ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MISSING DATA RECOVERY IN GPR WITH DEEP LEARNING

M.Sc. THESIS

Kübra TAŞ

Department of Electronics and Communication Engineering

Telecommunications Engineering Programme

JUNE 2021

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MISSING DATA RECOVERY IN GPR WITH DEEP LEARNING

M.Sc. THESIS

Kübra TAŞ
(0504161351)

Department of Electronics and Communication Engineering

Telecommunications Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Işın ERER

JUNE 2021

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DERİN ÖĞRENME İLE GPR GÖRÜNTÜLERİNDE VERİ KURTARMA

YÜKSEK LİSANS TEZİ

Kübra TAŞ
(0504161351)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Işın ERER

HAZİRAN 2021

Kübra TAŞ, a M.Sc. student of ITU Graduate School student ID 0504161351 successfully defended the thesis entitled “MISSING DATA RECOVERY IN GPR WITH DEEP LEARNING”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Işın ERER** Istanbul Technical University	....
Jury Members :	**Prof. Dr. Nebiye Musaoğlu**	....
	Assoc. Prof. Dr. Nur Hüseyin Kaplan	....
		....

Date of Submission : **11 June 2021**
Date of Defense : **14 July 2021**

FOREWORD

I would like to present my sincere thanks to my Advisor Assoc.Prof.Dr. Işın Erer, for her supervision and support throughout this unforgettable experience for me.

I would like to share my deepest appreciation with the person, who did and will survive me through many adventures in my life, my fiance and partner in crime Ufuk Altun. You make my dreams come through. And I would like to thank the muse of my life, my sister İlayda and my twin soul Derya for giving me an endless energy and motivation. Also my deepest gratitude to my beautiful and caring mother Serap and my beloved father Muharrem for their unconditional love and support. You and my beautiful baby girl Biber are the center of my universe, the reason for my existence and laughter.

JUNE 2021

Kübra TAŞ
Electronics and Communication Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

SN-GAN	: Spectrally Normalized Generative Adversarial Networks
GAN	: Generative Adversarial Networks
GoDec	: Go Decomposition
GI	: Generative Inpainting
GPR	: Ground Penetrating Radar
IoU	: Intersection of Union
LmaFit	: Low-rank Matrix Fitting
mAP	: Mean Average Percentage
NNM	: Nuclear Norm Minimization
NMC	: Non-negative Matrix Factorization
DIP	: Deep Image Prior
PEN-Net	: Pyramid-Context Encoder Network
PSNR	: Peak Signal-to-Noise Ratio
R-CNN	: Region Based Convolutional Neural Network
RoI	: Region of Interest
RPN	: Region Proposal Network
SSIM	: Structural Similarity Index
SU-Net	: Simplified U-NET
SVD	: Singular Value Decomposition
COCO	: Common Objects in Common

LIST OF TABLES

Page

LIST OF FIGURES

Page

MISSING DATA RECOVERY IN GPR WITH DEEP LEARNING

SUMMARY

Ground Penetrating Radar (GPR) is a measurement device used for detecting buried targets and its fundamental function relies on radar-based principle. Basically, it works by collecting a radar signal at the receiver that is sent from a transmitter. It also contains a control unit for recording the radar signal collected at the receiver. These signals, directed to a particular area or field, are used to detect objects buried under the surface and cannot be seen from the outside and convert this detection into visual interpretations. Considering all objects in the universe have a permeability level, GPR takes advantage of the fact that it causes a difference in measured permeability levels. If an object buried underground has a relative permeability concerning the environment in which the object is buried, the radar signals collected at the receiver side would present distinguishable patterns. This is because when the radar signals that cross the object and the ground where the object is located, would be collected back to the receiver would present patterns in parallel with the relative permittivity. In an instant that the GPR navigating through an area starts to send radar signals that will target the buried object on the area, hyperbola patterns are formed in the control unit of the GPR to illustrate this object. Additionally, the fact that it is independent of any material type, ensures that GPR has a great advantageous point in terms of applicability.

These advantages aside, many problems may be encountered during collecting radar data properly in the GPR method. These problems may originate from different sources, such as technical glitches that may occur during data acquisition, limited physical access to some regions, or low scanning resolution. In this case, A-scans are collected corrupted or incomplete and these A-scans become inapplicable for any post-processing on the obtained data, such as clutter removal or object detection. For this reason, in the literature, various methods are proposed as missing data recovery methods on GPR data. Many of these approaches treat the GPR data as a matrix and apply different methods to recover missing values on the matrices using existing values.

While these methods can be considered conventional methods, deep learning-based methods, which are adapted to many different applications with the increase in the processing capacity of computers, are gradually strengthening their place in the literature. This process is also known as image inpainting and is performed with many different deep learning architectures in the image processing domain. And image painting is the process of recovering an image with corrupted or missing parts, similar to the original. In many applications in the literature, image inpainting is developed using AutoEncoder or Generative Adversarial Network (GAN) architectures. Alternatively, the U-Net architecture, which was originally designed for object detection tasks, is successfully adapted to image inpainting.

In this thesis, different missing data recovery techniques have been applied on GPR data that are generated by simulation and GPR data that are taken from real life, that both conventional and deep learning-based missing data recovery methods are applied on artificially corrupted data.

In the first step, pixels or columns are removed randomly on both simulated and real GPR data. These data removals are performed at four different levels, 10%, 30%, 50%, and 80%, relative to the entire simulated data. In addition, columns of certain thickness are removed in blocks from the simulated data. For each of these scenarios, missing data recovery was performed using the GoDec, NNM, LmaFit, and NMC methods. The results of these approaches, which are referred to as matrix completion methods in the literature, are listed with different metrics. In addition to these methods, deep learning-based missing data recovery methods were also run on each scenario. Generative Inpainting, PEN-Net, and Deep Image Prior methods are applied as deep learning-based missing data recovery methods, and the results obtained with each method are listed comparatively.

On the real data, pixels and columns are removed randomly at two different levels, %30 and %80. On these artificially corrupted data, conventional missing data recovery methods, GoDec, NNM, LmaFit, and NMC, are applied. And similarly, as deep learning-based missing data recovery methods Generative Inpainting, PEN-Net and Deep Image Prior methods, Generative Inpainting, PEN-Net, and Deep Image Prior methods are applied.

In this thesis, relying on the obtained results, two different deep learning-based data recovery methods are proposed for data recovery on GPR data. It has been revealed that the semantic details in the image are successfully captured even at the limit values with the deep learning-based data recovery process. In this case, when compared to conventional methods, the advantage of deep learning-based data recovery methods emerges, especially in cases where there is more than 50% information loss. Additionally, with the data line created on the Deep Image Prior method, the success of this method on both simulated and real data is demonstrated with numerical and visual results.

In addition to these analyzes, object detection application is carried out in GPR data to validate the success of deep learning-based missing data recovery method, which is selected as Deep Image Prior. And the comparative results of both applications with appropriate metrics are presented. In addition, comparisons between the proposed method and the deep learning-based methods are presented.

DERİN ÖĞRENME İLE
GPR GÖRÜNTÜLERİNDE VERİ KURTARMA

ÖZET

Yere Nüfuz Eden Radar (YNR), gömülü hedeflerin algılanmasında kullanılan ve radara dayalı bir prensiple çalışan bir ölçüm yöntemidir. Radarın kullanılması sebebiyle ölçümün yapıldığı malzeme üzerinde herhangi bir tahribat yaratmaz ve bu sebeple uygulama alanları oldukça esneklik gösterir. Temelinde, vericiden gönderilen bir sinyalin alıcı yardımıyla toplanması yöntemiyle çalışır. Ayrıca alıcıda toplanan radar sinyalinin kaydedilmesi için bir kontrol ünitesi içerir. Belirli bir zemine ya da yapıya yönlendirilen bu sinyaller, yüzeyin altında kalan ve dışarıdan görülmesi mümkün olmayan nesnelerin tespitinin ve haritalamasının yapılması için kullanılır. Evrende yer alan tüm nesnelerin bir elektriksel geçirgenlik seviyeleri vardır ve YNR, farklı nesnelerin birbirinden farklı elektriksel geçirgenlik seviyelerinin olmasından yararlanır. Herhangi bir zeminin altında gömülü olan bir nesnenin, gömülü olduğu ortama göre göreceli bir elektriksel geçirgenlik değerine sahip olması durumunda, alıcı tarafında toplanan radar sinyalleri belirli bir örüntü oluşturacaktır. Bunun sebebi, nesneye ve nesnenin bulunduğu zemine çarpıp, alıcıya geri yansıyan radar sinyallerinin, göreceli elektriksel geçirgenliğe paralel olarak farklılık gösterecek olmasıdır. Bir zemini dolaşan YNR, zeminde bulunan gömülü nesneye denk gelecek radar sinyallerini göndermeye başladığı anda YNR'nin kontrol ünitesinde, bu nesneyi sembolize edecek şekilde hiperbol desenler oluşur. Ayrıca herhangi bir malzeme türüne bağlı olmaması, YNR'nin uygulanabilirlik anlamında oldukça avantajlı bir noktada konumlanmasını sağlar. Sonuç olarak bir nesnesin, gömülü olduğu alandan farklı bir elektriksel geçirgenliğe sahip olmasıyla durumunda, nesnenin tespiti mümkün hale gelebilir. Ayrıca bu ölçümler için nesne üzerinde herhangi bir tahribat yapılmasının gerekli olmayışı, YNR yöntemini alternatiflerine göre tercih edilmek konusunda avantaj sağlar.

Bu avantajlarının yanı sıra, YNR yönteminde radar verilerinin düzgün toplanılabilmesi konusunda sorunlar yaşanmaktadır. Bu sorunlar, yöntemin tercih edilmesi ve uygulanabilmesi konusunda dezavantajlar yaratmaktadır. Bu sorunlar, veri toplanması esnasında yaşanabilecek herhangi bir teknik aksaklık, bazı alanlara fiziksel erişimin kısıtlı olması ya da tarama çözünürlüğü olarak sıralanabilir.Bu sebeple literatürde eksik YNR verilerini kurtarmak için birçok farklı metot önerilmiştir. Bu metotların birçoğu, YNR verisini bir matris olarak ele alır ve matrisler üzerinde eksik olan değerlerin, var olan değerler kullanılarak elde edilmesine dayanırlar. Farklı bir deyişle matris tamamlama olarak literatürde yer edinmiş bu metot üzerine, yine literatürde yer edinmiş birçok matematiksel yaklaşım sunulmuştur.

Konvansiyonel yöntemler olarak değerlendirilebilecek bu yöntemlerin yanı sıra, günümüzde bilgisayarların işlem kapasitelerinin artması ile beraber birçok farklı uygulamaya adapte edilen derin öğrenme tabanlı yöntemler de gittikçe literatürde yerlerini sağlamlaştırmaktadır. Görüntü işleme alanında, iç boyama olarak da bilinen

ve derin öğrenmeye dayalı bir çok farklı mimari ile gerçekleştirilen bu işlem, bozulmuş ya da eksik kısımları olan bir görüntünün orjinaline benzerlik gösterecek şekilde onarılması işlemidir. Literatürde birçok uygulamada iç boyama işlemi AutoEncoder ya da Çekişmeli Üretken Ağ mimarileri kullanılarak geliştirilmiştir. Alternatif olarak orjinalinde nesne tanımlaması üzerine tasarlanan U-Net mimarisi de ayrıca iç boyama işlemine başarılı bir şekilde adapte edilebilmiştir.

Tezin ilk kısmında, literatürde önerilen konvansiyonel eksik veri kurtarma yöntemleri üzerine yapılan çalışmaların kapsamlı bir listesi sunulmuştur. Eksik verinin bir matris olarak ele alındığı ve buna yönelik matris tamamlama üzerine farklı algoritmalar sunan metotların büyük bir kısmı hakkında bilgi verilmiştir. Ardından kronolojik bir sıraya uygun olarak, eksik veri kurtarma üzerine derin öğrenme tabanlı olarak yapılan çalışmalar aktarılmıştır. Tezin ikinci ve üçüncü kısımlarında, hem konvansiyonel hem de derin öğrenme tabalı yöntemler, uygulandıkları algoritmalarla beraber özetlenmiştir. Tezin dördüncü kısmında yürütülen çalışmaların tamamı kapsamlı bir şekilde aktarılmıştır ve bulgular tezin son kısmında sunulmuştur.

