operation is applied by an $Up-Conv.$ block. With these blocks, the number of feature channels is halved, and a concatenation procedure is applied. As a result, the number of feature channels remains the same as before. Then, the number of channels is reduced to half by convolution operations. For any wf and $d = 6$, Figure 4.12 shows U-Net denoiser model architecture.

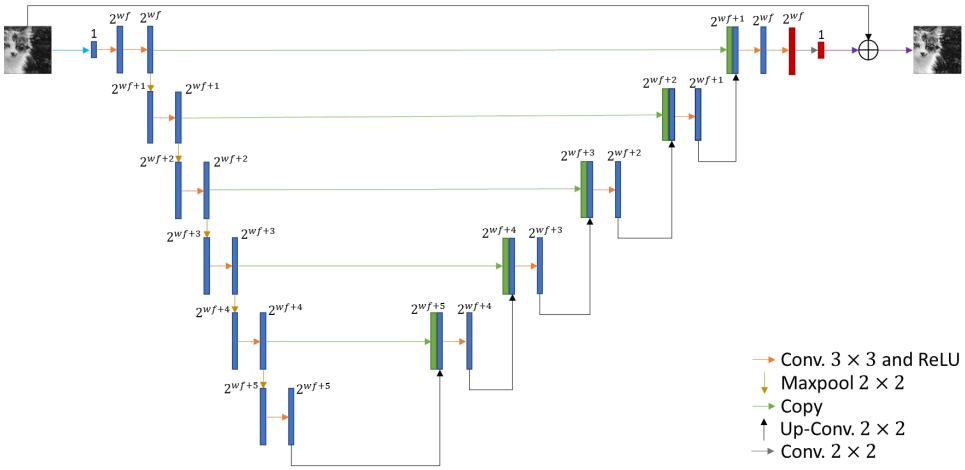

Figure 4.12 : Model architecture of denoiser of *U-Net Unrolled Net* for $d = 6$.

4.5 Results

4.5.1 Ablation studies

In this section, several simulation analyzes are given. With these simulations, some preliminary results for models are obtained before the simulated results that include with and without airy disc effects at numerous N_s scenarios.

The first analysis is the validation pSNR results that are compared for three unrolled methods, which are *Base*, *Residual*, and *U-Net Unrolled Nets*. For this purpose, N_s is limited to 5, and there are two designs for the *U-Net Unrolled Net* architecture. In Figure 4.13, all unrolled methods are shown.

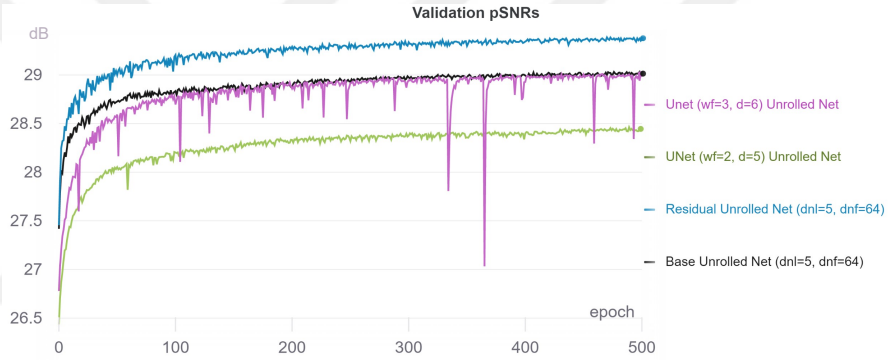

Figure 4.13 : Validation pSNRs for all unrolled methods using $N_s = 5$.

The *Res Unrolled Net* converges faster than other methods and is compared to the *Base Unrolled Net*, they have the same network structure in terms of both *dnl* and dnf. The *dnl* is N_l, and the dnf is the N_c in model architectures given in both Figure 4.3 and Figure 4.5. Thus, both models have the same dnl/dnf structure as $5/64$. Both *U-Net Unrolled Nets* perform worse performance than *Base* and *Res Unrolled Nets*. Moreover, using $wf = 2$ and $d = 5$ gets the worst performance in terms of validation pSNR. On the other hand, a *U-Net Unrolled Net* using $wf = 3$ and $d = 6$ has approximately the same performance with *Base Unrolled Net*. However, the model is more unstable than *Base Unrolled Net*. This is because its performance is not seen as continuously increasing with regard to epochs. In the following simulations, the *U-Net Unrolled Net* is referred to using the *U-Net Unrolled Net* with $wf = 3$, $d = 6$ case.

The test dataset includes 235 images with the size of 400×400. Test scenarios for including the *U-Net Unrolled Net* contain 150×150 and 310×310 images. To obtain these resolutions, a resizing operation is used [84]. This resizing operation eliminates anti-aliasing effects as a default operation. Table 4.2 expresses test results for 150×150 test images.

Table 4.2 : Test results of all unrolled networks for 150×150 images.

Network	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times$100)	LPIPS ($\times$100)
U-Net Unrolled	18.11	26.24	35.82	84.96	**21.66**
Base Unrolled	18.07	26.25	35.99	85.00	22.54
Residual Unrolled	**18.28**	**26.57**	**36.33**	**85.64**	21.93

There is an average pSNR result for each network. The table also includes minimum and maximum obtainable pSNR levels for the constructed simulation test set. When they are compared for pSNR levels, both *Base* and *U-Net Unrolled Nets* have similar performances whereas *Res Unrolled Net* has the best performance in terms of pSNR. Moreover, the average Structural Similarity Index (SSIM) result is given, and SSIM evaluates the similarity of two images by comparing their local patterns of pixel intensities. It produces a score between 0 and 1, with 1 denoting complete identity. More similarity between the photos is indicated by a higher SSIM score [85]. SSIM metric is used as $SSIM(\times 100)$ to magnify differences between networks. Again, both *Base* and *U-Net Unrolled Nets* have similar performances whereas *Res Unrolled Net* has the best performance for SSIM comparison. On the contrary, the *U-Net Unrolled Net* obtains the best performance from LPIPS comparison, which is the Learned Perceptual Image Patch Similarity. This metric is founded on the idea of perceptual similarity, which considers how people view images. Instead of focusing on pixel-level differences between images as SSIM, LPIPS uses a deep neural network to extract high-level features and evaluates the similarities between these features in the two images [86]. For LPIPS comparison, results are multiplied by 100 to magnify differences. In LPIPS, the lowest score indicates the best result. As a result, according to LPIPS, on average, *U-Net Unrolled Net* gives the best-reconstructed images.

Table 4.3 demonstrates test results for 310×310 test images. The reason for not using 300×300 images is that the *U-Net Unrolled Net* has 6 layers for the contracting path. Before an image is used as input of the network, it is padded 5 pixels in all directions

to acquire an image with the size of 320×320. Then, after each layer operation in the contracting path, an input image is shrunk to 160×160, 80×80, 40×40, 20×20, and 10×10, one after another. Similarly, for an input image with the size of 150×150, it is expanded to 160×160 by the same padding process. Then, after each layer operation in the contracting path, it is finally diminished to 5×5.

Table 4.3 : Test results of all unrolled networks for 310×310 images.

Network	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times$100)	LPIPS ($\times$100)
U-Net Unrolled	18.20	28.83	37.99	88.16	**20.35**
Base Unrolled	18.33	28.95	38.63	88.19	21.38
Residual Unrolled	**18.65**	**29.33**	**39.43**	**88.81**	20.61

When metrics are repeated for newly defined test images, the *Res Unrolled Net* has the best performance for pSNR cases and SSIM. For pSNR comparisons, even the *Base Unrolled Net* has better performance than the *U-Net Unrolled Net*. However, they have slightly the same performance for SSIM. In LPIPS comparison, the *U-Net Unrolled Net* gives the best result.

For 310×310 test images, execution times can be obtained as in Table 4.4. For the comparison, all networks are run for 1000 times, and their average execution times are calculated. To make results concrete, all methods include the same SLM filters and noise realizations for each trial.

Table 4.4 : Execution time details of all unrolled networks for 310×310 images.

Network	Execution time (msec)
U-Net Unrolled	33.9
Base Unrolled	28.5
Residual Unrolled	28.8

The *Base Unrolled Net* has 148929 parameters, and the *Residual Unrolled Net* has 149505 parameters. On the other hand, the *U-Net Unrolled Net* requires 1944745 parameters. As a result, it needs more time to execute than the others. When it is compared to both *Base* and *Res Unrolled Nets*, it requires approximately 18.94%, and 17.70% more computation times, respectively.

In conclusion, when all models are compared for average pSNR, SSIM, and execution times, both *Base* have *Res Unrolled Nets* give better performance than *U-Net Unrolled*

Net. The only difference is seen in the LPIPS metric, which is a DL-based technique. Therefore, in Chapter 4.5.2, *U-Net Unrolled Net* results are ignored, and *Base* and *Res Unrolled Nets* are compared with PnP ADMM methods.

4.5.2 Simulated results

4.5.2.1 Simulated results without airy disc effects

In this section, two test sets with the sizes 150×150 and 300×300 are simulated without airy disc effects at 60 dB. Therefore, the *JCnR Net* is not simulated in this part, and both *Base* and *Res Unrolled Nets* are simulated for $N_{it} = 8$. Moreover, simulated PnP ADMM methods are *TV ADMM*, *TVFL1 ADMM*, and *DnCNN ADMM*. To show the baseline for methods, *LS* is also given.

All methods except *LS* are built on a CS-based algorithm. Besides, under the noiseless assumption, *LS* with $N_s = 25$ can successfully recover the image if $DSR = 5$ for both height and width. CS-based techniques, on the other hand, can sample wisely and need fewer snapshots than other techniques. As a result, N_s is chosen for simulations as 5, 10, and 15. For the *DnCNN ADMM*, prior is provided by a pre-trained network constructed on 17 layers and each layer includes 64 features [82].

A sample image for the test set of 150×150 images is shown in Figure 4.14. The size of the FPA measurements is 30×30 due to $DSR = 5$. Because of the small size of the LR image and the fact that $N_s = 5$, the reconstructed images are either blurry or unnatural to human sight. This is because *TV* and *TVFL1*, which are traditional PnP methods, generate hazy images, whereas *DnCNN* produces the sharpest image among all reconstructed techniques. If this reconstructed image is closely examined, it becomes clear that it is not natural. For instance, the cat's forehead is not perceived as natural, and its feather is so sharp. Results using *Base* and *Res Unrolled Nets* are more accurate than the result of *DnCNN*, although the cat's mustache in unrolled methods is not restored as good as *DnCNN*. The best pSNR is obtained by *Res Unrolled Net*. Yet, the best SSIM and LPIPS metric results are obtained by *DnCNN*. Generally, *DnCNN* provides a better-reconstructed image than others, but its output is fairly close to that of unrolled methods.