Bu tezde, simülasyon ile oluşturulmuş YNR verileri ve gerçek hayattan alınmış farklı YNR verileri üzerine, farklı eksiltme teknikleri uygulanmıştır ve yapay olarak eksiltme uygulanan veriler üzerinde hem konvansiyonel hem de derin öğrenme tabanlı eksik veri kurtarma teknikleri denenmiştir.

İlk aşamada hem simüle hem de gerçek YNR verileri üzerinde piksel ve sütun bazlı rastsal eksiltmeler yapılmıştır. Bu eksiltmeler simüle verinin tamamına oranla birbirinden farklı dört seviyede, %10, %30, %50, ve %80, gerçekleştirilmiştir. Buna ek olarak, simüle veri üzerinde belirli kalınlıkta sütunlar blok halinde eksiltilmiştir. Bu senaryoların her biri için, GoDec, NNM, LmaFit ve NMC yöntemleri kullanılarak eksik veri kurtarma işlemi gerçekleştirilmiştir. Literatürde matris tamamlama yöntemleri olarak geçen bu yaklaşımların sonuçları farklı metriklerle beraber listelenmiştir. Bu yöntemlerin yanı sıra, derin öğrenme tabanlı eksik veri kurtarma yöntemleri de her bir senaryo üzerinde çalıştırılmıştır. Derin öğrenme tabanlı olarak Generative Inpainting, PEN-Net ve Deep Image Prior yöntemleri kullanılmıştır, ve her bir yöntem ile elde edilen sonuçlar da kıyaslamalı olarak listelenmiştir. Bu üç yöntem arasından PEN-Net ve Deep Image Prior yöntemleri başarılı sonuç vermiş, ancak Generative Inpainting yöntemi belirlenen bir eğitim aralığında yeterli bir sonuç üretememiştir.

Gerçek veri üzerinde ise iki farklı seviyede, %30 ve %80, piksel ve sütun bazlı rastsal eksiltmeler yapılmıştır. Bu eksik görüntüler üzerinde de konvansiyonel veri kurtarma yöntemlerinden GoDec, NNM, LmaFit ve NMC yöntemleri kullanılmıştır. Ve simüle veri üzerindeki başarısı göz önünde bulundurulduğunda, gerçek veri üzerinde derin öğrenme tabanlı veri kurtarma yöntemlerinden olan Deep Image Prior yönteminden faydalanılmıştır.

Bu analizlerin yanı sıra derin öğrenme tabanlı veri kurtarma yöntemleri ile tamamlanan YNR verilerinin, işlem başarısının test edilmesi için, YNR verilerinde nesne tespiti uygulaması yapılmıştır. Aynı uygulama, orjinal YNR verileri ile de gerçekleştirilmiş, ve iki uygulamanın da uygun metriklerle kıyaslamalı sonuçları sunulmuştur. Ayrıca literatürde derin öğrenme tabanlı YNR verilerine yönelik önerilen veri kurtarma işlemleri üzerine kıyaslamalar sunulmuştur.

Bu tezde, elde edilen sonuçlarla beraber, YNR verileri üzerinde veri kurtarma işlemine uygun iki farklı derin öğrenme tabanlı yöntemler olan Generative Inpainting ve PEN-Net yöntemleri önerilmiştir. Derin öğrenme tabanlı veri kurtarma işlemi ile görüntüde anlamsal detayların sınır değerlerde bile başarılı bir şekilde yakalandığı ortaya koyulmuştur. Bu durumda konvansiyonel yöntemlerle kıyaslandığında, özellikle %50'nin üzerinde veri kaybının olduğu durumlarda derin öğrenme tabanlı veri kurtarma yöntemlerinin farkı ortaya çıkmaktadır. Ayrıca bir derin öğrenme tabanlı veri kurtarma yöntemi olan Deep Image Prior yöntemi üzerine oluşturulan veri hattı ile, hem simüle hem de gerçek veriler üzerinde bu yöntemin başarısı nümerik ve görsel sonuçlarla beraber gösterilmiştir.

Toplamda simüle veri üzerinde 4 farklı seviyede hem piksel hem de sütun bazlı eksiltme senaryoları değerendirilmiştir. Buna ek olarak simüle veri üzerinde blok halinde sütun kalınlığında veri parçası silinmiş ve kalan veri üzerinde eksik veri kurtarma işlemi yapılmıştır. Tezde önerilen derin öğrenme tabanlı veri kurtarma işleminin başarısının doğrulanması amacıyla simüle veri üzerinde nesne tanıımlama ileri işlemi uygulanmıştır. Bu uygulamada herhangi bir işlemin uygulanmadığı simüle veri ile kıyaslanmıştır.

Ayrıca simüle YNR verileri üzerinde yapılan tahliller sonucu başarısı doğrulanan derin öğrenme tabanlı eksik veri kurtarma yöntemi, gerçek YNR verileri üzerinde de değerlendirilmiştir. Bu değerlendirme konvansiyonel eksik veri kurtarma yöntemleriyle de kıyaslamalı olarak verilmiştir.

1. INTRODUCTION

Ground-penetrating radar is a geophysical technique, with the objective to measure field amplitude versus time excitation. It is a nondestructive measurement method and allows the detection of objects lying beneath the ground or buildings. A wide range of items can be detected with GPR signals. A GPR system consists of two antennas (a transmitter and a receiver) and an environment for data storage. A GPR controller radiates electromagnetic pulses through the transmitter antenna. Afterward, reflected waves from buried material or surfaces are collected at the receiver antenna. GPR can detect any material, based on the material's electrical or magnetic responses. As long as the magnetic response of a material and its surroundings has a relative permittivity, the collected data contains enough information to distinguish the material from its surroundings. A critical advantage of GPR is that its detection capabilities are not limited to a set of material types, e.g. it can also detect non-metal materials. In fact, any material which has a different permittivity than its surrounding environment has the potential to be detected with GPR.

Aside from its advantages, GPR-based detection also presents various challenges. Especially, GPR data collection is highly vulnerable to situations such as difficult access to some regions, extremely damaged A-scans, data with clutter that makes objects less visible, failures during data acquisition. As a result of these situations, partial information loss is a very common problem in capturing GPR measurements. Moreover, missing data leads to poor performance of clutter removal, and poor interpretation, identification, and analysis of GPR data. In order to overcome these problems, several missing data recovery methods are introduced in the literature. In general, these methods are subject to linear interpolation or matrix completion [3].

In the literature, various techniques are developed for matrix completion, such as convex relaxation-based algorithm [4], gradient-based algorithm [5], and alternating minimization-based algorithm [6]. In matrix completion, nuclear norm minimization (NNM) plays an important role in aspects of computational complexity. The nuclear norm of a matrix is an approximation of the matrix rank by shrinking all singular values equally. For matrix completion via convex optimization, singular value thresholding (SVT) is proposed in [4], which aims to minimize the nuclear norm of a matrix. In [7], the authors focus their attention to minimizing the rank of the matrix, instead of solely minimizing the nuclear norm with convex relaxation. For this purpose, the authors of [7] propose an accelerated proximal gradient algorithm .In [8], the augmented Lagrangian method (ALM) and the alternating direction method (ADM) are used for nuclear norm minimization. In [9], in order to improve the computational performance of matrix completion, the authors propose fixed point and Bergman iterative algorithms to solve the nuclear norm minimization problem. In [9] a fast and robust algorithm (Fixed Point Continuation with Approximate SVD (FPCA)) is defined to solve the rank minimization problem of very large matrices. In [10], a solution for the non-negative matrix factorization-and-completion problem is introduced by using an algorithm based on the classic alternating direction augmented Lagrangian method.

One of the main drawbacks of the above-mentioned methods is the high computational complexity since they include SVD in each iteration. In order to accelerate the process and decrease the complexity, Go Decomposition (GoDec) is proposed in [11]. GoDec is able to efficiently process low-rank matrix representations and accelerate the calculations by using bilateral random projections (BRP). Low-rank matrix optimization is also considered in [12]. In this study, an algorithm that relies on a nonlinear successive over-relaxation (SOR) is presented to solve a linear least-squares problem per iteration.

As an alternative to the nuclear norm minimization, the authors of [13] preferred to concentrate on non-convex rank relaxation and proposed a top-N recommender system (MClodget) for matrix completion. Since existing methods are not designed for non-linear structures, in [14] non-linear structures are considered. For this purpose, deep matrix factorization is proposed in [14], by relying the model on a nonlinear latent variable model.

It should be noted that the methods that are mentioned so far approach the missing data recovery as a matrix completion problem. In addition to these methods, also patch-based missing data recovery methods are proposed in the literature [15–18]. Early studies in the literature with patch-based method principally focused on texture synthesis [15, 19]. In [20], the authors applied texture synthesis for padding missing regions on an image. In most applications, data recovery in the missing region is handled by sampling data from surrounding regions. The metric that is used to perform this filling operation is based on distance information between data points, such as Euclidean or Scale-Invariant-Feature-Transform (SIFT) distance. In [21], the filling operation is handled by completing lines coming through to the missing region. Also in [16,22,23] alternative approaches are proposed in order to improve the performance of filling operation by presenting a better filling order. Patch-Match is proposed to find similar patches more abruptly [17]. In general, patch-based methods can be considered successful for inpainting images with similar contexts. However, since this approach can not capture image semantics, it performs poorly on images with complex patterns.

Among these matrix completion methods, APG [7], LmaFit [12], Optspace [24] and NNM [25] are applied to seismic data reconstruction problem. Wang et al. [26] compared the different versions of NNM and LmaFit algorithms and NNM outperformed the LmaFit method on both synthetic and real seismic data. Yang et al. [27] compared LmaFit with the APG method in their study and LmaFit showed superior performance wconcerning the APG method when the original signal has a complicated structure. For other cases, they proposed that APG can be a better choice. Xue [28] compared the LmaFit and Optspace algorithms in 3D seismic data reconstruction and LmaFit outperform the Optspace method in his/her study for both simulated and real datasets. However, the author suggested that for some extreme cases Optspace catches up and sometimes can exceed the performance of LmFit.

The methods for matrix completion summarized above performed well in many image processing, remote sensing, and seismic data reconstruction problems and the best ones are selected in this study for the cases of GPR data reconstruction for a different level of missing case on both simulated, and real datasets. Most of the popular matrix completion methods above have not been applied to the GPR data and proposed ones presented limited performance.Since the composition of GPR data as a data matrix is different from the other matrix completion problems where the target signature is very weak and it can be mixed with the missing pixels during the matrix completion process. Thus, every matrix completion algorithm can not be successful for GPR data with missing pixels due to the particular problem mentioned [29]. In [29], it has been showen that NMC and LmaFit methods have the superior achievement for missing data recovery on GPR images.

In recent years, deep learning-based methods are also introduced to the literature for missing data recovery [19, 30–34]. Deep-learning-based methods are commonly set up with two sub-networks: Completion and auxiliary discriminative networks. The completion network focuses on image semantics and constructs the missing points in the image by using semantics. The discriminative network performs improvements over the image to propose conceivable images by using generative adversarial training [35]. In the literature, these sub-networks are regarded as content generation and texture refinement, respectively. Texture refinement can be explained as matching the features of the simulated neural patches with patches that are collected from a known spot of the image. With this combination, contextual information and high-frequency details of the image can be captured [36].

The application of convolutional neural networks with context encoders was a striking breakthrough for missing data recovery problems. One of the first examples of the usage of the context encoders is given in [36] where the authors use a context encoder to perform missing data recovery by manipulating information on the surrounding area. In order to obtain satisfying image quality on generated images, loss information is utilized. In [37], the authors used transformation invariant image feature loss to provide more conceivable images, while in [38] feature-oriented inpainting framework is proposed by using loss information. In [31], a global adversarial loss is used to preserve the texture consistency between the original image and synthesized image. Also in [39], the authors exploited semantic parsing loss in order to optimize facial structures of face images.

In images with more complex patterns, synthesized images become more obscure. Because of this reason, after the generation of the content, the texture optimization step is added to the complete procedure. Finally, with the extension of the procedure into a two-step process, [30], a neural patch synthesis application is proposed, and the content and texture of the image are optimized collectively. However, texture optimization includes iterative steps and this creates computationally expensive calculations. In order to optimize process time for this step, a trainable framework is proposed to optimize textures and inpainting application is facilitated into a straight-forward pipeline [40,41]. As an alternative in [42], a special shift-connection layer is proposed by passing into the U-Net architecture to perform inpainting of regions with detailed textures.

On the other hand in [32], to train a network per image to generate image priors is proposed. The method is named as Deep Image Prior and an image prior is an underlying assumption of what we expect an image to be. For a natural image, prior is the assumption that the image to be noise and hole-free. In this paper, the authors argue against the approach that supervised learning is necessary for building proper image priors. As stated in [32], the proposed model is a kind of unsupervised learning approach that outperforms supervised alternatives (such as convolutional sparse coding (CSC) [43]). Deep image prior can be implemented with different neural network architectures, such as SkipNet [44], Residual Neural Networks (ResNet) [45] or with U-Net (with or without skip-connections) [46]. Eventually it can be said that this model does not learn from the data, and can be considered as a transition from optimization-based methods to deep learning-based methods.

In [32], authors suggest that using only a generator part of a network, low-level features can be captured for learning image semantics and a randomly initialized neural network can be used for inpainting. In [32], authors try to build a connection between two popular methods by constructing a new prior using a convolutional neural network with randomized weights. Deep image prior is used in medicine for denoising dynamic PET images [47] since it can perform denoising with only the target image and no dataset is required. Thus it can be said that the deep image prior technique is suitable for the cases with learning objectives in small datasets.

In this thesis, deep learning based missing data recovery methods are exploited for GPR images. Various missing data scenarios are examined for increasing levels of corruption on the images. Extensive simulation results are compared with existing methods in the literature. Performances of conventional methods and different deep learning-based methods are presented. Additionally, since in missing data recovery, enhancement in the image quality is aimed for post-processes, detection performances of both original and recovered images are evaluated.

1.1 Contributions

Contributions of this thesis can be listed in 2-folds. In the first fold, an outline is given regarding the missing data recovery methods. Also, four different state-of-the-art methods are explained in details. Both numerical and visual analysis on GPR data of these method are presented.

In the second fold, three individual deep learning-based missing data recovery methods are presented. And three pipelines corresponding to each method are designed. By using these pipelines, missing data recovery on GPR data is performed. Similar to the first fold, numerical and visual results that are obtained with these methods are given.

1.2 Outline of the Thesis

In Chapter 2 we present an introductory information about GPR and its scan types. For addressing the common problem of missing data acquisition on GPR data, we explain the conventional missing data recovery methods on GPR data. In this chapter, we also present four different and state-of-the-art missing data recovery methods, which rely on matrix completion. Chapter 3 is dedicated to explain three state-of-the-art and deep learning-based missing data recovery methods. In Chapter 4 we give information about the datasets that are used, our testing methods and implementation details. Also we present results that we obtained by using the techniques presented in Chapter 2 and Chapter 3. Lastly, the thesis is concluded in Chapter 5.

2. GROUND PENETRATING RADAR (GPR)

In this section, we present introductory information about GPR and its scan types. These scan types are listed as A-Scan, B-Scan, and C-Scan.

As next, for addressing the common problem of missing data acquisition on GPR data, we explain the conventional missing data recovery methods on GPR data. We also present four different and state-of-the-art missing data recovery methods, which rely on matrix completion.

2.1 Introduction

Ground-penetrating radar is a technique used for detecting objects under the ground or inside the buildings. It has a wide remit, such as for detecting explosive material underground, scanning bridges, buildings, airports for an internal fraction, or detecting it the positioning of the pipes and cables. Generally, GPR has similarities with conventional radar applications. However, instead of measuring radio waves or frequency with a probe through the air, in GPR revealing any hidden object through the ground or a building is performed. The common point is both methods use radar waves.

In a GPR following components are included,

- A control system
- One transmitter antenna
- One receiver antenna
- A data storage unit

Usually, a GPR system is coupled with a monitoring device for real-time analysis, as illustrated in Fig. 2.1. The controller creates electromagnetic pulses via the transmitter antenna. These pulses are sent to the medium, through the hidden object. And

Figure 2.1 : A GPR device constructed with (1) controller, (2) transmitter antenna, (3) receiver antenna, (4) data storage unit [1]

when electromagnetic pulses reach the object, a part of these pulses are reflected. The receiver antenna collects these reflected electromagnetic waves and forwards information that carried these waves to the data storage unit. However, it is worth mentioning that this information does not contain the image of the hidden object. Also, the medium that the object is located is mostly ground, either open soil, asphalt, or concrete. The ground term in the GPR is used as an identifier of one of these mediums.

The principle of GPR relies on the difference between the transmitted waves and the received waves. It is known that every material in the universe has a unique permittivity. And GPR technique benefits from the relative permittivity between different objects. When an electromagnetic wave is transmitted to the object, a different response from the object and the medium, that the object is placed, will be reflected. Assuming that both the object and the environment would have a different permittivity, the difference in the reflection can be measured and used as an identifier for the object. This fact makes GPR method advantageous since the method is not bounded by a range of material types. However, this conclusion fails when the buried object and the medium have similar permittivity. Then, since the reflected waves would not carry a difference, the object could not be identified.

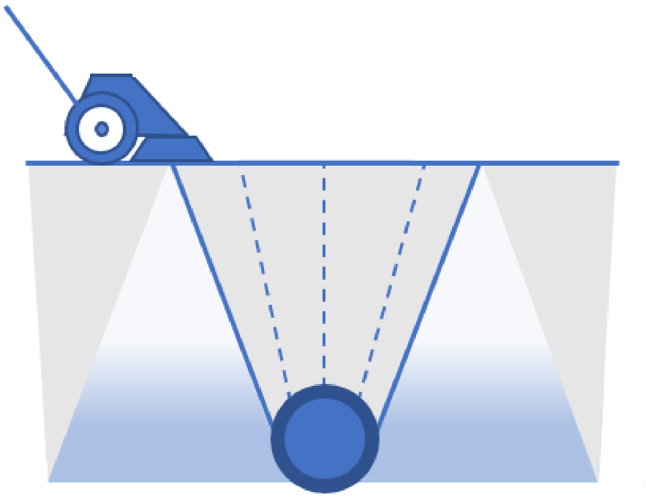


Figure 2.2 : Diagram of GPR electromagnetic waves encountering an object

Interpretation of GPR data is visualized as a two-dimensional representation of the vertical view of what lies in the ground. However, this interpretation does not have any resemblance with the actual image of the underground. In GPR data, signals are created by the boundary between two or more objects, and signals are reflected when the measured relative permittivity level is changed. And the change in the signal amplitude gives information about the degree of the change in permittivity levels. These changes create hyperbolic shapes.

When a GPR system goes through a surface, at the first step, the edge of the object will be reflected. As illustrated in Fig. 2.2, this distance does not signify the actual depth that the object buried. The measured depth shortens until the point where the GPR system is positioned right above the object. After this point, the measured depth will start to increase again, until the GPR system reaches the endpoint of the object. The processing step of the GPR data is conducted by first recording the data in a proper dimension to a data storage unit and then applying proper algorithms.

This principle of the GPR system is depicted in Fig. 2.3. When the object is detected for the first time, the starting point of the hyperbola is created. The obtained information is stored while the GPR system continues its movement and this information is used for shaping the hyperbola. When the scanning is completed, the point that gives the shortest distance to the surface corresponds to the tip of the hyperbola.

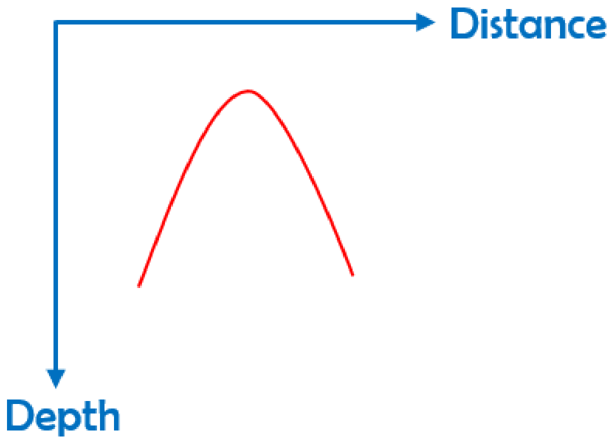


Figure 2.3 : Hyperbola belonging to the encountered object

2.2 Scan Types

When these scans are plotted in Cartesian's coordinate system, x and y axes represent the medium, while z axis represents the depth. Data collected with a GPR system can be represented as follows

$$f(x,y,z) = A(x_i, y_j, z_k) \tag{2.1}$$

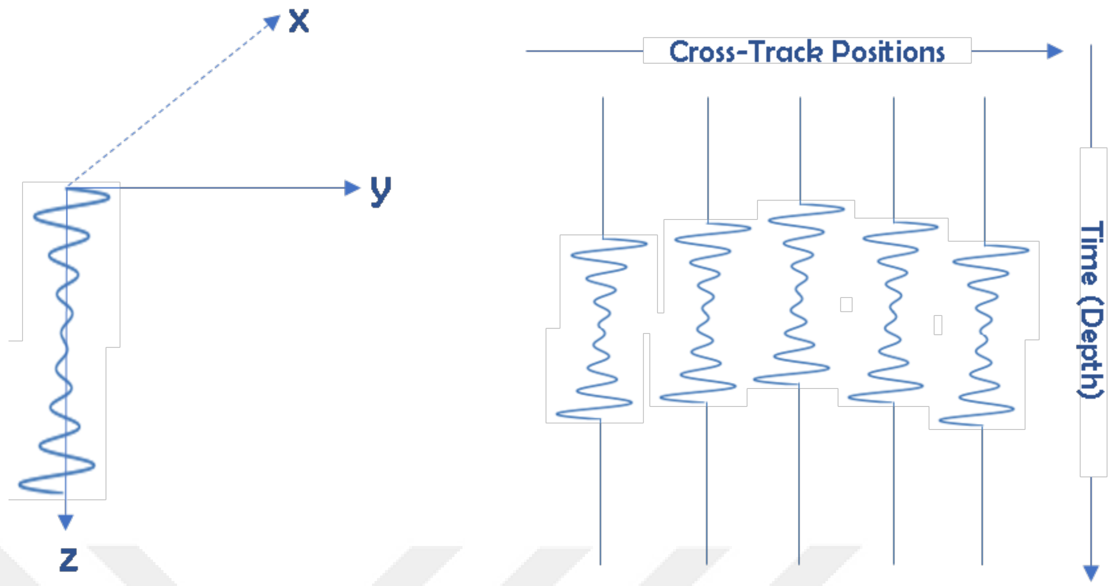
where i, j and k are consisted of $i = 1,...,P$, $j = 1,...,M$ and $k = 1,...,N$, where P, M and N represents three dimensions. In A-scan processing, the received data is considered as the convolution of different number of time functions. Each one of these time functions express the responses collected from the electromagnetic wave data.

2.2.1 A-scan

An A-scan is the collection of the data through the time and can be represented as follows

$$f(z) = A(x_i, y_j, z_k) \tag{2.2}$$

where i and j are kept constant, while k is consisted of $k = 1,...,N$. A-scan data is used to identify the type of a buried object, with a known location. A-scan is obtained by measuring a one-dimensional signal by keeping x and y location information constant. As illustrated in Fig. 2.4.a, z axis represents the depth while the measurement location are represented with the y axis.



(a) (b)

Figure 2.4 : (a) Raw A-Scan Data (b) Concatenation of multiple A-Scans.

2.2.2 B-scan

B-scan data is used to detect buried objects with unknown location information. During the movement of the transmitter, both transmitter and receiver signals are collected and stack together, at each point. And B-scan is generated via stacking these collected A-scan signals. To summarize, it can be said that B-scans are two-dimensional data, that are consisted of A-scans belonging to the along-track position of the GPR system, as illustrated in Fig. 2.5.