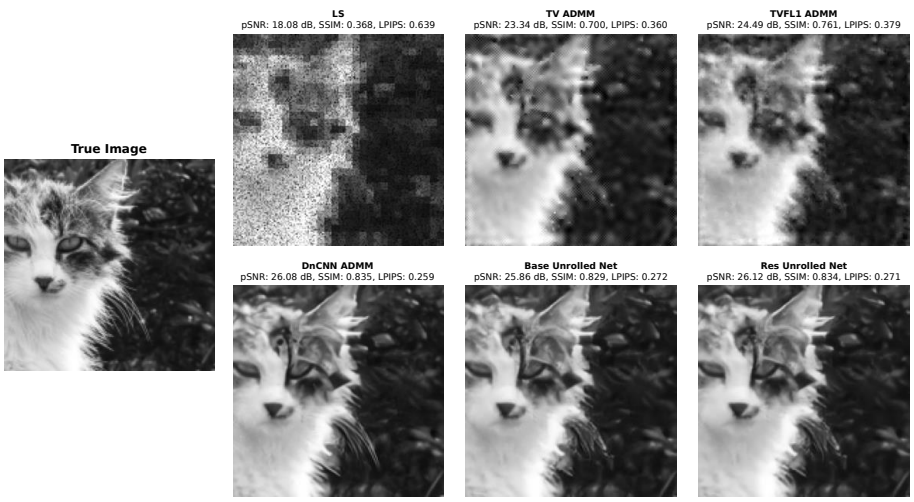

Figure 4.14 : Reconstructed 150×150 images for an airy-disc free simulation with $N_s = 5$.

If the test image is used with the size of 300×300, FPA measurements become 60×60 images. Figure 4.15 illustrates the same sample image with the size of 300×300. Since image size is increased, obtainable pSNR levels are also increased. Still, *DnCNN* produces an unnatural image if the cat's forehead is closely looked at. The *Res Unrolled Net* acquires the best image for SSIM and LPIPS results, whereas for pSNR, *DnCNN* is the winner. For human sight, unrolled methods produce more appealing results than *DnCNN*. Therefore, thanks to the increased size of the test image, unrolled methods obtain better results.

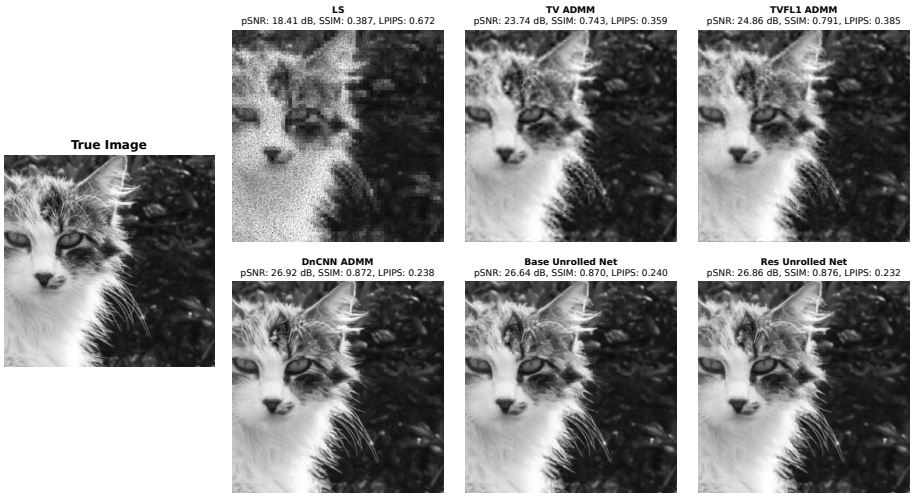

Figure 4.15 : Reconstructed 300×300 images for an airy-disc free simulation with $N_s = 5$.

In Table 4.5, image quality metric findings for a test set of 150×150 images are given. For all image quality metrics, the *Res Unrolled Net* gets the best results. In terms of average pSNR, both *Res Unrolled Net* and *DnCNN* have similar results, but when SSIM and LPIPS are also investigated for these methods, the *Res Unrolled Net* is fairly better than *DnCNN* method.

Table 4.5 : Image quality metric results for airy-disc free simulations with $N_s = 5$ in a test set of 150×150 images.

Reconstruction Method	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times$100)	LPIPS($\times$100)
LS	13.52	16.62	20.85	33.26	62.09
TV ADMM	16.41	22.19	29.22	68.17	42.14
TVFL1 ADMM	17.23	23.66	32.33	75.72	37.38
DnCNN ADMM	17.74	26.57	35.83	85.42	23.40
Base Unrolled Net	18.07	26.25	36.02	85.00	22.46
Residual Unrolled Net	**18.29**	**26.58**	**36.48**	**85.64**	**21.87**

In Table 4.6, image quality metric results are expressed for a test set of 300×300 images. Since the input image size is bigger, all methods obtain better pSNR results than 150×150 counterparts. For all image quality metrics, the *Res Unrolled Net* obtains the best results.

Table 4.6 : Image quality metric results for airy-disc free simulations with $N_s = 5$ in a test set of 300×300 images.

Reconstruction Method	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times$100)	LPIPS ($\times$100)
LS	13.99	17.28	21.09	33.46	63.41
TV ADMM	15.85	24.25	32.2	72.54	38.83
TVFL1 ADMM	16.53	25.91	34.55	79.57	34.59
DnCNN ADMM	18.27	29.1	38.74	87.97	22.95
Base Unrolled Net	18.21	28.8	38.58	88.01	21.32
Residual Unrolled Net	**18.55**	**29.17**	**39.51**	**88.61**	**20.56**

When $N_s = 10$, more information is acquired by reconstruction methods. Figure 4.16 illustrates this case for the same image with the size of 150×150. Both *Base* and *Res Unrolled Nets* have better SSIM and LPIPS performances than *DnCNN*. On the other hand, *DnCNN* obtains the best pSNR. In conclusion, both *DnCNN* and *Unrolled Nets* have very similar results.

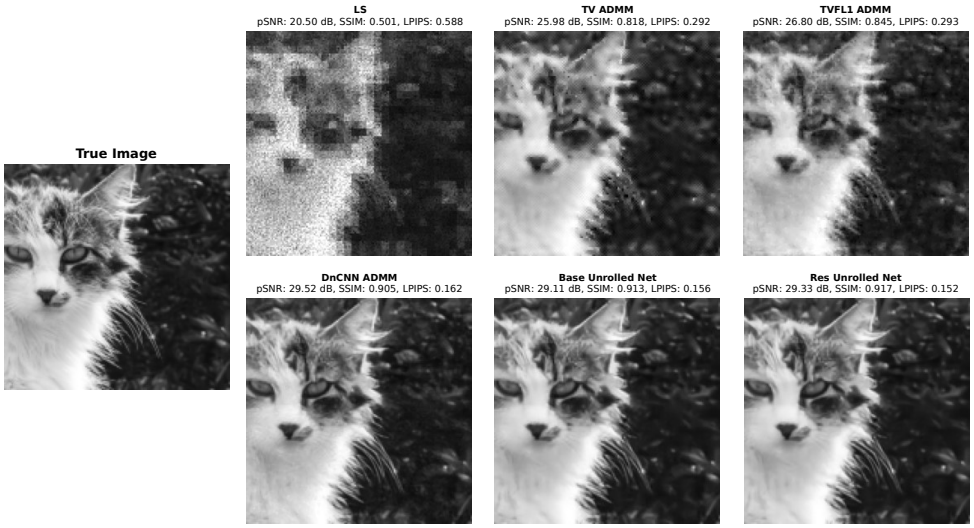

Figure 4.16 : Reconstructed 150×150 images for an airy-disc free simulation with $N_s = 10$.

Figure 4.17 shows the same sample image with the size of 300×300 for $N_s = 10$. In this case, all image quality metric results are slightly improved for *Unrolled Nets*. This is because now, models that are trained for $N_s = 10$ are used. Moreover, the test image size is increased. As a result, all methods achieve higher pSNR levels than 150×150 counterparts. However, *Unrolled Nets* attain better SSIM and LPIPS results when they are compared to *DnCNN*. Furthermore, *DnCNN* gets worse LPIPS performance than its 150×150 counterpart.

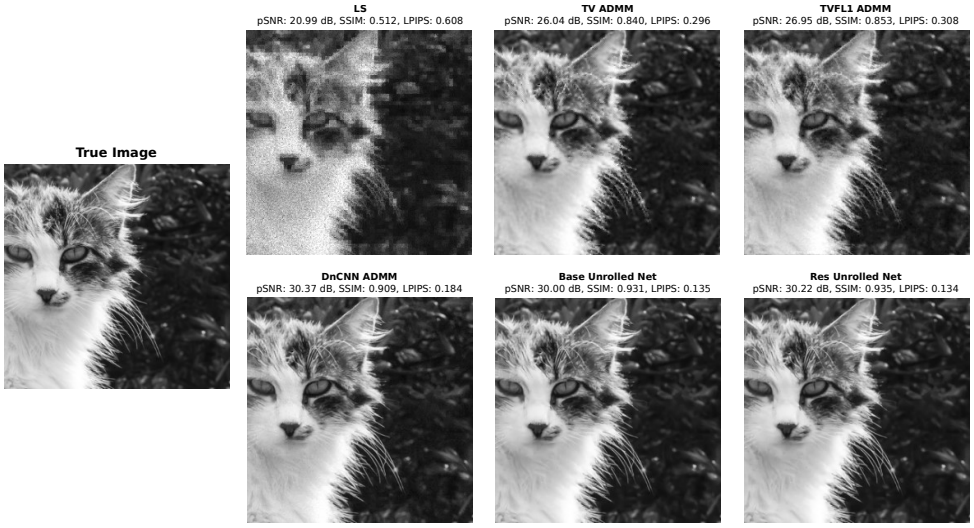

Figure 4.17 : Reconstructed 300×300 images for an airy-disc free simulation with $N_s = 10$.