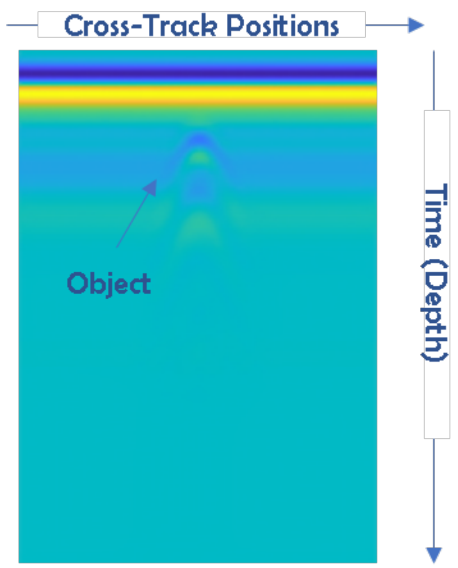


Figure 2.5 : Raw B-scan data

A sample B-scan data is given in Fig. 2.5, with the hyperbole-shaped reflection, caused by a buried object. These hyperbola shapes are widely used for identifying buried objects. Also, B-scan data can be represented as follows:

$$f(z) = A(x_i, y_j, z_k) \tag{2.3}$$

where j kept constant and i and k are consisted of $j = 1, \ldots, M$ and $k = 1, \ldots, N$, respectively. In this thesis, B-scan GPR data is used.

2.2.3 C-scan

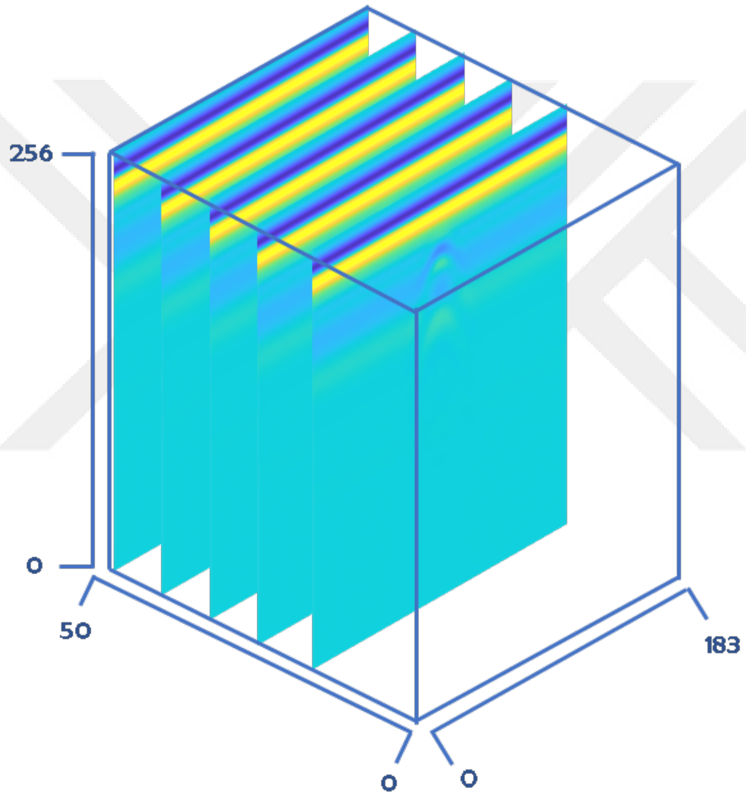


Figure 2.6 : Raw C-scan data

As illustrated in Fig. 2.6, C-scans are generated by stacking B-scan data that are collected any time instantly. These scans are illustrated in three dimensions and the x, y, and z axis represent horizontal location, vertical location, and depth, respectively. Also, C-scan data can be formulated as follows

$$f(z) = A(x_i, y_j, z_k) \tag{2.4}$$

where i, j and k are consisted of $i = 1, \ldots, P$, $j = 1, \ldots, M$ and $k = 1, \ldots, N$.

2.3 Missing Data Recovery on GPR Images

Missing data phenomenon frequently occurs in GPR measurements. It may be caused by many different reasons such as limited scanning resolutions, failures during the acquisition process, instrumental problems, the lack of accessibility to some areas under test, and severely corrupted A-scans. Since both B-scans and C-scans are constructed by assembling A-scans, any data loss affects complete processes that are conducted with GPR data. As a result, recovery of the missing data is essential to perform processes such as clutter removal, target detection, and identification [27].

Missing data problems occur often in GPR images due to constraints such as scanning resolutions, downfalls during obtaining data, and the existence of interrupting objects [29]. The general implementation of the missing data recovery methods can be briefly described for the GPR problem as follows.

Let $\mathbf{X} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ be the original GPR data matrix and $\mathbf{M} \in \mathbb{R}^{m\times n}$ be the GPR data matrix with missing information where m and n denote the dimensions, namely depth and antenna locations. We should note that , $\mathbf{M}$ can be directly used in the matrix completion methods for the pixel-wise case. However, the pre-transformation is required as given in [29] for the column-wise case to recover the input GPR data matrix with missing information since at least one observation for each row and column is necessary according to matrix completion theory [4]. More details about pre-transformation can be found in [29]. After the described pre-processing step, recovery of the missing samples in the GPR data matrix can be formulated as

$$\left[\mathbf{P}_{\boldsymbol{\Omega}}(\mathbf{M})\right]_{i,j} = \begin{cases} \mathbf{M}_{i,j}(i,j) \in \boldsymbol{\Omega} \\ 0, \quad \text{otherwise} \end{cases} \tag{2.5}$$

If the GPR data matrix $\mathbf{M}$ is low rank and its singular vectors are spread enough, the missing entries can be recovered by solving the following optimization problem

$$\min_{\mathbf{Z}} \ \text{rank}(\mathbf{Z}) \quad \text{s.t.} \quad \|\mathbf{P}_{\boldsymbol{\Omega}}(\mathbf{Z}-\mathbf{M})\|_{\mathrm{F}} \leq \delta \tag{2.6}$$

The unknown variable matrix is $\mathbf{Z}$, $\|.\|_{\mathrm{F}}$ denotes the Frobenius norm of the GPR data matrix, δ is a tolerance parameter that limits the error. Since the optimization problem in (2.6) is NP-hard, it should be relaxed to a convex optimization problem as

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_{*} \quad \text{s.t.} \quad \|\mathbf{P}_{\boldsymbol{\Omega}}(\mathbf{Z}-\mathbf{M})\|_{\mathrm{F}} \leq \delta \tag{2.7}$$

where $\|\cdot\|_*$ denotes the nuclear norm. (2.7) can be solved by matrix completion methods [11, 14].

Several techniques are proposed in the literature to perform data loss recovery. For instance, matrix completion methods are used to fill the missing regions on GPR images as data recovery. Due to the high computational complexity of alternative approaches, in this thesis following approaches are used as conventional missing data recovery for benchmark; nuclear norm minimization (NNM) [25], low-rank matrix fitting (LmaFit) [12], Non-negative Matrix Factorization (NMC) [10] and Go Decomposition (GoDec) [11].

2.3.1 Nuclear Norm Minimization

In the NNM method, the low-rank approximation is the backbone of the proposed algorithm in [25]. In this paper, the authors aim to maximize the likelihood of the missing entries in the input matrix, independent from the mean square error minimization.

In this approach, assuming that the matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ be the original GPR data and we search for the matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ such that the difference between the entries are minimized, subject to a given rank and this approach can be written as follows

$$\min \sum_{(i,j)\in\Omega} |\mathbf{X}_{ij} - \mathbf{M}_{ij}|^2 = \|\mathbf{PX} - \mathbf{PM}_F^2\| \quad \text{s.t.} \quad rank(\mathbf{X}) = k. \tag{2.8}$$

A problem such as it requires a special solution with the constraints in (2.8) is known as Eckart-Young Theorem [48]. The solution to this problem is formulated by singular value decomposition (SVD) and as given in [25], the authors propose a solution that relies on maximizing the likelihood over the missing entries, where the algorithm can be written as follows

$$\mathbf{X}_n = D_k((I - \mathbf{PX}_{n-1} + \mathbf{PM})) \tag{2.9}$$

where $D_k\mathbf{X}$ represents the best rank k approximation in the Frobenius norm.

2.3.2 Low-rank Matrix Factorization

Similarly, assume that $\mathbf{M} \in \mathbb{R}^{m \times n}$ is the GPR data matrix with missing information. LmaFit is an approach for resolving a low-rank matrix optimization problem, such as matrix completion, sparse matrix separation, and matrix compressive sensing. Specifically, the matrix completion problem can be formulated as

$$\min_{\mathbf{M}\in\mathbb{R}^{m\times n}} \operatorname{rank}(\mathbf{M}) \quad \text{s.t.} \quad \mathbf{M}_{ij} = A_{ij}, \quad \forall (i,j) \in \varepsilon, \tag{2.10}$$

where $\varepsilon \subset \{(i,j) : 1 \leq i \leq m, 1 \leq j \leq n\}$, $\mu > 0$, $vec(Z)$ is the vectorization of the matrix Z. Here, alternative to replace the rank problem with the nuclear norm minimization, a different approach is followed. First expressing the set of rank k matrices as follows

$$\mathbb{R}^{m\times n}(k) := \{\mathbf{M} \in \mathbb{R}^{m\times n} : \operatorname{rank}(\mathbf{M}) = k\}. \tag{2.11}$$

Here, proposed approach with LmaFit can be formulated as follows

$$\min_{Z} \|\mathbf{M} - \mathbf{Z}\|_F^2 \quad \text{s.t.} \quad Z_{ij} = A_{ij}, \quad \forall (i,j) \in \varepsilon \tag{2.12}$$

2.3.3 Non-negative Matrix Completion

In the NMC method, the authors aim to find non-negative low-rank matrices X and Y, such that **M** matrix can be represented as the product of these two low-rank matrices, XY [**?**]. In order to find these non-negative low-rank matrices, the authors benefit from the augmented Lagrangian method.

In order to ease NMC algorithm, the problem can be summarized as follows

$$\min_{Z} \{ \|\mathbf{P}_\Omega(\mathbf{Z} - \mathbf{M})\|_F^2 \}. \tag{2.13}$$

Here (2.13) can also be interpreted as follows

$$\begin{aligned} &\min_{(U,V,X,Y,X)} \quad \frac{1}{2}\|XY - Z\|_F^2 \\ &\text{s.t.} \quad X = U, \quad Y = V, \quad U \geq 0, \quad V \geq 0, \quad \mathbf{P}_\Omega(\mathbf{Z} - \mathbf{M}) = 0. \end{aligned} \tag{2.14}$$

If we express the augmented Lagrangian of (2.14), it can be written as follows

$$\begin{aligned} \mathscr{L}_A(X,Y,Z,U,V,\Lambda,\Pi) \quad &= \quad \frac{1}{2}\|XY - Z\|_F^2 + \Lambda \bullet (X - U) \\ &\quad + \Pi \bullet (Y - V) + \frac{\alpha}{2}\|X - U\|_F^2 + \frac{\beta}{2}\|Y - V\|_F^2, \end{aligned} \tag{2.15}$$

where

$$A \bullet B := \sum_{i,j} a_{ij} b_{ij}. \tag{2.16}$$

In (2.15) $\Pi \in \mathbb{R}^{q\times n}$ and $\Lambda \in \mathbb{R}^{m\times q}$ denote Lagrangian multipliers. $\alpha > 0$ and $\beta > 0$ are used as penalty parameters. NMC solution to the problem, defined in (2.13), can be calculated by taking the derivative of (2.15) with respect to each variable in the equation, one by one, as follows

$$\begin{aligned}
X_{k+1} =& argmin \mathscr{L}_A(X, Y_k, Z_k, U_k, V_k, \Lambda_k, \Pi_k),\\
Y_{k+1} =& argmin \mathscr{L}_A(X_{k+1}, Y, Z_k, U_k, V_k, \Lambda_k, \Pi_k),\\
Z_{k+1} =& argmin \mathscr{L}_A(X_{k+1}, Y_{k+1}, Z, U_k, V_k, \Lambda_k, \Pi_k)\\
U_{k+1} =& argmin \mathscr{L}_A(X_{k+1}, Y_{k+1}, Z_{k+1}, U, V_k, \Lambda_k, \Pi_k),\\
V_{k+1} =& argmin \mathscr{L}_A(X_{k+1}, Y_{k+1}, Z_{k+1}, U_{k+1}, V, \Lambda_k, \Pi_k).
\end{aligned} \tag{2.17}$$

Convergence of (2.17) and corresponding proof is given in [49].