In Table 4.7, image quality metrics are given for the same test set of 150×150 images. For all given metrics except Min. pSNR, the *Res Unrolled Net* gets the best results. For Min. pSNR metric, *DnCNN* achieves the best-case scenario.

Table 4.7 : Image quality metric results for airy-disc free simulations with $N_s = 10$ in a test set of 150×150 images.

Reconstruction Method	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times$100)	LPIPS ($\times$100)
LS	14.91	18.89	23.72	46.17	56.23
TV ADMM	19.05	25.38	33.05	81.93	29.71
TVFL1 ADMM	19.48	26.40	34.98	85.21	27.31
DnCNN ADMM	**21.33**	29.95	36.95	91.27	14.89
Base Unrolled Net	21.06	29.70	38.83	92.33	12.91
Residual Unrolled Net	21.24	**30.11**	**39.35**	**92.75**	**12.56**

The same simulations can be conducted for the identical test set of 300×300 images with $N_s = 10$, and Table 4.8 demonstrates image quality metric results for this case. The *Res Unrolled Net* achieves the best performance on all criteria except Min. pSNR, while *DnCNN* excels in this area.

Table 4.8 : Image quality metric results for airy-disc free simulations with $N_s = 10$ in a test set of 300×300 images.

Reconstruction Method	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times$100)	LPIPS ($\times$100)
LS	15.45	19.80	23.98	45.73	57.23
TV ADMM	17.83	27.34	34.73	83.71	28.91
TVFL1 ADMM	18.20	28.67	36.39	86.87	26.33
DnCNN ADMM	**21.46**	32.04	38.20	91.63	17.24
Base Unrolled Net	20.77	32.26	41.18	93.73	13.30
Residual Unrolled Net	20.82	**32.72**	**42.04**	**94.12**	**13.00**

For the last simulations in this section, N_s is selected as 15. As the number of information increases, it is anticipated that all methods can offer improved pSNR outcomes. Figure 4.18 shows the identical test image with the size of 150×150 for $N_s = 15$. For all metrics, both *Base* and *Res Unrolled Nets* obtain better results than *DnCNN*. Out of pSNR metric, *Unrolled Nets* have very similar results for this case. In terms of pSNR, *Res Unrolled Net* have a better result than *Base Unrolled Net*.

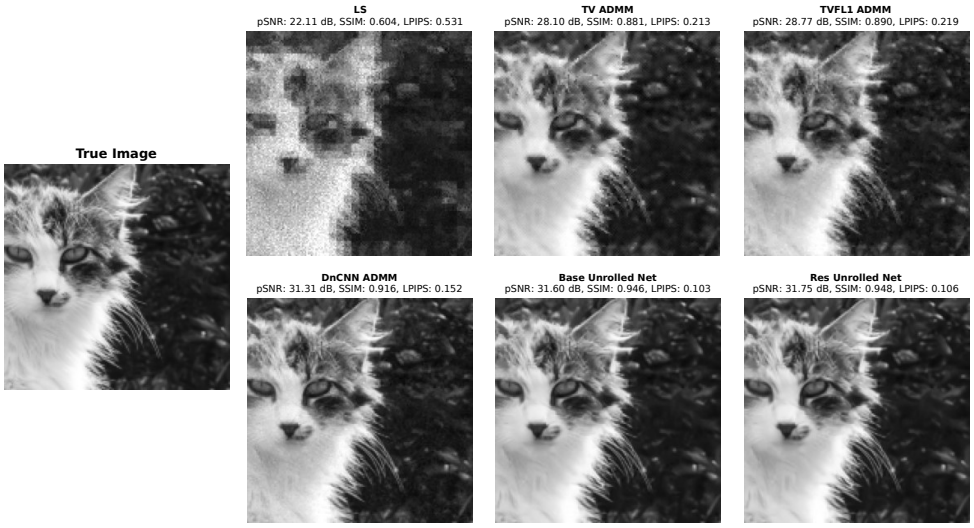

Figure 4.18 : Reconstructed 150×150 images for an airy-disc free simulation with $N_s = 15$.

If the same test image with the size of 300×300 is used, the results are seen as in Figure 4.19. Similarly, both *Base* and *Res Unrolled Nets* obtain better reconstruction results than *DnCNN*. Moreover, the LPIPS performance of *DnCNN* degrades in this case. On the other hand, *Unrolled Nets* are not affected much and have similar performances to each other.

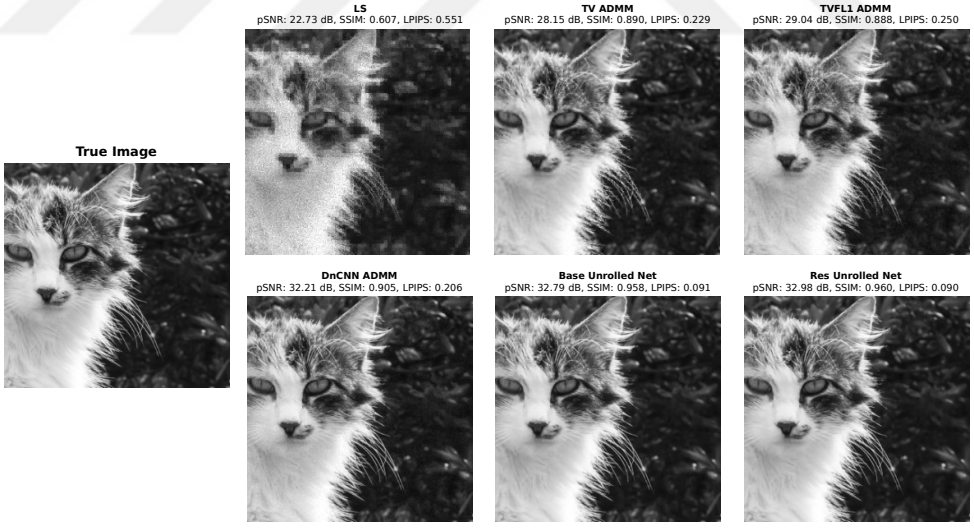

Figure 4.19 : Reconstructed 300×300 images for an airy-disc free simulation with $N_s = 15$.

Table 4.9 illustrates image quality performances for the same test set of 150×150. The *DnCNN* has only one best performance in image quality measurements, which is the Min. pSNR. *Base* and *Res Unrolled Nets* have very similar outcomes. *Base Unrolled*

Net produces the greatest results for SSIM and LPIPS, whereas *Res Unrolled Nets* produces the best average and maximum pSNRs. This is because the information is enhanced for techniques with $N_s = 15$, and the results are quite comparable in *Unrolled Nets*.

Table 4.9 : Image quality metric results for airy-disc free simulations with $N_s = 15$ in a test set of 150×150 images.

Reconstruction Method	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times 100$)	LPIPS ($\times 100$)
LS	16.14	20.53	25.59	56.76	50.37
TV ADMM	21.39	28.09	34.84	88.77	20.77
TVFL1 ADMM	21.90	28.91	35.87	90.09	20.21
DnCNN ADMM	**24.07**	31.59	36.01	92.27	13.84
Base Unrolled Net	23.65	32.43	40.04	**95.40**	**8.52**
Residual Unrolled Net	23.69	**32.60**	**40.28**	95.35	9.34

Table 4.10 illustrates image quality performances for the same test set of 300×300 images. Similar to their 150×150 counterparts, *Base* and *Res Unrolled Nets* produce the greatest results overall, with the exception of the Min. pSNR metric. All measures reach improved pSNR levels due to the increased size of the test image. *Base Unrolled Net* and *Res Unrolled Net* are the two finest for SSIM and LPIPS, respectively. On the other hand, *Res Unrolled Net* produces the best results for average and maximum pSNR measures.

Table 4.10 : Image quality metric results for airy-disc free simulations with $N_s = 15$ in a test set of 300×300 images.

Reconstruction Method	Min. pSNR (dB)	Avg. pSNR (dB)	Max. pSNR (dB)	SSIM ($\times 100$)	LPIPS ($\times 100$)
LS	16.69	21.58	25.98	55.80	51.42
TV ADMM	20.17	29.83	36.16	89.13	22.10
TVFL1 ADMM	20.44	30.91	37.25	90.36	21.06
DnCNN ADMM	**24.09**	32.97	37.75	91.17	17.82
Base Unrolled Net	23.42	34.79	42.84	**96.01**	**9.84**
Residual Unrolled Net	23.44	**34.90**	**43.67**	95.90	10.60

Before concluding the airy-disc free simulations, Table 4.11 demonstrates average execution times for both 150×150 and 300×300 test images after 1000 trials. It is clear that *Base* and *Res Unrolled Nets* achieve performance at least equal to that of *DnCNN* with less required execution time. The reason is that *Unrolled Nets* use 8 iterations, whereas *DnCNN* iterates 30 times. Since the pre-trained model of the *DnCNN* has 17 layers and 64 features, using 30 iterations leads to slow execution time compared to *Unrolled Nets* with $N_{it} = 8$. Both *TV* and *TVFL1* use 60 iterations, and

when image size is increased to 300×300, *DnCNN* gets the worst average execution time even if both *TV* and *TVFL1* include more iterations than *DnCNN*.

Table 4.11 : Average execution time results of all reconstruction methods except *LS* for 150×150 and 300×300 images.

Reconstruction Method	Execution time for 150×150 (msec)	Execution time for 300×300 (msec)
TV ADMM	172.6	197.8
TVFL1 ADMM	184.9	208.2
DnCNN ADMM	101.8	284.2
Base Unrolled Net	**12.5**	**28.0**
Residual Unrolled Net	12.7	29.0

4.5.2.2 Simulated results with airy disc effects

In this section, only a test set of 300×300 images is presented at 60 dB with airy disc effects.

In the previous part, traditional PnP approaches like *TV* and *TVFL1* did not outperform *DnCNN* and *Unrolled Nets*. Thus, in this section, these methods are not given as competitive methods. Furthermore, both *Base* and *Res Unrolled Nets* obtain very similar results. Therefore, only *Res Unrolled Net* is used.

For the *JCnR Net*, the reconstruction part of the model has the same architecture of *Res Unrolled Net*. All reconstruction methods except the *JCnR Net* have a calibration process outside of their methods. This is because the *JCnR Net* includes the calibration process is in its inside before the reconstruction step. On the other hand, both *DnCNN* and *Res Unrolled Net* require calibration network outputs as inputs of their techniques. The calibration model is *Unshuffled SLM Calibration Net* with a selective one-hot encoded ADR range input. The calibration network has a $5/64$ configuration to keep similar to a calibration model in the *JCnR Net*. However, as explained in Chapter 4.3.3, to get better models, each ADR range has its *JCnR Net*. These are the differences between calibration methods.