2.3.4 Go Decomposition

In GoDec, the authors in [11] proposes to decompose matrix $\mathbf{M}$ with its corresponding sparse part S, low-rank part L and the noise G as $\mathbf{M} = L + S + G$. Again, let assume that $\mathbf{X} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ is the original GPR data matrix and $\mathbf{M} \in \mathbb{R}^{m\times n}$ is the GPR data matrix with missing information The problem to this approach can be stated as follows

$$\min_{L,S} ||\mathbf{M} - L - S||_F^2 \quad \text{s.t.} \quad rank(L) \leq r, card(s) \leq k. \tag{2.18}$$

As described in [11], naïve GoDec is a solution defined for the problem stated in (2.18) and is formulated as follows

$$\begin{cases} L_t = & arg \min_{rank(L)\leq r} ||\mathbf{M} - L - S_{t-1}||_F^2; \\ S_t = & arg \min_{card(S)\leq k} ||\mathbf{M} - L_t - S||_F^2. \end{cases}$$

As stated in the article, solving both of the equations given in (2.18) can be done by updating the L_t and S_t with singular value hard thresholding $X - S_{t-1}$ and entry-wise thresholding $X - L_t$, respectively.

3. MISSING DATA RECOVERY WITH DEEP LEARNING

In this section, we present three state-of-the-art deep learning-based missing data recovery methods. These methods are also used for image inpainting in the computer vision field and after explaining the principles of the methods, we give algorithms of each method for missing data recovery.

3.1 Generative Inpainting

Originally, the generative inpainting method is an optimized version of the method proposed in [31] and a similar configuration with [31] is followed. Input consisted of a couple of corrupted images and a binary mask. At the output, the recovered image is received. Input to the network is a masked image with a size of 256×256. Throughout the training, randomly generated rectangle masks are applied to the image. Also, an image with a different number of masks can be feed to the model.

It is worth mentioning that we preferred to utilize Generative Inpainting for missing data recovery since the method relies on Generative Adversarial Networks (GANs). It is known that GAN models are capable of producing in-existent images or parts of images. Missing data recovery with GANs is a field of image processing and is named image inpainting.

In order to take proposed contributions in [31] to further and improve the training, the authors present a more complex model. Essentially this model includes two sub-networks, coarse networks, and refinement networks. The coarse network takes a corrupted image as input and provides coarse predictions. These predictions are then forwarded to the refinement network in order to produce refined outputs. It is stated that input to the refinement network is a more hole-free image. Hence, compared with the coarse network, the encoder of the refinement network can grasp the image details better.

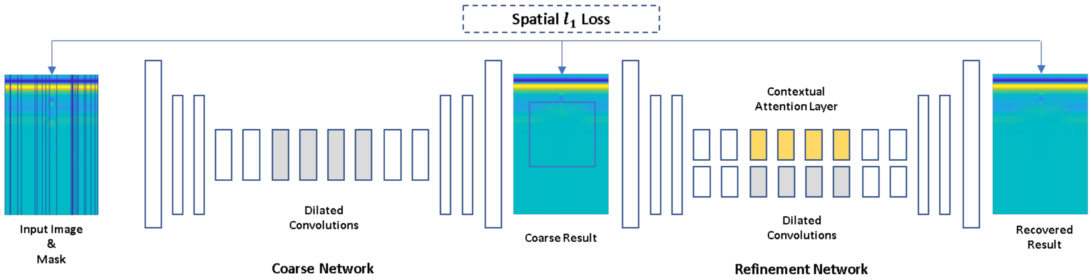


Figure 3.1 : Generative Inpainting Model

In generative inpainting, the refinement network is optimized by using reconstruction and GAN losses. And in this method, the authors combine the Wasserstein Generative Adversarial Network-Gradient Penalty (WGAN-GP) loss with the global and local outputs at the last part of the refinement network. The main goal of this combination is to preserve both the global and local image texture, similar to the [31]. To calculate the difference between the recovered and ground-truth image, $W(\mathbb{P}_r, \mathbb{P}_g)$, *Wasserstein-1* distance is used in WGAN models. This difference can be expressed as follows by exploiting *Kantorovich-Rubenstein* duality faithful to the representation in [41]

$$\min_{G} \max_{D \in \mathscr{D}} E_{\mathrm{x} \sim \mathbb{P}_r}[D(\mathrm{x})] - E_{\tilde{\mathrm{x}} \sim \mathbb{P}_g}[D(\tilde{\mathrm{x}})], \tag{3.1}$$

where assuming that z is the input given to the generator, and x̃ is the obtained model distribution from the generator via $G(z)$ function. Also $\mathscr{D}$ represents the set of 1-Lipschitz functions [41].

Using *Wasserstein-1* distance aside, another improvement in generative inpainting when compared with the model proposed in [31], is the implementation of attention propagation. In convolutional neural networks, a kernel is applied to an image in each layer. This detail obstructs exploiting information located in a distant spatial position in the image. In the generative inpainting method, the authors implement an Attention Propagation approach, to tackle this problem as depicted in Fig. 3.1. In order to calculate the attention score, in the first step, a similarity between two distinctive patches extracted from the image. The similarity between the two distinctive patches are calculated as a cosine similarity, which can be represented as follows

$$s_{x,y,x',y'} = \langle \frac{f_{x,y}}{||f_{x,y}||}, \frac{b_{x',y'}}{||b_{x',y'}||} \rangle, \tag{3.2}$$

Afterward, in order to calculate the attention score for each pixel, by counterbalancing the calculated similarity with λ constant as follows

$$s^*_{x+i,y,x'+i,y'} = softmax(\lambda s^*_{x+i,y,x'+i,y'}) \tag{3.3}$$

Here, an attention score is calculated by considering a fusion with left-right propagation and top-down propagation. Assuming that these propagation are done with a kernel, with a size of k, the attention score can be written as follows

$$\hat{s}_{x,y,x',y'} = \sum_{i \in \{-k,\ldots,k\}} s^*_{x+i,y,x'+i,y'}. \tag{3.4}$$

3.1.1 Generative Inpainting on GPR data

As it is stated previously, missing data situations are encountered frequently during GPR image acquisition. In conclusion, a deep learning-based missing data recovery method is examined thoroughly on GPR images. We divided our data set into 2 parts for training and testing. And we simply trained the Generative Inpainting model with training GPR images. At the end of the training, we obtained a model with learned weights.

Afterward, we used a test dataset in order to measure the performance of the trained model. We presented the measured performance and comparative analysis with the following two different deep learning-based missing data recovery methods.

Based on our experimental observations, results between the deep-learning-based missing data recovery methods vary. This variation may be up to the fact that our GPR data differs greatly from the images that these models are developed and tested. In a conclusion, in this thesis, we present that not every method is suitable for missing data recovery applications on GPR images.

Learned weights with the Generative Inpainting method on GPR data are given in Algorithm 1. *TrainSet* is generated at the first step by applying masks, U $\in \mathbb{R}^{m\times n}$ mask matrix , on GPR data, M$\in \mathbb{R}^{m\times n}$ and afterwards assigning the obtained corrupted GPR data to $TrainSet$.

As next *TrainSet* is forwarded to the model and until the epoch converges to its corresponding value, training is continued. At the end of the training, learned weights are obtained belonging to the Generative Inpainting model. In the end, the model with the learned weights is deployed with test images and results are recorded for further comparison.

Algorithm 1: Generative Inpainting training algorithm on GPR Images

---

Input : GPR image with missing data
Output : Learned Weights
Initialize M_i, $i \in \{1,...,70\}$, U_i,
$i \in \{1,...,70\}, TrainSet = [], epoch \in \{1,...,10000\}$, batch = 0;
while $i \leq 70$ **do**
  $M_i = X_i \odot U_i$;
  Increment i;
  Assign M_i to $TrainSet$;
end
while $epoch \leq 10000$ **do**
  Take a batch from the TrainSet ;
  while *batch* **do**
    Optimize the predictions obtained by using (3.1)*;*
    Refine predictions based on the attention score by using (3.4) *;*
  end
  Increase epoch ;
end

3.2 Pyramid-Context Encoder Network

In the next step, we continued our analysis with PEN-Net architecture. PEN-Net is an Auto-Encoder model and these models are widely used for image inpainting. Hence, because of our studies with Generative Inpainting architecture do not present promising results, as given in Section 4.4, in the next step we utilized an Auto-Encoder architecture for missing data recovery with a deep learning-based algorithm. Here, missing data recovery is regarded as one the fields of image processing with a deep learning-based algorithm, which is named as image inpainting.

PEN-Net consists of three main parts. The first part, Pyramid-context Encoder, is constructed with U-Net architecture [50]. Specifically, missing regions on an image are encoded with masks, which are constructed from learned semantic features.

While encoding latent features with the semantics of an image, missing regions are filled with a low-level feature map which is constructed by transferring rich details from a high-level feature map. This improvement is provided by embedding Attention Transfer Network (ATN) to each layer of the encoder since an ATN can memorize the similarity between patches that are located inside and outside of a missing region.

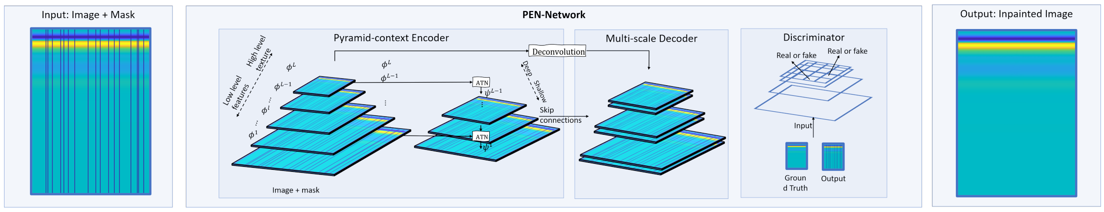


Figure 3.2 : Block diagram of the PEN-Net architecture [2]. Image is fed with mask, ATNs are applied in each layer of Pyramid-context Encoder. Afterwards this encoder is connected to a multi-scale decoder and results are forwarded to a discriminator. Discriminator tries to detect a fake image. When a fake image is plausible enough to the discriminator, image is forwarded as output image.

Between these patches, an ATN learns textures with high-level features, and this information is used for padding missing regions [40]. The purpose of this is to use the learned similarity as a map to build a feature transfer for missing regions, assuming that similar pixel semantics would lead to similar textures [2].

An illustration of PEN-Net architecture is given in Fig. 3.2. Faithful to the notations and definitions given in [2], we define $f(\cdot,\cdot)$ as ATN operation, ϕ as a feature map from deeper to shallow levels for L number of layers. Then, ATN applications can be formulated as follows [2].

$$\begin{aligned}
\psi^{L-1} &= f(\phi^{L-1}, \phi^{L}),\\
\psi^{L-2} &= f(\phi^{L-2}, \psi^{L-1}),\\
&\ldots,\\
\psi^{1} = f(\phi^{1}, \psi^{2}) &= f(\psi^{1}, f(\psi^{2}, \ldots f(\psi^{L-1}, \psi^{L}))).
\end{aligned} \tag{3.5}$$

For missing areas in the image, filling operations are performed (as in [40–42]) by simply transferring related features from outside regions to inside of the missing regions. The application of this step is based on learning high-level feature maps.

In the second part of the model, a multi-scale decoder is considered. Here, high-level textures are fed to the decoder directly from the pyramid-context encoder, and low-level features are fed from ATNs as depicted in Fig. 3.2. A combination of low-level features and high-level texture information provides building objects that are even missing in the masked region [2].

In the last part, a Patch-Generative Adversarial Networks (Patch-GAN) is integrated as the discriminator [51]. Here, the purpose is to create similar images such that the discriminator can not distinguish a real image from a fake image that the generator created.

As it is stated previously, missing data problem occurs frequently during GPR image acquisition. Hence in this paper, a deep learning-based missing data recovery approach is used as we exploited PEN-Net to improve missing data recovery performance on GPR images. The missing data problem is sub-categorized into two parts: Column-wise missing data, and pixel-wise missing data. Here, both of these cases are conducted via masking data on the image randomly. Three individual scenarios are taken into consideration with ratio of $10\%, 30\%$ and 50% missing data on the complete image.

Structure of the project is constituted with creating the dataset by masking GPR images and forwarding the dataset to PEN-Net architecture. After training the network individually with datasets, we performed missing data recovery on test images. In the next section, we present our results and compare with existing methods in the literature.

After obtaining patches from the feature map, cosine similarity is calculated between inside and outside patches as

$$s^l_{i,j} = \frac{x^l_i \cdot x^l_j}{||x^l_i||_2||x^l_j||_2}, \tag{3.6}$$

where x^l_i and x^l_j denote the patches that are obtained from the feature map and $||.||$ denotes norm operation . After this, a soft-max operation is applied on cosine similarity to obtain an attention score for every individual patch as follows [2]

$$\alpha^l_{j,i} = \frac{exp(s^l_{i,j})}{\sum_{i=1}^{N} exp(s^l_{i,j})}. \tag{3.7}$$

It should be noted that the attention score is used in a weighted approach to fill an adjacent low-level feature map. Here, the context is calculated from a high-level feature map as

$$x^{l-1}_j = \sum_{i=1}^{N} \alpha^l_{j,i} \;\; x^{l-1}_i \cdot, \tag{3.8}$$

where x^{l-1}_i is a patch from outside of the masked region and x^{l-1}_j is a patch from the missing region to be filled with weighted x^{l-1}_i.

As in [41], these operations can be defined in convolutional operations for automated learning, and low-level feature maps can be constructed using the similarity scores that are calculated from high-level feature maps.

3.2.1 PEN-Net on GPR data

As it is stated previously, missing data problem occurs frequently during GPR image acquisition. Hence in this thesis, another deep learning-based missing data recovery approach is used as we exploited PEN-Net to improve missing data recovery performance on GPR images.

And the structure of the thesis is consisted with creating the dataset by masking GPR images and forwarding the dataset to PEN-Net architecture. After training the network individually with datasets, we performed missing data recovery on test images.

The algorithm regarding missing data recovery with PEN-Net on GPR data is given in Algorithm 2. At first, specially designed masks, $U \in \mathbb{R}^{m \times n}$ mask matrix, are applied to input GPR data, $M \in \mathbb{R}^{m \times n}$ and corrupted GPR data are obtained. These data are assigned to $TrainSet$. Afterwards the complete $TrainSet$ is forwarded to the model. At the end of the specified epoch, learned weights are obtained.

Algorithm 2: PEN-Net training algorithm on GPR Images.

Input : GPR image with missing data
Output : Learned Weights
Initialize M_i, U_i, $i \in \{1, ..., 70\}, TrainSet = [], epoch = 0, batch = 0$;
while $i \leq 70$ **do**
    $M_i = X_i \odot U_i$;
    Append M_i to $TrainSet$;
    Increment i;
end
while $epoch \leq 10000$ **do**
    Take a batch from the TrainSet ;
    while *batch* **do**
        Reconstruct features with ATN application by using (3.5)*;*
        Refine predictions based on the attention score by using (3.6) *;*
    end
    Increase epoch ;
end

3.3 Deep Image Prior

Deep image prior is the last architecture that we utilized for missing data recovery, and it is relied on a SkipNet architecture. These architectures are used for image inpainting, super resolution and image restoration. Here, we utilized this model for missing data recovery and deployed the model to our GPR datasets.

Image generation with deep networks are generally designed with a generator and decoder networks, that can map a random vector y to an image X$\in \mathbb{R}^{m\times n}$ such that X$= f_\theta(y)$. In [52], the authors propose an architecture with a generator and decoder, that generates images from a random distribution and can be trained with back-propagation completely. However, on the contrary of such studies, in deep image prior distribution of the image is consisted with corrupted observation M. In this case, the goal is to present a solution for inverse problems such as denoising [53] or missing data recovery [54].

Prior of an image can be explained as our intuitive assumption regarding the ideal version of an image. Ideally an image is expected to be without a noise or a corruption. So for a neural network, image priors are used as references to perform a task such as denoising, missing data recovery or super-resolution. For this purpose, image priors can be generated in two methods. In first method, a neural network can be trained with a dataset, that is consisted of complete and corrupted images for missing data recovery application. Throughout the training, the model produces complete images at the output and it can be said that the model learns the prior from the dataset. And, priors that are obtained with this method called as learned priors. The drawback of this method is that a large dataset including both corrupted and complete images is necessary. In second method, priors are generated from the corrupted image itself and priors that are obtained with this method are named as explicit/handcrafted priors. In the scenarios where limited samples exist, handcrafted priors are more advantageous than learned priors, since handcrafted priors do not require a data set for training.

In deep image prior, it is intended to examine the image prior before learning any of the parameters, which is obtained with a certain generator network structure. This is handled by optimizing the neural network in terms of parameter set, for an image X$= f_\theta(y)$, where image is X$\in \mathbb{R}^{m\times n}$, code vector is represented as $y \in \mathbb{R}^{C'\times m'\times n'}$, and

```
=================================================================
Layer (type:depth-idx)                   Param #
=================================================================
Sequential                               --
├─Sequential: 1-1                        --
|    └─Sequential: 2-1                   --
|    |    └─Conv2d: 3-1                  816
|    └─BatchNorm2d: 2-2                  32
|    └─LeakyReLU: 2-3                    --
|    └─Sequential: 2-4                   --
|    |    └─Conv2d: 3-2                  6,416
|    └─BatchNorm2d: 2-5                  32
|    └─LeakyReLU: 2-6                    --
|    └─Sequential: 2-7                   --
|    |    └─Sequential: 3-3              1,030,016
|    |    └─BatchNorm2d: 3-4             128
|    |    └─Sequential: 3-5              18,464
|    |    └─BatchNorm2d: 3-6             64
|    |    └─LeakyReLU: 3-7               --
|    └─Upsample: 2-8                     --
├─BatchNorm2d: 1-2                       64
├─Sequential: 1-3                        --
|    └─Conv2d: 2-9                       4,624
├─BatchNorm2d: 1-4                       32
├─LeakyReLU: 1-5                         --
├─Sequential: 1-6                        --
|    └─Conv2d: 2-10                      17
├─Sigmoid: 1-7                           --
=================================================================
Total params: 1,060,705
Trainable params: 1,060,705
Non-trainable params: 0
=================================================================
```

Figure 3.3 : Model Graph

network parameters are represented as θ. Here, m and m' are used for height, n and n' for width and C' for code tensor. And the model is selected as SkipNet architecture with depth level as four and the model applies convolution, up-sampling and non-linear activation itself, as given in Fig. 3.3.

In deep image prior, inverse tasks are examined such as denoising or inpainting, in order to show power of parametrization. Inverse applications can be explained as energy minimization problems, as follows

$$X^* = \min_X E(X,M) + R(X), \tag{3.9}$$

where $E(X;M)$ represents data term of the missing data recovery task, M represents corrupted image and $R(X)$ term represents regularization term.

In these applications, regularization term is helpful for grasping the generic prior of images. However, most of the times regularization term is hard to identify. Simply, a regularizer can be a Total Variation (TV), which is the measure of the complexity of an image with respect to its spatial variation. In deep image prior application, this term is re-defined as the implicit image prior, which is created by the neural network as follows

$$\theta^* = \arg\min_{\theta} E(f_\theta(X;M)), \tag{3.10}$$

where

$$X^* = f_{\theta^*}(y). \tag{3.11}$$

Here, θ^* is the minimizer factor and gradient descent can be used as an optimizer to obtain θ^*. For a given parameter set θ^*, recovered image can be represented as X$^* =$ $f_{\theta^*}(y)$. Taking into consideration (3.10) in terms of (3.9), the regularizer *R(X) = 0* for all images that can be generated by a deep convolutional network with a certain architecture. Considering that the network is not pre-trained with any data in this approach, image priors can be generated explicitly.

Images with missing data can be expressed as Hadamard's product of a mask with an image. Assuming that the mask is composed with binary values such as $u \in \{0,1\}^{H\times W}$, then the data term of the masked image can be written as follows

$$E(X;M) = \|(X-M)\odot u\|^2, \tag{3.12}$$

where $(X-M)$ represents corrupted image and $\odot$ represents Hadamard's product. In (3.12), missing pixel values are not involved and if the optimization would be performed over pixel values of the image, energy value would remain steady. Because of this reason a data prior is required and it is calculated by optimizing the data term with respect to (3.10), as can be seen in Fig. 3.4.

In this application, SkipNet is selected as the model, since skip connection provided adequate flexibility to capture image semantics. SkipNet, is consisted of convolutional networks and based on a given input, specific layers are inserted or discarded. In this model gating modules that are positioned each layer, makes binary decisions based on the activations from previous layer. Binary decisions of gating modules are used to specify which layer will be run or discarded, as in Fig. 3.5. Simply, output of the gated layer or layers can be calculated faithful to the notations in [44] as follows

$$x^{i+1} = G^i(x^i)F^i(X^i) + (1-G^i(x^i))x^i, \tag{3.13}$$

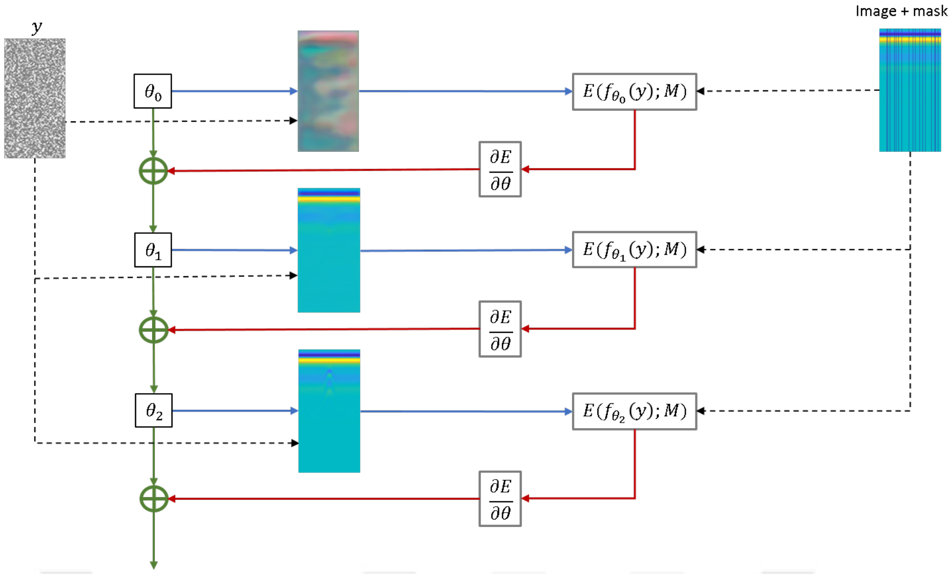


Figure 3.4 : Deep Image Prior Model

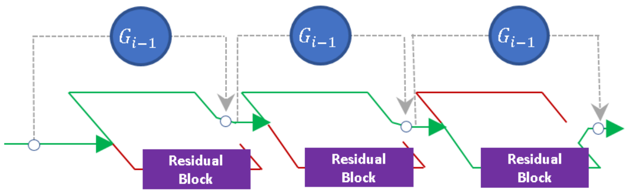


Figure 3.5 : SkipNet Architecture

where x^i is the input, $F^i(x^i)$ is the output of the i^{th} layer or layers and $G^i(x^i) \in \{0,1\}$ is the gating function of the i^{th} layer.

3.3.1 Deep Image Prior on GPR data

As it has been mentioned above, data loss commonly occur during collecting GPR data and as a conclusion GPR images are obtained with corruption. Hence recovery of missing data is an important priority for GPR images. In the literature, there are various applications of matrix completion for recovering data loss in GPR images. However we aimed to examine recovering missing data in GPR images thoroughly and provide a more comprehensive solution.

In order to resolve missing data problem extensively, we concentrated deep learning based missing data recovery methods. By exploiting the implication of deep learning based recovery methods can capture the image semantics, we proposed an approach relies on deep image prior. And in this thesis, deep image prior model is selected as the methodology for missing data recovery, based on both size of the available dataset and the technique's performances on publicly available datasets [32]. Considering that

Algorithm 3: DIP training algorithm on GPR Data

Input : **GPR image with missing data**
Output : **Recovered GPR image**
Initialize M_i, U_i, $i \in \{1,...,70\}$, $iteration = 0, i = 0$;
while $i \leq 70$ **do**
  $M_i = X_i \odot U_i$;
  while $iteration \leq 10000$ **do**
    Obtain optimized parameter set as given in (3.10)*;*
    Obtain new image prior with optimized parameter set as given in (3.9)*;*
    Increment iteration;
  end
  Increment i;
end

we aimed to cover an extensive analysis of missing data phenomenon in GPR images, we examined distinctive missing data cases. Here, we utilized missing data scenarios with block masks, as well as with masks that are created with randomly occluded pixels as in (3.12). Randomly occluding pixels can be handled as both occluding the pixels in a random position in the image, or occluding pixels in columns, that are selected randomly. Considering that data loss occurs without a pattern in GPR images, in this paper we employed our proposed approach with a variety of missing data cases.