For simulations, N_s is selected as 5, 10, and 15. Since there are 9 ADR ranges, to make results more attractive, only 3 ADRs are selected as 2, 5, and 8 for image comparisons. Moreover, in image comparisons, only 300×300 images are used, and overall image quality metric results of all ADRs are given in table formats for 300×300. Since these

simulations contain calibration networks in default, both *Calib Net + DnCNN ADMM* and *Calib Net + Res Unrolled Net* in image comparisons are shortened as *DnCNN* and *Res Unrolled Net* in text, respectively. In the same way, *Joint Calib and Recon Net* in images are abbreviated as the *JCnR Net* in text.

Figure 4.20 demonstrates reconstruction results for $N_s = 5$ and $ADR = 2$. The *Raw LS* represents an LS solution without using any calibration process. After using the calibration network, the LS solution is slightly improved. The *JCnR Net* produces the best result in terms of given image quality metrics, which are pSNR, SSIM, and LPIPS. When the car's plate is closely investigated, both *Res Unrolled Net* and *JCnR Net* are better. Since $ADR = 2$, all models use a calibration model with a $1.5 - 2.5$ ADR range, and this is the easiest range in all ADR ranges given in Table 3.1.

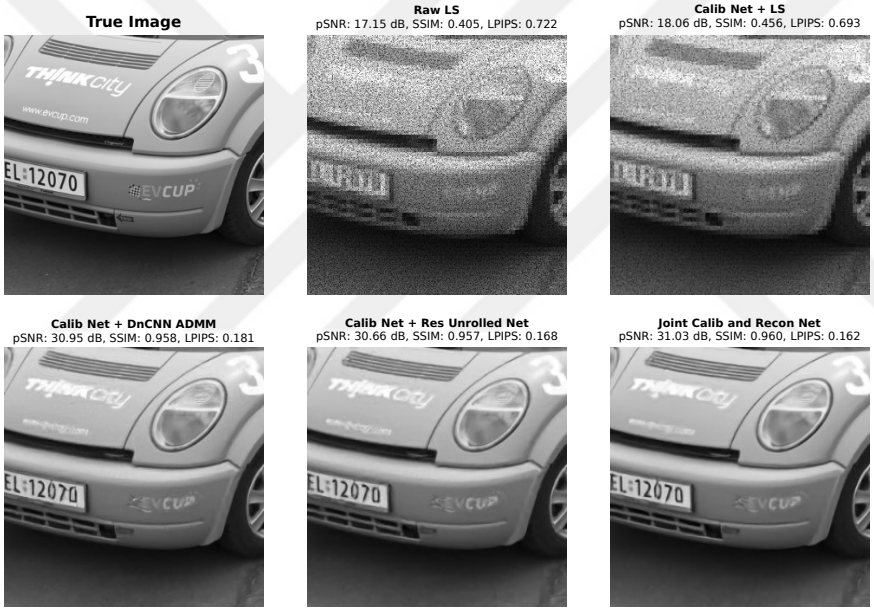

Figure 4.20 : Reconstructed 300×300 images for $ADR = 2$ simulation with $N_s = 5$.

When $ADR = 5$, the reconstruction outcomes for $N_s = 5$ are presented in Figure 4.21, representing a more challenging scenario. The *JCnR Net* achieves the best metric results. The plate in *DnCNN* case is seen as the best, but *DnCNN* produces some artifacts on the bonnet of the car. One is on the left of the lamb and the other one is below the website on the bonnet. However, the *JCnR Net* does not produce such artificial effects. As a result, the *JCnR Net* attains a more robust result for this case.

Figure 4.21 : Reconstructed 300×300 images for $ADR = 5$ simulation with $N_s = 5$.

Figure 4.22 gives the reconstruction results for $ADR = 8$ with $N_s = 5$. The plate in the image of *JCnR Net* is the most readable one. Moreover, both *DnCNN* and *Res Unrolled Net* results have artifacts around the lamb and the hubcap of the car. On the contrary, the *JCnR Net* output has no such artifacts. The borders of the car are the most natural in the output of the *JCnR Net*. The number above the lamb in *JCnR Net* output also is the sharpest in all methods. As a result, for all given image quality metrics, the *JCnR Net* gets the best result in them.

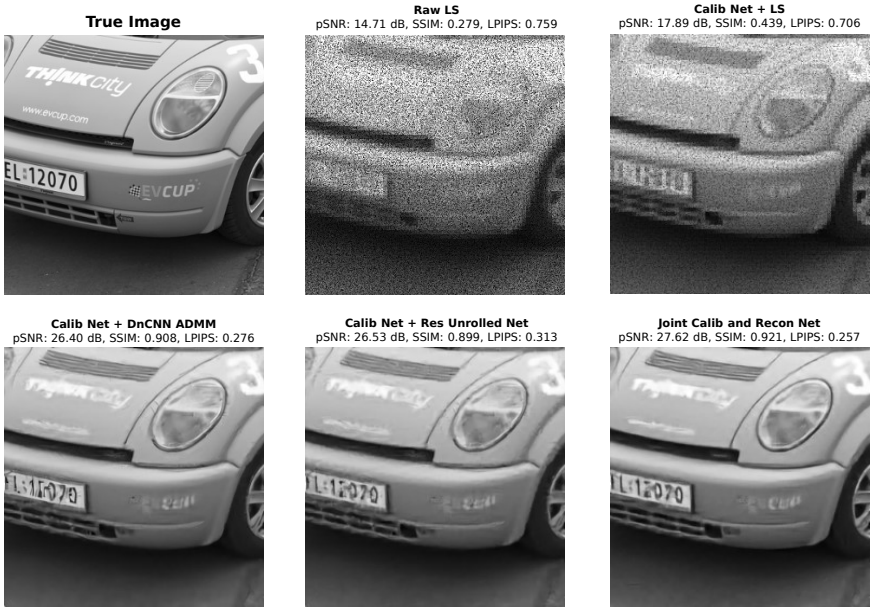

Figure 4.22 : Reconstructed 300×300 images for $ADR = 8$ simulation with $N_s = 5$.

In Table 4.12, average pSNR results for each related ADR are calculated in the test set of 300×300 images when $N_s = 5$. For each ADR, the *JCnR Net* obtains the best results in terms of pSNR. Moreover, for bigger simulated ADR, the difference between *JCnR Net* and *DnCNN* increases. It implies that using a calibration network as a part of a whole model improves the model output's performance. Both *Res Unrolled Net* and *DnCNN* have very similar average results until $ADR = 7$. Then, for bigger ADRs, *Res Unrolled Net* gets better results than *DnCNN*.

Table 4.12 : Average pSNR (dB) results for all ADRs when $N_s = 5$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	16.45	15.88	15.49	15.07	14.78	14.42	14.17	13.94	13.69
Calib Net + LS	17.25	17.22	17.21	17.15	17.12	17.11	17.07	17.00	16.95
Calib Net + DnCNN ADMM	27.74	27.15	26.66	26.30	25.91	25.45	24.75	24.01	23.21
Calib Net + Res Unrolled Net	27.74	27.14	26.66	26.33	25.96	25.56	24.98	24.36	23.75
Joint Calib and Recon Net	**27.87**	**27.52**	**27.17**	**26.89**	**26.62**	**26.30**	**25.94**	**25.53**	**25.05**

In Table 4.13, average SSIM results are multiplied by 100 to make them more interpretable for $N_s = 5$. These results are calculated for each related ADR in the same test set of 300×300 images. The *JCnR Net* achieves the best results for all ADRs. Similar to the average pSNR metric in Table 4.12, the difference between *DnCNN* and *JCnR Net* is raised in terms of SSIM ($\times 100$). In fact, for bigger ADRs, the differences are also bigger. The *Res Unrolled Net* performance is similar to *DnCNN* performance, but slightly, *DnCNN* attains better results than *Res Unrolled Net*.

Table 4.13 : Average SSIM ($\times 100$) results for all ADRs when $N_s = 5$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	41.1	38.0	36.2	34.1	32.0	30.4	29.1	27.2	26.0
Calib Net + LS	45.8	45.4	45.1	44.6	44.3	44.0	43.3	42.6	41.8
Calib Net + DnCNN ADMM	**90.5**	89.5	88.5	87.6	86.7	85.5	83.7	81.4	78.9
Calib Net + Res Unrolled Net	90.3	89.2	88.1	87.3	86.3	85.2	83.5	81.4	79.2
Joint Calib and Recon Net	**90.5**	**89.8**	**89.2**	**88.4**	**87.7**	**87.0**	**86.0**	**84.7**	**83.2**

Table 4.14 gives LPIPS results multiplied by 100 for $N_s = 5$. Similarly, these results are computed for each corresponding ADR in the same test set of 300×300 images. For the first two ADRs, 2 and 3, the *JCnR Net* does not get the best results. When $ADR = 2$, *Res Unrolled Net* obtains the best LPIPS results. Besides, for $ADR = 3$, *DnCNN* achieves the best result. However, for the rest of the ADRs, the *JCnR Net* attains the best results. Since this metric is a DL-based technique, these results may be expected. Yet, when ADR is increasing, the *JCnR Net* gets consistently the best results, which indicates that using a joint calibration and reconstruction process as a model improves reconstruction results in terms of LPIPS.

Table 4.14 : Average LPIPS ($\times 100$) results for all ADRs when $N_s = 5$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	66.1	67.1	68.2	69.3	70.2	71.0	71.4	72.0	72.4
Calib Net + LS	63.7	63.8	64.1	64.2	64.3	64.5	64.8	65.3	65.5
Calib Net + DnCNN ADMM	25.7	**26.7**	28.3	29.7	31.2	32.9	35.3	38.1	40.6
Calib Net + Res Unrolled Net	**25.2**	26.9	28.6	30.0	31.6	33.3	35.7	38.4	40.8
Joint Calib and Recon Net	26.1	27.0	**27.8**	**29.1**	**30.1**	**31.2**	**32.3**	**33.9**	**35.8**

Figure 4.23 illustrates the reconstruction results for $ADR = 2$ with $N_s = 10$. Since N_s increases, the plate is more readable for calibrated *LS* result. For the metrics, the *JCnR Net* outcomes are the best-reconstructed image. The plate and borders on it are the sharpest among the models. Furthermore, The difference in LPIPS results is the biggest in metric differences. This is because *DnCNN* gives a fuzzy image if it is zoomed. As a result, from the point of human-sight perspective, the *JCnR Net* produces a more preferable image.