At first step we generated input images and corresponding masks. As next, the complete pipeline is designed as feeding images and matching masks together to the network one by one. Each mask is applied on the image as formulated in (3.12), and resulting image is forwarded to the network as the input image. During the training, the recovery process starts with no prior image and continues until the iteration point, by producing recovered image prior at each iteration as depicted in Fig. 3.4. Finally produced image is compared with the ground truth image and error measurement conducted. We examined missing data recovery performance under ten individual cases and employed the pipeline that we defined above, with each of these case. In the next section, we share measured difference between the pixel values of the recovered and the ground truth images, we present visually comparable outputs, and plot loss curves of the model for each case. Also, we compare our results with existing methods in the literature. Additionally algorithm of deep image prior on GPR data is given in Algorithm 3. Here, each image $M \in \mathbb{R}^{m \times n}$, is masked with $U \in \mathbb{R}^{m \times n}$ mask matrix. Resulting corrupted image is forwarded to the model. Finally, recovered image is obtained at the end of the model.

4. EXPERIMENTAL STUDY AND RESULTS

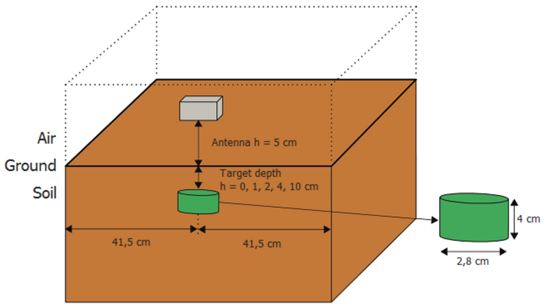
In this section, we present the performance metrics that we used for comparison between conventional and deep learning-based missing data recovery methods. Afterward, we present our implementation details for each of the deep learning-based missing data recovery methods. Later, we present our thoroughly conducted results with simulated data and real data. We show our analysis about object detection performance between the original and recovered data, and as of last we present a comparison between the methods in the literature and our findings.

4.1 GPR Datasets

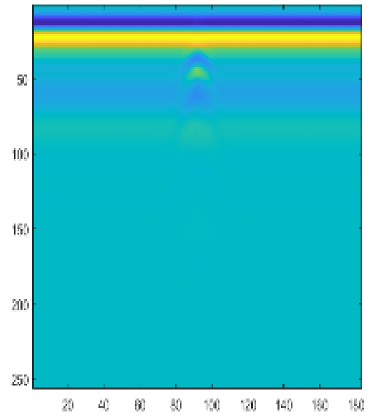
In this study, images used for missing data recovery tasks are generated by using the gprMax simulation software [55]. In the simulation setup, an aluminum target is placed. Also, surface and soil types are considered as flat and dry sandy soil. Designed simulation setup and sample generated image are depicted in Fig. 4.1. In order to present a comprehensive examination for missing data recovery, we considered missing data scenarios in 4-fold.

In the first fold, corrupted images are generated via occluding pixel values in an image randomly with different levels, as described in (3.12). These levels are selected as $10\%, 30\%, 50\%$ and 80% and corresponding images are given in Fig. 4.2. In the second fold, a similar approach is applied as occluding column-wise pixels and the same levels are used as in the first fold. Also, the images for column-wise missing data cases are depicted in Fig. 4.3.

In the third fold, instead of occluding pixels from the images randomly, a block-wise mask is applied. The block mask is chosen from the different locations of the images. The shape of the block mask is adjusted as $mask \in \mathbb{R}^{1\times H\times 10}$ and sample images are given in Fig. 4.17. In this scenario, 100 images with corresponding masks are generated out of 20 images, by masking every image 5 times randomly.

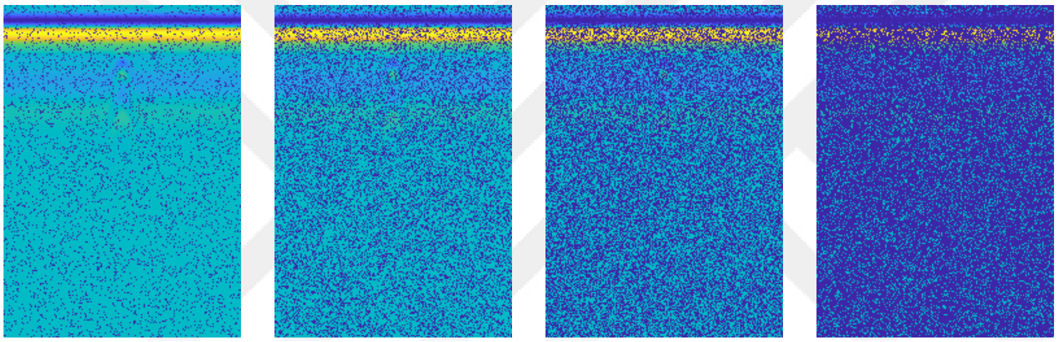


(a) Simulation Setup



(b) Generated Image.

Figure 4.1 : Simulation setup and correspondingly generated sample image



(a) (b) (c) (d)

Figure 4.2 : (a) 10% Missing Data in Pixel-Wise (b) 30% Missing Data in Pixel-Wise (c) 50% Missing Data in Pixel-Wise (d) 80% Missing Data in Pixel-Wise.

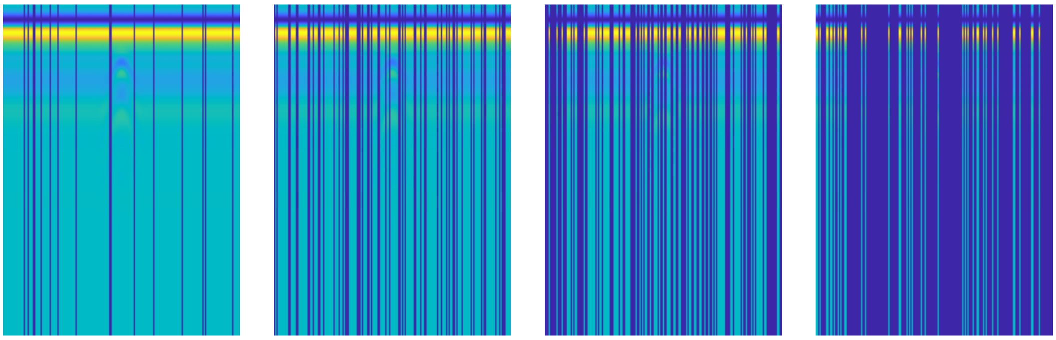
In the last fold, we used 2 real GPR data to validate our results that are obtained with simulation data. We applied with a 30% and 80% level for both pixel-wise mask and column-wise mask. Original images are depicted in Fig. 4.4.

To state it briefly, we defined eight individual cases in two folds with common missing data levels and one case in one fold with a block-wise mask. In the last case, we used real GPR data.

4.2 Performance Metrics

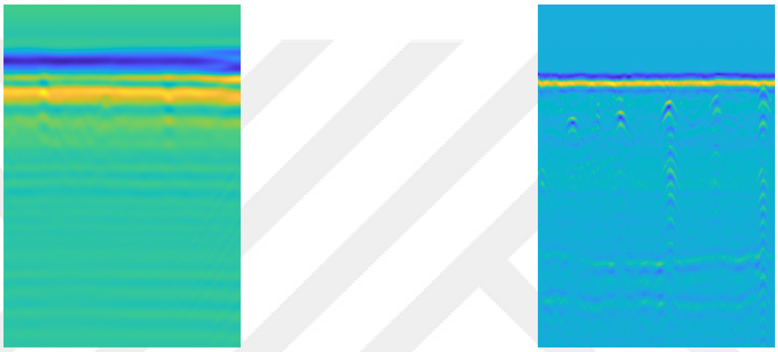
4.2.1 Peak Signal-to-Noise Ratio (PSNR)

In our study, we used two different metrics in order to compare the variance of the recovered images from the original images. And one of the metrics used on recovered



(a) (b) (c) (d)

Figure 4.3 : (a) 10% Missing Data in Column-Wise (b) 30% Missing Data in Column-Wise (c) 50% Missing Data in Column-Wise (d) 80% Missing Data in Column-Wise.



(a) (b)

Figure 4.4 : (a) Real GPR Image - 1 (b) Real GPR Image - 2.

images is the peak signal-to-noise ratio. In PSNR (dB) calculation, the following approach is used;

$$\begin{aligned} PSNR(dB) &= 10\log_{10}\left(\frac{\alpha^2}{MSE}\right), \\ &= 20\log_{10}\left(\frac{\alpha}{\sqrt{MSE}}\right), \end{aligned} \tag{4.1}$$

where α stands for maximum pixel value in the image and mean square error (MSE) can be calculated as follows

$$MSE = \frac{1}{n}\sum(y-\hat{y})^2, \tag{4.2}$$

where $(y-\hat{y})^2$ constitutes squared difference between the actual value of the pixel and the pixel value in the inpainted image.

4.2.2 Structural Similarity Index (SSIM)

Also, we used the structural similarity index (SSIM) as the second metric for comparison between recovered and original images. SSIM is a perceptual metric that

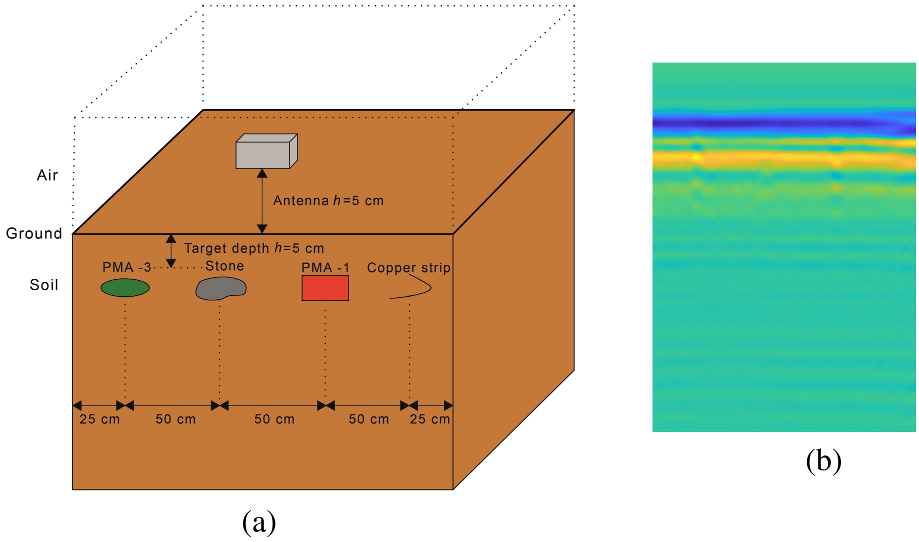


Figure 4.5 : (a) The experimental setup of the scenario, (b) Original real GPR data.

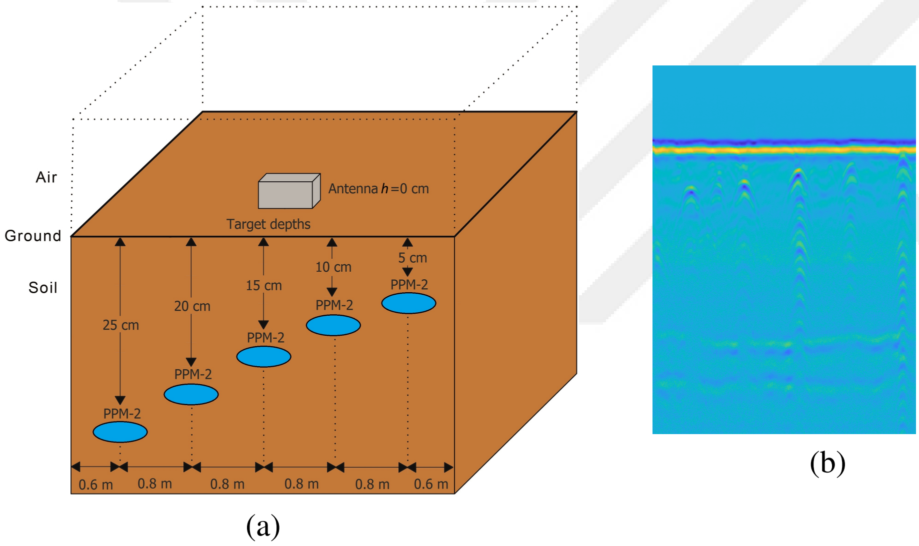


Figure 4.6 : (a) The experimental setup of the scenario, (b) Original real GPR data.

quantifies image quality caused by losses or corruption in the image. SSIM metric is calculated as follows

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{4.3}$$

and

$$\begin{aligned} c_1 &= (k_1 L)^2 \\ c_2 &= (k_2 L)^2, \end{aligned} \tag{4.4}$$

where μ_x, μ_y, σ_x^2, σ_y^2, σ_{xy} and L denote average of x, average of y, variance of x, variance of y, co-variance of x and y, and the dynamic range of the pixel values of the image, respectively. Here, c_1 and c_2 are calculated as follows

where $k_1 = 0.01$ and $k_2 = 0.03$ by default.