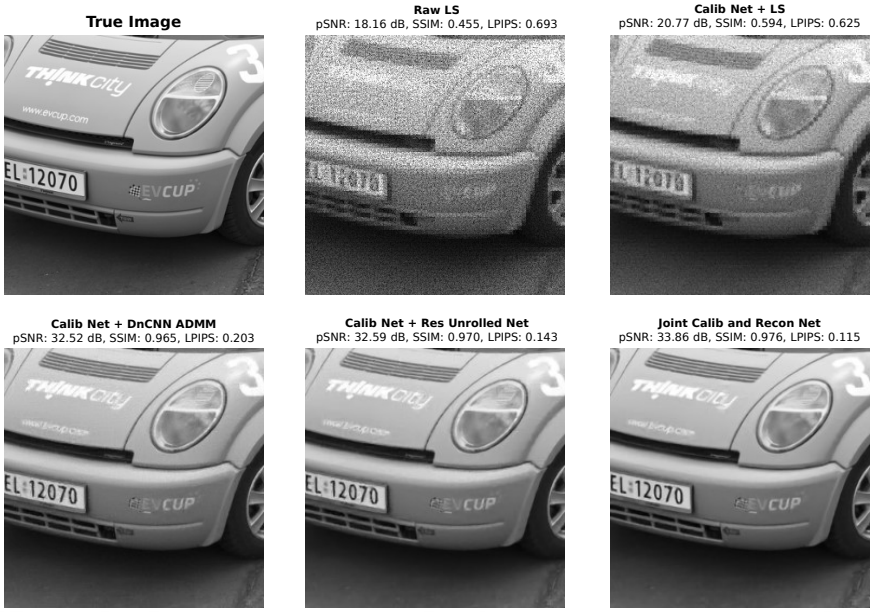

Figure 4.23 : Reconstructed 300×300 images for $ADR = 2$ simulation with $N_s = 10$.

Figure 4.24 indicates the reconstruction results for $ADR = 5$ with $N_s = 10$. When airy disc effects are stronger, metric results are degraded. However, the performance decline in the *JCnR Net* is not as big as other techniques like *DnCNN*. The plate on *JCnR Net* is preferable among them, and the metric results of the model are better than others.

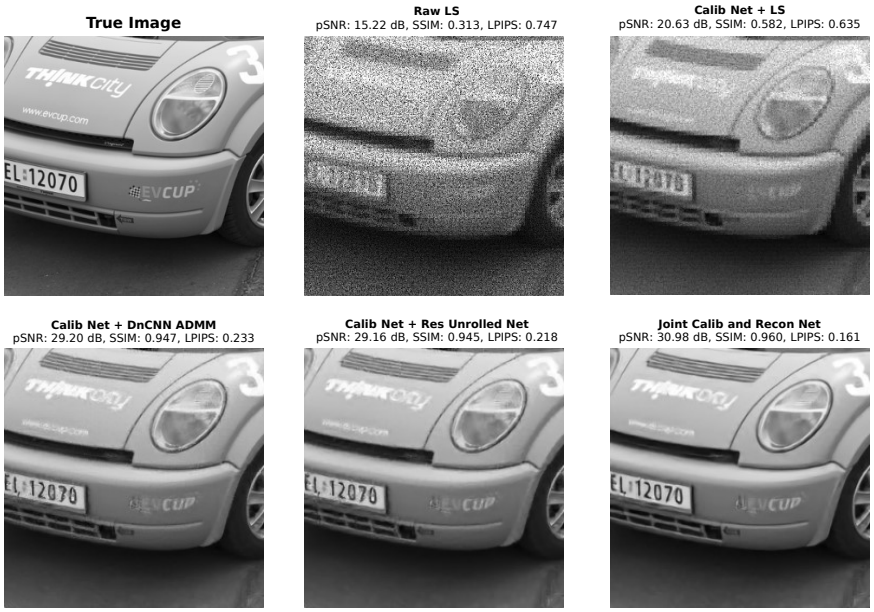

Figure 4.24 : Reconstructed 300×300 images for $ADR = 5$ simulation with $N_s = 10$.

In Figure 4.25, for $N_s = 10$, all models use $ADR = 8$ to simulate airy disc effects. The ADR deterioration affects both *DnCNN* and *Res Unrolled Net* more than the *JCnR Net*. This is because the calibration and reconstruction processes of *JCnR Net* are connected as an optimization problem in the training step. Furthermore, the model loss not only changes the weights of the calibration step but also affects the reconstruction weights. As a result, the overall model gets more robust results even if ADR is larger.

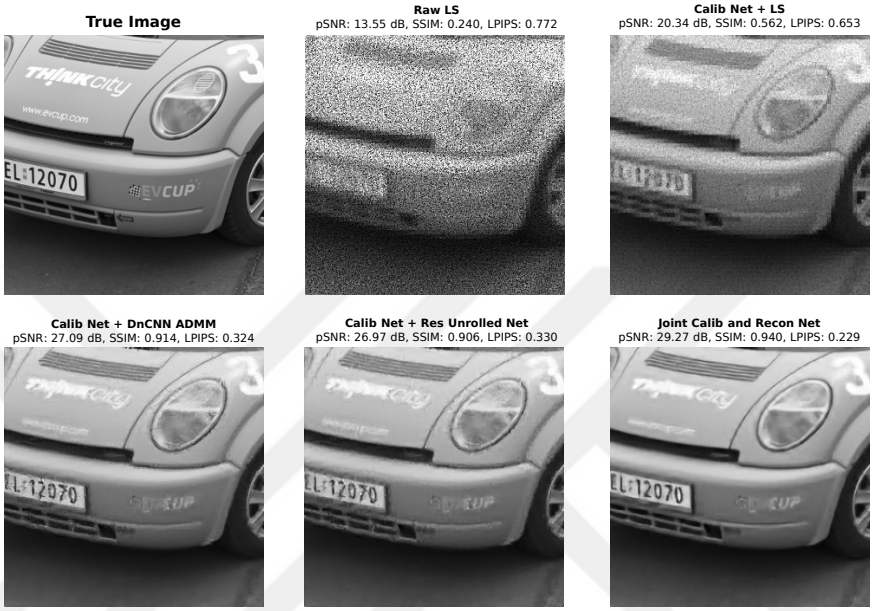

Figure 4.25 : Reconstructed 300×300 images for $ADR = 8$ simulation with $N_s = 10$.

In Table 4.15, average pSNR levels of the reconstruction methods are given for each ADR when $N_s = 10$. The average pSNR results imply that the *JCnR Net* produces the best-reconstructed images for all ADRs. The difference between the model and other methods gradually increases when ADR is bigger. If results of *Raw LS* for $Ns = 5$ and $N_s = 10$ are compared to each other, using more snapshots leads to pSNR degradation for high ADRs. More airy disc-corrupted LR images are taken, and their direct combination in the *Raw LS* process leads to getting weak reconstruction performances. On the other hand, using a calibration network gives benefits to LS reconstruction to produce better reconstruction results with higher pSNR levels. This shows that the calibration network is critical for the reconstruction process.

Table 4.15 : Average pSNR (dB) results for all ADRs when $N_s = 10$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	17.44	16.21	15.37	14.75	14.19	13.66	13.23	12.95	12.67
Calib Net + LS	19.67	19.62	19.54	19.52	19.44	19.36	19.25	19.10	18.93
Calib Net + DnCNN ADMM	29.47	28.58	27.87	27.35	26.77	26.18	25.31	24.39	23.43
Calib Net + Res Unrolled Net	29.61	28.58	27.83	27.31	26.74	26.15	25.30	24.42	23.57
Joint Calib and Recon Net	**30.32**	**29.53**	**29.01**	**28.57**	**28.08**	**27.59**	**27.01**	**26.44**	**25.80**

In Table 4.16, for $N_s = 10$, SSIM results are multiplied by 100 to represent the results in a more readable format. Like average pSNR metric results in Table 4.15, average SSIM results point out that the *JCnR Net* has the best-reconstructed images for all ADRs.

Table 4.16 : Average SSIM ($\times 100$) results for all ADRs when $N_s = 10$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	47.1	41.4	37.2	34.2	31.4	28.3	26.2	24.2	23.1
Calib Net + LS	57.7	57.2	56.5	56.1	55.4	54.7	53.6	52.3	50.8
Calib Net + DnCNN ADMM	93.5	92.5	91.4	90.4	89.2	87.8	85.7	83.1	80.1
Calib Net + Res Unrolled Net	93.7	92.4	91.1	90.1	88.8	87.4	85.1	82.4	79.4
Joint Calib and Recon Net	**94.4**	**93.4**	**92.7**	**92.1**	**91.1**	**90.1**	**88.8**	**87.4**	**85.7**

LPIPS results for $N_s = 10$ are given in Table 4.17. When LPIPS results for $N_s = 5$ and $N_s = 10$ are examined, *Res Unrolled Net* with $N_s = 10$ performs worse than the same method with $N_s = 5$ for bigger ADRs like 8,9, and 10. This is not seen for *DnCNN*, and it indicates that *Res Unrolled Net* is affected more badly than *DnCNN* from larger ADRs. Although the calibration network is used, to get better results from *Res Unrolled Net*, eps should be selected for slightly higher LR pSNR levels. In terms of data fidelity, this encourages the network to get closer to the forward model. The *JCnR Net*, on the other hand, achieves the best performance for all ADRs by combining both the calibration and reconstruction networks with marginally selected lower eps values as in Table 4.1.

Table 4.17 : Average LPIPS ($\times 100$) results for all ADRs when $N_s = 10$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	64.2	67.3	68.4	69.4	70.3	71.2	72.1	73.0	73.4
Calib Net + LS	58.3	58.7	59.0	59.3	59.8	60.2	60.8	61.5	62.4
Calib Net + DnCNN ADMM	20.9	21.8	23.4	25.3	27.4	29.8	33.1	36.4	39.7
Calib Net + Res Unrolled Net	19.8	22.3	24.7	26.8	29.4	32.2	35.8	39.5	42.8
Joint Calib and Recon Net	**18.7**	**21.0**	**22.4**	**23.7**	**25.1**	**27.0**	**28.8**	**30.9**	**33.0**

Figure 4.26 includes the reconstructed images for $ADR = 2$ with $N_s = 15$. In such a scenario, the *JCnR Net* reconstructs the best image in terms of given image quality metrics.

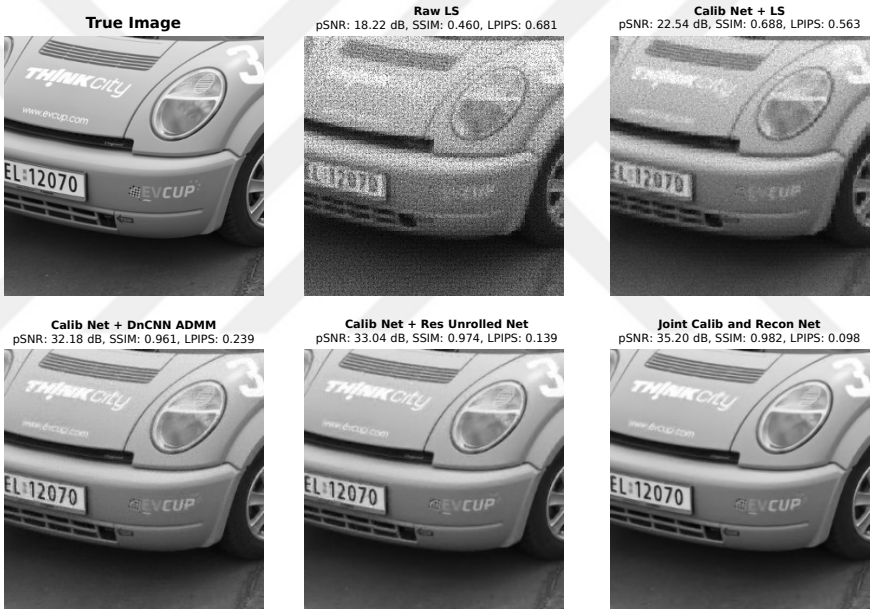

Figure 4.26 : Reconstructed 300×300 images for $ADR = 2$ simulation with $N_s = 15$.

In Figure 4.27, the reconstructed images are obtained for $ADR = 5$ with $N_s = 15$. The airy disc effects are amplified in this situation, although the *JCnR Net* is only marginally damaged compared to other techniques. In terms of image quality metrics and human-eye perspective, the model differs significantly from other methods. As an example, if the plates are investigated, the *JCnR Net* gives the most preferable result.

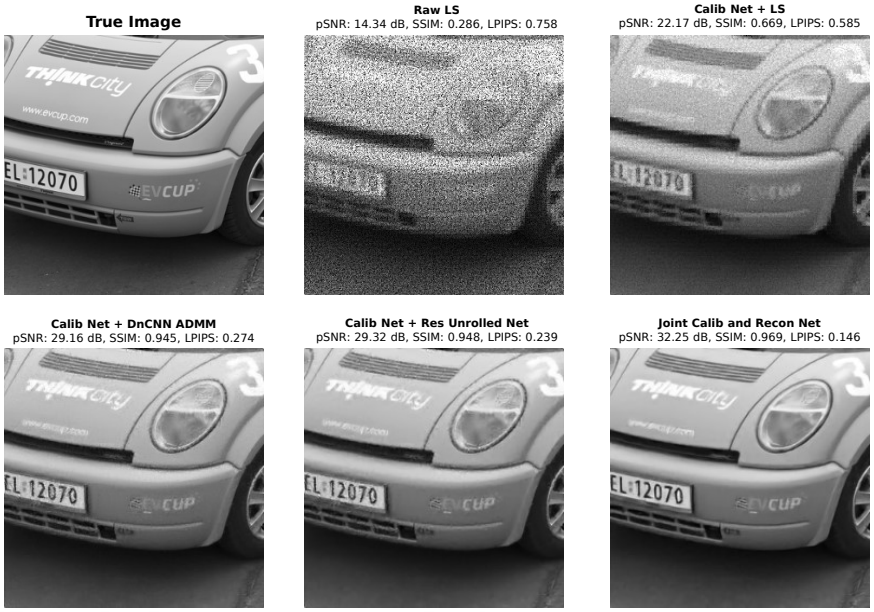

Figure 4.27 : Reconstructed 300×300 images for $ADR = 5$ simulation with $N_s = 15$.

Figure 4.28 shows the reconstructed images for $ADR = 8$ with $N_s = 15$. This is one of the hardest cases of airy disc effects in simulations. Thanks to relatively high N_s, all methods that use calibration processes get a reasonable solution to read the plates in them. In the *JCnR Net*, the plate is the most pleasing. Moreover, it produces the sharpest image if the hubcaps of cars are closely examined. These visual differences support the image quality metric results of the method.

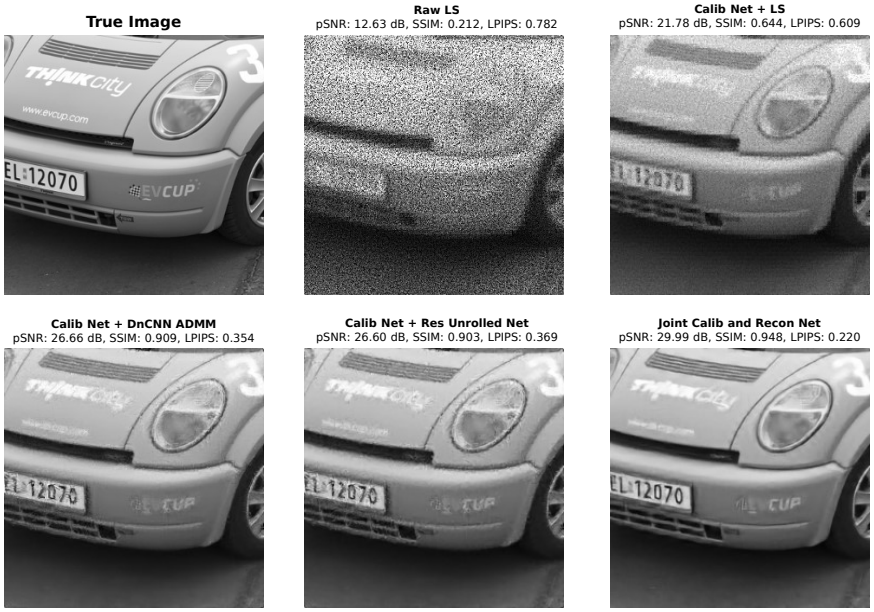

Figure 4.28 : Reconstructed 300×300 images for $ADR = 8$ simulation with $N_s = 15$.

In Table 4.18, average pSNR results are calculated for all ADRs when $N_s = 15$. For all ADRs, the *JCnR Net* achieves the highest pSNR levels among all given reconstruction methods. *Res Unrolled Net* gets better results than *DnCNN* for the first five ADR cases. *Raw LS* shows that higher ADRs make more trouble for the reconstruction process. However, using a calibration network improves reconstruction outputs. These results imply that a combined network including both calibration and reconstruction processes obtains a more robust structure even for higher ADRs.

Table 4.18 : Average pSNR (dB) results for all ADRs when $N_s = 15$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	17.56	15.86	14.84	14.02	13.40	12.82	12.35	11.98	11.66
Calib Net + LS	21.30	21.20	21.07	20.99	20.85	20.74	20.51	20.25	19.98
Calib Net + DnCNN ADMM	29.90	28.97	28.15	27.60	26.91	26.25	25.25	24.26	23.26
Calib Net + Res Unrolled Net	30.47	29.22	28.28	27.67	26.94	26.22	25.14	24.09	23.07
Joint Calib and Recon Net	**31.70**	**30.71**	**30.07**	**29.54**	**28.96**	**28.29**	**27.72**	**27.00**	**26.31**

In Table 4.19, average SSIM results are multiplied by 100 to make differences more distinguishable for $N_s = 15$. The *JCnR Net* obtains the highest SSIM results for all ADRs. The difference between the model and others increases while airy disc effects are increased by ADR. This result shows that only the use of *Res Unrolled Net* provides us to get similar results with *DnCNN* in a shorter execution time. However, a calibration model as a part of *JCnR Net* improves both the output's robustness to airy disc effects and image quality metric results.

Table 4.19 : Average SSIM ($\times 100$) results for all ADRs when $N_s = 15$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	48.2	41.4	36.1	32.2	29.1	26.2	24.1	22.3	20.2
Calib Net + LS	66.5	65.8	64.9	64.2	63.2	62.3	60.6	58.7	56.6
Calib Net + DnCNN ADMM	94.2	93.3	92.2	91.1	89.8	88.3	85.8	83.0	79.7
Calib Net + Res Unrolled Net	94.9	93.6	92.3	91.2	89.7	88.0	85.3	82.0	78.3
Joint Calib and Recon Net	**95.9**	**95.1**	**94.3**	**93.6**	**92.8**	**91.7**	**90.5**	**88.9**	**87.3**

Table 4.20 gives average LPIPS results multiplied by 100 for $N_s = 15$. The results indicate that the *JCnR Net* has the best reconstruction results. For all ADRs, it produces the smallest LPIPS scores among the methods. Similar to SSIM results given in Table 4.19, the difference between the model and other methods increases while ADR is increased.

Table 4.20 : Average LPIPS ($\times 100$) results for all ADRs when $N_s = 15$ in a test set of 300×300 images.

Recon. Method	ADR								
	2	3	4	5	6	7	8	9	10
Raw LS	63.1	67.4	69.2	70.1	71.1	72.2	73.0	73.4	74.0
Calib Net + LS	53.4	54.0	54.8	55.4	56.1	56.8	57.9	59.1	60.4
Calib Net + DnCNN ADMM	21.4	22.0	23.8	25.7	28.2	30.7	34.5	38.0	41.5
Calib Net + Res Unrolled Net	18.4	21.2	24.2	26.7	29.8	33.0	37.4	41.4	45.0
Joint Calib and Recon Net	**15.8**	**17.4**	**19.5**	**20.7**	**22.6**	**24.9**	**26.5**	**29.0**	**31.4**

In conclusion, airy disc effects have a significant impact on the quality of a reconstruction process. Employing a calibration model increases input quality for reconstruction approaches, although bigger ADRs restrict this advantage. Furthermore, a network that includes both calibration and reconstruction processes, as well as an optimization challenge in loss definition, enhances model outputs. As a result, the model delivers more resilient responses to airy disc effects, and image quality metric scores demonstrate that model outputs perform better.

5. CONCLUSIONS

In this thesis, we have presented a joint calibration and reconstruction model for FPA imaging systems. First, to calibrate faulty FPA measurements of FPA detectors, we developed an online calibration model. Second, to improve super-resolution performances, we designed an unrolling-based model on the iterative structure of ADMM. Instead of using a hand-crafted or model-based prior function, we used a CNN block to learn a prior model of the original signal. Lastly, we constructed a joint model including both the calibration network and unrolling-based super-resolution model. For the joint model, we have presented results in comprehensive simulations for different N_s scenarios. Moreover, we ran the simulations multiple times to achieve a variety of airy disc effects with ADR.

For the calibration of airy disc-corrupted LR images, we have shown that an online calibration model reduces airy disc effects. To achieve this, we used SLM filters as one of the inputs of the model. For comparisons, models with and without SLM input are investigated. Furthermore, we added a calibration method using exact PSFs to compare learning-based techniques with a conventional deconvolution technique. We obtained average pSNR results of techniques for each ADR range. We have demonstrated that a model that incorporates SLM information is superior to other methods.

For the reconstruction, the ADMM-based unrolled network demonstrates comparable performance to the *DnCNN ADMM* method. The unrolling-based method employs an iterative process, but the number of iterations of the model is fewer than the number of iterations of the *DnCNN ADMM* method. As a result, in a shorter time, the unrolling-based technique achieves similar performances. For numerical results, we have obtained average pSNR, SSIM, and LPIPS results. Then, we demonstrated that an unrolling-based method attains better results than PnP ADMM techniques for different numbers of snapshots.

For the last step of the thesis, we combined both calibration network and unrolling-based reconstruction. To train the whole model, we defined a combined loss that takes into account both the LR image and the HR image. The proposed approach involves the utilization of two-loss coefficients. Initially, emphasis is placed on the LR loss, followed by a gradual increase in the priority of the HR loss until a predetermined stopping threshold is attained. For different numbers of snapshots, we presented numerical and visual findings. The joint model produces the best results in terms of average pSNR, SSIM, and LPIPS for all ADR ranges, according to numerical results. The joint model, in particular, makes up the difference in numerical and visual performance in bigger ADR ranges. This means that combining LR and HR losses improves the joint model's stability, and this impact is particularly pronounced in bigger ADR simulations.

As a possible direction for further research, the joint model can be modified to accommodate FPA video processing. The joint model's calibration network can be modified to incorporate a multi-snapshot model, thereby enhancing the accuracy of corrected FPA measurements. Furthermore, the calibration process can generate an HR image that can serve as an initial reference point in the unrolling-based reconstruction procedure of the joint model. By implementing these modifications, it can be possible to enhance the model's performance in relation to image quality metrics or reduce the number of iterations required for the model while maintaining similar image quality performances.

REFERENCES

[1] **Gibson, G.M.**, **Johnson, S.D. and Padgett, M.J.** (2020). Single-pixel imaging 12 years on: a review, *28*(19).

[2] **Radwell, N.**, **Mitchell, K.J.**, **Gibson, G.M.**, **Edgar, M.P.**, **Bowman, R. and Padgett, M.J.** (2014). Single-pixel infrared and visible microscope, *1*(5).

[3] **Edgar, M.P.**, **Gibson, G.M.**, **Bowman, R.**, **Sun, B.**, **Radwell, N.**, **Mitchell, K.J.**, **Welsh, S.S. and Padgett, M.J.** (2015). Simultaneous real-time visible and infrared video with single-pixel detectors, *5*(1).

[4] **Gibson, G.M.**, **Sun, B.**, **Edgar, M.P.**, **Phillips, D.B.**, **Hempler, N.**, **Maker, G.T.**, **Malcolm, G.P.A. and Padgett, M.J.** (2017). Real-time imaging of methane gas leaks using a single-pixel camera., *25*(4).

[5] **Hadfield, R.H.** (2009). Single-photon detectors for optical quantum information applications, *3*(12).

[6] **Ma, J.** (2009). A Single-Pixel Imaging System for Remote Sensing by Two-Step Iterative Curvelet Thresholding, *6*(4).

[7] **Shin, J.**, **Bosworth, B.T. and Foster, M.A.** (2016). Single-pixel imaging using compressed sensing and wavelength-dependent scattering., *41*(5).

[8] **Sun, M.J.**, **Edgar, M.P.**, **Gibson, G.M.**, **Sun, B.**, **Radwell, N.**, **Lamb, R.A. and Padgett, M.J.** (2016). Single-pixel three-dimensional imaging with time-based depth resolution, *7*(1).

[9] **Shi, D.**, **Yin, K.**, **Huang, J.**, **Huang, J.**, **Yuan, K.**, **Zhu, W.**, **Xie, C.**, **Liu, D.**, **Wang, Y. and Wang, Y.** (2019). Fast tracking of moving objects using single-pixel imaging, *440*.

[10] **Shrekenhamer, D.**, **Watts, C.M. and Padilla, W.J.** (2013). Terahertz single pixel imaging with an optically controlled dynamic spatial light modulator, *21*(10).

[11] **Donoho, D.** (2004). *Compressed sensing*.

[12] **Candès, E.J.**, **Romberg, J. and Tao, T.** (2006). Stable signal recovery from incomplete and inaccurate measurements, *59*(8).

[13] **Baraniuk, R.G.**, **Davenport, M.A.**, **DeVore, R.A. and Wakin, M.B.** (2008). A Simple Proof of the Restricted Isometry Property for Random Matrices, *28*(3).

[14] **Duarte, M.F.**, **Davenport, M.A.**, **Takhar, D.**, **Laska, J.N.**, **Sun, T.**, **Kelly, K.F. and Baraniuk, R.G.** (2008). Single-Pixel Imaging via Compressive Sampling, *25*(2).

[15] **Chen, H.**, **Asif, M.S.**, **Sankaranarayanan, A.C. and Veeraraghavan, A.** (2015). FPA-CS: Focal plane array-based compressive imaging in short-wave infrared, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2358–2366.

[16] **Hopkins, H.H.** (1943). The airy disc formula for systems of high relative aperture, *55*(2).

[17] **Mahalanobis, A.**, **Shilling, R.Z.**, **Murphy, R. and Muise, R.** (2014). Recent results of medium wave infrared compressive sensing, *53*(34).

[18] **Jin, X.P.**, **Zhao, Q.**, **Liu, X.F. and Xiong, A.D.** (2021). *Mid-wave infrared super-resolution imaging based on compressive calibration and sampling*, `2109.04887`.

[19] **Smith, B.**, **Hellman, B.**, **Gin, A.**, **Espinoza, A. and Takashima, Y.** (2017). Single chip lidar with discrete beam steering by digital micromirror device, *25*(13).

[20] **Valencia, A.**, **Scarcelli, G.**, **D'Angelo, M. and Shih, Y.** (2005). Two-photon imaging with thermal light., *94*(6).

[21] **Katz, O.**, **Bromberg, Y. and Silberberg, Y.** (2009). Compressive ghost imaging, *95*(13).

[22] **Xu, Z.H.**, **Chen, W.**, **Penuelas, J.**, **Padgett, M.J. and Sun, M.J.** (2018). 1000 fps computational ghost imaging using LED-based structured illumination, *26*(3).

[23] **Necipoglu, S.**, **Cebeci, S.A.**, **Basdogan, C.**, **Has, Y.E. and Guvenc, L.** (2011). Repetitive control of an XYZ piezo-stage for faster nano-scanning: Numerical simulations and experiments, *21*(6).

[24] **Li, X.**, **Qi, N.**, **Jiang, S.**, **Wang, Y.**, **Li, X. and Sun, B.** (2020). Noise Suppression in Compressive Single-Pixel Imaging, *Sensors*, *20*(18), `https://www.mdpi.com/1424-8220/20/18/5341`.

[25] **Kar, O.F.**, **Güngör, A.**, **İlbey, S. and Güven, H.E.** (2018). An efficient parallel algorithm for single-pixel and FPA imaging, *Computational Imaging III*, volume10669, International Society for Optics and Photonics, SPIE, p.106690J, `https://doi.org/10.1117/12.2304342`.

[26] **Born, M. and Wolf, E.** (1999). *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, Cambridge University Press.

[27] **Airy, G.B.** (1835). On the diffraction of an object-glass with circular aperture, *Transactions of the Cambridge Philosophical Society*, *5*, 283.

[28] **Wu, Z. and Wang, X.** (2019). Focal plane array-based compressive imaging in medium wave infrared: modeling, implementation, and challenges, *58*(31).

[29] **Cao, Q.** (2003). Generalized Jinc functions and their application to focusing and diffraction of circular apertures, *J. Opt. Soc. Am. A*, *20*(4), 661–667, `https://opg.optica.org/josaa/abstract.cfm?URI=josaa-20-4-661`.

[30] **Astropy**. *AiryDisk2DKernel*, `https://docs.astropy.org/en/stable/api/astropy.convolution.AiryDisk2DKernel.html`.

[31] **Hansen, P.C.** (2010). *Discrete Inverse Problems: Insight and Algorithms*.

[32] **Tikhonov, A.N.**, **Arsenin, V.Y. and Willoughby, R.A.** (1979). Solutions of Ill-Posed Problems, *21*(2).

[33] **Louis, A.K.** (1992). Medical Imaging: State-of-the-Art and Future Development, *8*(5).

[34] **Craig, I.J.D. and Brown, J.C.** (1986). *Inverse Problems in Astronomy, A guide to inversion strategies for remotely sensed data*.

[35] **Bertero, M.**, **De Mol, C. and Pike, E.R.** (1985). Linear inverse problems with discrete data. I. General formulation and singular system analysis, *1*(4).

[36] **Charnes, A.**, **Frome, E.L. and Yu, P.L.** (1976). The equivalence of generalized least squares and maximum likelihood estimates in the exponential family., *71*(353).

[37] **Wallace, G.K.** (1991). The JPEG still picture compression standard, *34*(4).

[38] **Bahçeci, M.U.**, **Güngör, A. and Çetintepe, C.** (2022). Adaptive Transmit Beam Pattern Design for Compressed Sensing-based Direction of Arrival Estimation, *2022 IEEE International Symposium on Phased Array Systems Technology (PAST)*.

[39] **Krahmer, F.**, **Kruschel, C. and Sandbichler, M.** (2017). *Total Variation Minimization in Compressed Sensing*, `1704.02105`.

[40] **Dong, W.**, **Wang, P.**, **Yin, W.**, **Shi, G.**, **Wu, F. and Lu, X.** (2019). Denoising Prior Driven Deep Neural Network for Image Restoration, *41*(10).

[41] **Bassett, R. and Deride, J.** (2016). Maximum a Posteriori Estimators as a Limit of Bayes Estimators.

[42] **Boyd, S.**, **Parikh, N.**, **Chu, E.**, **Peleato, B. and Eckstein, J.** (2011). *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*.

[43] **Attouch, H.**, **Chbani, Z.**, **Fadili, J.M. and Riahi, H.** (2021). Fast Convergence of Dynamical ADMM via Time Scaling of Damped Inertial Dynamics.

[44] **Wang, Y.**, **Yin, W. and Zeng, J.** (2019). Global Convergence of ADMM in Nonconvex Nonsmooth Optimization, *78*(1).

[45] **Afonso, M.V.**, **Bioucas-Dias, J.M. and Figueiredo, M.A.T.** (2009). An Augmented Lagrangian Approach to the Constrained Optimization Formulation of Imaging Inverse Problems.

[46] **Zhang, K.**, **Li, Y.**, **Zuo, W.**, **Zhang, L.**, **Van Gool, L. and Timofte, R.** (2021). Plug-and-Play Image Restoration with Deep Denoiser Prior, (01).

[47] **Wei, K.**, **Aviles-Rivero, A.**, **Liang, J.**, **Fu, Y.**, **Schönlieb, C.B. and Huang, H.** (2020). *Tuning-free Plug-and-Play Proximal Algorithm for Inverse Imaging Problems*.

[48] **Yuan, X.**, **Liu, Y.**, **Suo, J. and Dai, Q.** (2020). *Plug-and-Play Algorithms for Large-scale Snapshot Compressive Imaging*.

[49] **Alver, M.B.**, **Saleem, A. and Çetin, M.** (2021). Plug-and-Play Synthetic Aperture Radar Image Formation Using Deep Priors, *IEEE Transactions on Computational Imaging*, *7*.

[50] **Eksioglu, E.M.** (2016). Decoupled Algorithm for MRI Reconstruction Using Nonlocal Block Matching Model: BM3D-MRI, *Journal of Mathematical Imaging and Vision*, *56*(3), 430–440.

[51] **Bahçeci, M.U.**, **Güngör, A. and Çukur, T.** (2022). *Sıkıştırılmış Algılayıcı Kamera için bir Kalibrasyon Yöntemi*, (Submitted, TR Patent).

[52] **Güngör, A.**, **Bahçeci, M.U.**, **Ergen, Y.**, **Yelboğa, T.**, **Ekiz, O.O.**, **Sözak, A. and Çukur, T.** (2023). An Online Calibration-based Reconstruction Technique for Single Pixel Imaging, (Manuscript in preparation).

[53] **MacAulay, C.E. and Dlugan, A.L.P.** (1998). Use of digital micromirror devices in quantitative microscopy, ***R.C. Leif***, ***D.L. Farkas***, ***R.C. Leif and B.J. Tromberg***, *editors, Optical Investigations of Cells In Vitro and In Vivo*, volume3260, International Society for Optics and Photonics, SPIE.

[54] **McCann, M.T.**, **Jin, K.H. and Unser, M.** (2017). Convolutional Neural Networks for Inverse Problems in Imaging: A Review, *34*(6).

[55] **Ioffe, S. and Szegedy, C.** (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*.

[56] **Dubey, S.R.**, **Singh, S.K. and Chaudhuri, B.B.** (2022). *Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark*.

[57] **Shi, W.**, **Caballero, J.**, **Huszár, F.**, **Totz, J.**, **Aitken, A.P.**, **Bishop, R.**, **Rueckert, D. and Wang, Z.** (2016). *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*.

[58] **Agustsson, E. and Timofte, R.** (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[59] **Shorten, C. and Khoshgoftaar, T.M.** (2019). A survey on Image Data Augmentation for Deep Learning, *6*(1).

[60] **Richardson, W.H.** (1972). Bayesian-Based Iterative Method of Image Restoration, *62*(1).

[61] **Lucy, L.B.** (1974). An iterative technique for the rectification of observed distributions, *79*.

[62] **Monga, V.**, **Li, Y. and Eldar, Y.C.** (2021). Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing, *38*(2).

[63] **Chen, Y. and Pock, T.** (2017). Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration, *39*(6).

[64] **Zhang, X.**, **Lu, Y.**, **Liu, J. and Dong, B.** (2018). *Dynamically Unfolding Recurrent Restorer: A Moving Endpoint Control Method for Image Restoration*.

[65] **Zheng, S.**, **Jayasumana, S.**, **Romera-Paredes, B.**, **Vineet, V.**, **Su, Z.**, **Du, D.**, **Huang, C. and Torr, P.H.S.** (2015). Conditional Random Fields as Recurrent Neural Networks, *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE.

[66] **Liu, Z.**, **Li, X.**, **Luo, P.**, **Loy, C.C. and Tang, X.** (2017). *Deep Learning Markov Random Field for Semantic Segmentation*.

[67] **Hershey, J.R.**, **Roux, J.L. and Weninger, F.** (2014). *Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures*.

[68] **Ronneberger, O.**, **Fischer, P. and Brox, T.** (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*.

[69] **Hawkins, D.M.** (2004). The problem of overfitting., *44*(1).

[70] **Gungor, A.**, **Kar, O.F. and Guven, H.E.** (2018). A Matrix-Free Reconstruction Method for Compressive Focal Plane Array Imaging, IEEE.

[71] **Li, C.** (2010). *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing*.

[72] **Kar, O.F.**, **Gungor, A. and Guven, H.E.** (2019). Learning Based Regularization for Spatial Multiplexing Cameras, *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp.1–5.

[73] **Tian, Y.**, **Fu, Y. and Zhang, J.** (2022). Plug-and-play algorithms for single-pixel imaging, *Optics and Lasers in Engineering*, *154*, 106970.

[74] **Liao, X.**, **Li, H. and Carin, L.** (2014). Generalized Alternating Projection for Weighted-$\ell_{2,1}$ Minimization with Applications to Model-Based Compressive Sensing, *7*(2).

[75] **Lau, S.L.H. and Chong, E.K.P.** (2022). *Single-Pixel Image Reconstruction Based on Block Compressive Sensing and Deep Learning*.

[76] **Xu, B.**, **Wang, N.**, **Chen, T. and Li, M.** (2015). *Empirical Evaluation of Rectified Activations in Convolutional Network*.

[77] **Gholamalinezhad, H. and Khosravi, H.** (2020). Pooling Methods in Deep Neural Networks, a Review, *ArXiv*, *abs/2009.07485*.

[78] **Yuan, X.** (2016). Generalized alternating projection based total variation minimization for compressive sensing, *2016 IEEE International Conference on Image Processing (ICIP)*, pp.2539–2543.

[79] **Bian, L.**, **Suo, J.**, **Dai, Q. and Chen, F.** (2018). Experimental comparison of single-pixel imaging algorithms., *35*(1).

[80] **Xie, W.S.**, **Yang, Y.F. and Zhou, B.** (2014). An ADMM algorithm for second-order TV-based MR image reconstruction, *67*(4).

[81] **Zhang, K.**, **Zuo, W.**, **Chen, Y.**, **Meng, D. and Zhang, L.** (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, *IEEE Transactions on Image Processing*, *26*(7), 3142–3155.

[82] *DnCNN*, `https://github.com/cszn/DnCNN`.

[83] *Pytorch-UNet*, `https://github.com/jvanvugt/pytorch-unet`.

[84] *Skimage Transform Resize*, `https://scikit-image.org/docs/dev/api/skimage.transform.html#skimage.transform.resize`.

[85] **Wang, Z.**, **Bovik, A.**, **Sheikh, H. and Simoncelli, E.** (2004). Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing*, *13*(4), 600–612.

[86] **Zhang, R.**, **Isola, P.**, **Efros, A.A.**, **Shechtman, E. and Wang, O.** (2018). *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*.

CURRICULUM VITAE

Name Surname: Muhammet Umut Bahçeci

EDUCATION:

- **B.Sc.:** 2018, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Electronics and Communication Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2018-... Research Engineer, ASELSAN Research Center, ASELSAN.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Bahçeci M. U.**, Ekşioğlu E.M. (2023). Two Plug-and-Play Priors-based Compressive Sensing Reconstruction for Single-pixel Imaging with Few Snapshots, *2nd International Graduate Research Symposium (IGRS'23),* (Accepted for publication). (Article Instance)

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Bahçeci M. U.**, Kalfa M. (2020). Implementation and Control of Fundamental Signal Processing Blocks For Radar In FPGA, *2020 28th Signal Processing and Communications Applications Conference (SIU),* pp. 1-4, doi: 10.1109/SIU49456.2020.9302331. (Article Instance)

- **Bahçeci M. U.**, Güngör A., Çetintepe Ç., Tuncer T. E. (2021). A Comparison of Direction of Arrival Estimation Performance for Phased-MIMO Systems, *2021 29th Signal Processing and Communications Applications Conference (SIU),* pp. 1-4, doi: 10.1109/SIU53274.2021.9477803. (Article Instance)

- Fişne A., **Bahçeci M. U.**, Dursun M., Aydoğmuş S., Çetintepe Ç. (2022). High-Level Synthesis based Radar Hardware Implementation and Performance Comparison, *2022 Innovations in Intelligent Systems and Applications Conference (ASYU),* pp. 1-6, doi: 10.1109/ASYU56188.2022.9925373. (Article Instance)

- **Bahçeci M. U.**, Güngör A., Çetintepe Ç. (2022). Adaptive Transmit Beam Pattern Design for Compressed Sensing-based Direction of Arrival Estimation, *2022 IEEE International Symposium on Phased Array Systems Technology (PAST),* pp. 01-08, doi: 10.1109/PAST49659.2022.9974981. (Article Instance)

- **Bahçeci M. U.**, Kılıç B. (2023). Transmitter and Receiver Design in Compressive Sensing based Direction of Arrival Estimation, *2023 17th European Conference on Antennas and Propagation (EuCAP),* pp. 1-4, doi: 10.23919/EuCAP57121.2023.10133445. (Article Instance)