4.2.3 Common Objects in Common (COCO)

In order to compare the power of the missing data recovery, object detection is applied on both ground truth and recovered images. In object detection, the goal is to validate the performance of the proposed method by presenting detected objects on images. This validation is represented with 12 individual metrics for characterizing the performance of the detection. Metrics and their corresponding functions are given in Table 4.1.

In order to decide whether a prediction of object detection is correct with respect to an object or not, intersection over union (IoU) is used. In object detection, predicted bounding box B_p and ground truth bounding box B_{gt} represent the box of the designated object and the box of the detected object, respectively. And, IoU is the ratio of the area of the intersection of these boxes to the area of the union of these boxes which is calculated as follows

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}. \tag{4.5}$$

In object detection, interpretations of the predictions are done as follows,

- **True Positive:** A correct detection of a ground-truth bounding box (TP)
- **False Positive:** An incorrect detection of a in-existing object or a detection of an object in a wrong position in the image (FP)
- **False Negative:** An undetected ground-truth bounding box (FN)

With these interpretations, precision and recall notions are introduced and incorporated in performance calculation as follows

$$Pr = \frac{\sum_{n=1}^{S} TP_n}{TP+FP} \quad , \quad Rc = \frac{\sum_{n=1}^{S} TP_n}{TP+FN} \quad , \tag{4.6}$$

where S is the total number of bounding boxes. Here, TP, FP and FN are identified with a threshold limit t as follows

$$Prediction = \begin{cases} True\ Positive & IoU \geq t \\ False\ Positive & IoU < t. \end{cases} \tag{4.7}$$

By substituting (4.7) into (4.6), Average Recall is defined over all points as follows

$$AP_{all} = \sum_{n} (R_{n+1} - R_n) P_{interp}(R_{n+1}), \tag{4.8}$$

where

$$Pr_{interp}(R_{n+1}) = \max_{\tilde{R}:\tilde{R}\geq R_{n+1}} P(\tilde{R}). \tag{4.9}$$

In (4.8), AP is calculated as the interpolation of the precision at each level where the maximum precision of the points with higher recall value than R_{n+1}. By taking the average of the AP over the dataset, the mean AP (mAP) is obtained as follows

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i, \tag{4.10}$$

where AP_i represents the the AP of i^{th} class and N is the total number of the classes, that exist in the dataset. Calculations of the metrics given in Table. 4.1 rely on (4.10).

Table 4.1 : COCO Metrics.

Notation	Definition
AP	Average Precision
$AP^{IoU=0.50}$	$t = 0.50$ threshold level for IoU
$AP^{IoU=0.75}$	$t = 0.75$ threshold level for IoU
AP^{small}	Evaluation result of small ground-truth objects (area < 32^2 pixels)
AP^{medium}	Evaluation result of medium sized ground-truth objects (32^2 < area < 96^2 pixels)
AP^{large}	Evaluation result of medium sized ground-truth objects (area > 96^2 pixels)

4.3 Implementation

In this thesis, the pipeline is implemented and executed on our workstation running a 64-bit Windows 10 Intel Core i7-8700K CPU with a 3.70-GHz clock, 64 GB RAM, and an NVIDIA GeForce RTX2080 Ti Super GPU. Python3 is used as a programming language with PyTorch deep learning framework.

In generative inpainting, Spectrally Normalized for Generative Adversarial Network (SN-GAN) is selected as the deep learning architecture. For padding same padding is selected. The number of iteration is set as 10000, the learning rate is set to 0.01.

In the PEN-Net method, originally designed architecture is used as described in Section 3.2. Here, for optimizer Adam optimizer is selected. The number of iteration is set as 10000, the learning rate is set to 0.01.

In deep image prior, SkipNet with $dept = 4$ is selected as the deep learning architecture. For padding and optimizer, zero paddings and Adam optimizer are selected. The number of iteration is set as 10000, the learning rate is set to 0.01.

Number of the parameters in the model is calculated as $1,060,739$, as depicted in Fig.3.3.

4.4 Results

In this section, we presented our results obtained with both simulated data and real data. We examined and compared the methods in 4-fold. We used GoDec, LmaFit, NNM, and NMC methods as conventional methods for comparison and GI, PEN-Net, and DIP methods as deep learning-based methods.

Based on the observations of the missing data recovery results with simulated data, we observe that the GI method performs poorly and require longer training, compared with the alternative deep learning-based methods for missing data recovery. This requirement causes an infeasible training time, such as more than 150 hours. We presented GI results with also column-wise missing data recovery. However similar to the pixel-wise missing data recovery results, column-wise missing data did not result competitively, compared with PEN-Net and DIP methods. Hence, we continued our analysis with the most successful deep-learning-based method, which is the DIP method.

As a conclusion, we presented our results with pixel-wise and column-wise missing data with the comparison of four conventional methods and three deep learning-based methods. Afterward, we continued our analysis for block-wise missing data with the comparison of four conventional methods and one deep learning-based method, which is selected as the DIP method based on the outscored performance both in SSIM and PSNR values, and visual interpretations. We finalized our further analysis with missing data recovery on two individual real data. And we presented the object detection performance of the recovered images with the DIP method and compared the performance with the object detection results on the original images.

4.4.1 Simulated dataset

In this section, analyses that are performed by using a simulated dataset are thoroughly explained. These analyses include pixel-wise missing data, column-wise missing data, and additionally block-wise missing data scenarios.

Table 4.2 : SSIM and PNSR comparison of missing data recovery methods for pixel-wise information loss.

	Missing rate (%)	GoDec	NMC	LmaFit	NNM	GI	PEN-Net	DIP
SSIM	10	0.9999	1.0000	1.0000	**1.0000**	0.6884	0.9998	**1.0000**
	30	0.9998	0.9999	1.0000	**1.0000**	0.6884	0.9999	**1.0000**
	50	0.9997	0.9997	1.0000	**1.0000**	0.5564	0.9999	**1.0000**
	80	0.9963	0.9912	0.9935	0.9972	0.5895	0.9999	**0.9981**
PSNR	10	68.2841	78.5118	97.2125	111.2272	16.3839	**102.2172**	**118.0045**
	30	63.2072	67.6718	90.8240	92.2051	17.3651	**104.3120**	**117.8161**
	50	61.9782	62.1258	72.9267	69.4741	17.8765	**73.2647**	**93.2235**
	80	47.2469	42.0223	43.6032	48.3413	18.6670	45.7896	**57.7669**

4.4.1.1 Pixel-wise missing data

In pixel-wise missing data scenario, randomly selected pixels are occluded with levels of $10\%, 30\%, 50\%$ and 80%, as depicted in Fig. 4.2. Results are evaluated in both quantitative and visual aspects. Considering visual results, only 80% missing data perform poorly and objects in the recovered images are observed as more obscure. For quantitative analysis, obtained results for pixel-wise missing data cases are given in Table. 4.2 for each level and method. Compared with the column-wise missing data recovery results given in Table. 4.3, it can be said that pixel-wise missing data recovery is easier than column-wise missing data recovery. Also two out of the three deep-learning-based missing data recovery methods that we implemented outperform the other conventional methods in all missing data levels. And the deep learning-based missing data recovery method with the highest PSNR resulted such as

- In 10% missing data case, average DIP result is 6.1% better than the most successful conventional method (NNM)
- In 30% missing data case, average DIP result is 27.8% better than the most successful conventional method (NNM)
- In 50% missing data case, average DIP result is 27.8% better than the most successful conventional method (LmaFit)
- In 80% missing data case, average DIP result is 17.8% better than the most successful conventional method (NNM)

As can be seen in the Table.4.2, depending on the level of the missing data, the most successful method varies between the conventional methods. However, deep

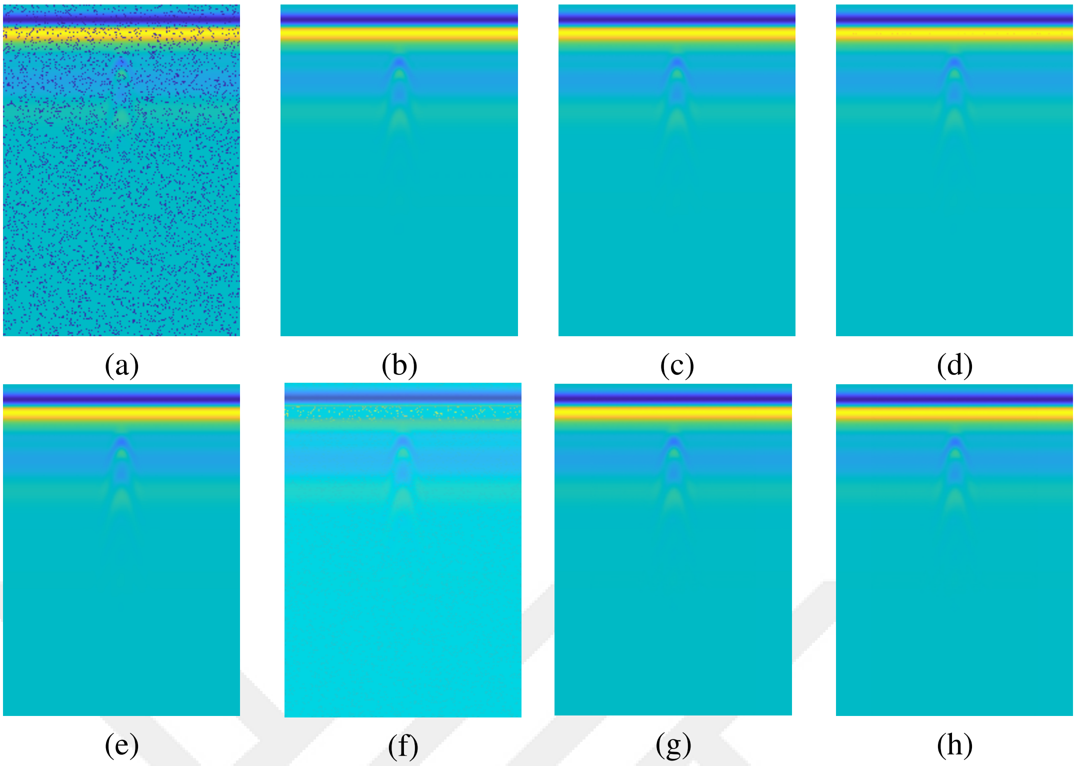


Figure 4.7 : Pixel-wise Missing Data Recovery. (a)-(f) 10% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

image prior outperforms other alternative methods with an approach based on either conventional methods or deep learning. We observe similar SSIM values from 10% to 50% missing data cases. As expected, values tend to decrease as the corruption rate increases.

When we go through PSNR values given in the first row of the Table 4.2, in the 10% missing data case we observe dramatic differences between the results of the conventional methods and results of the PEN-Net and DIP method. In Fig. 4.7a, 10% missing data image is depicted and recovery results are given in Fig. 4.7b-4.7h. In conclusion, since the data loss is not at a critical level for 10%, an observable difference does not occur between the recovered images with each method. However, for the GI method we observe that occluded pixels can not be recovered properly and as a conclusion, we obtain very low SSIM and PSNR values with this method. When we compare PEN-Net and DIP methods, we observe that for 10% missing data scenario,

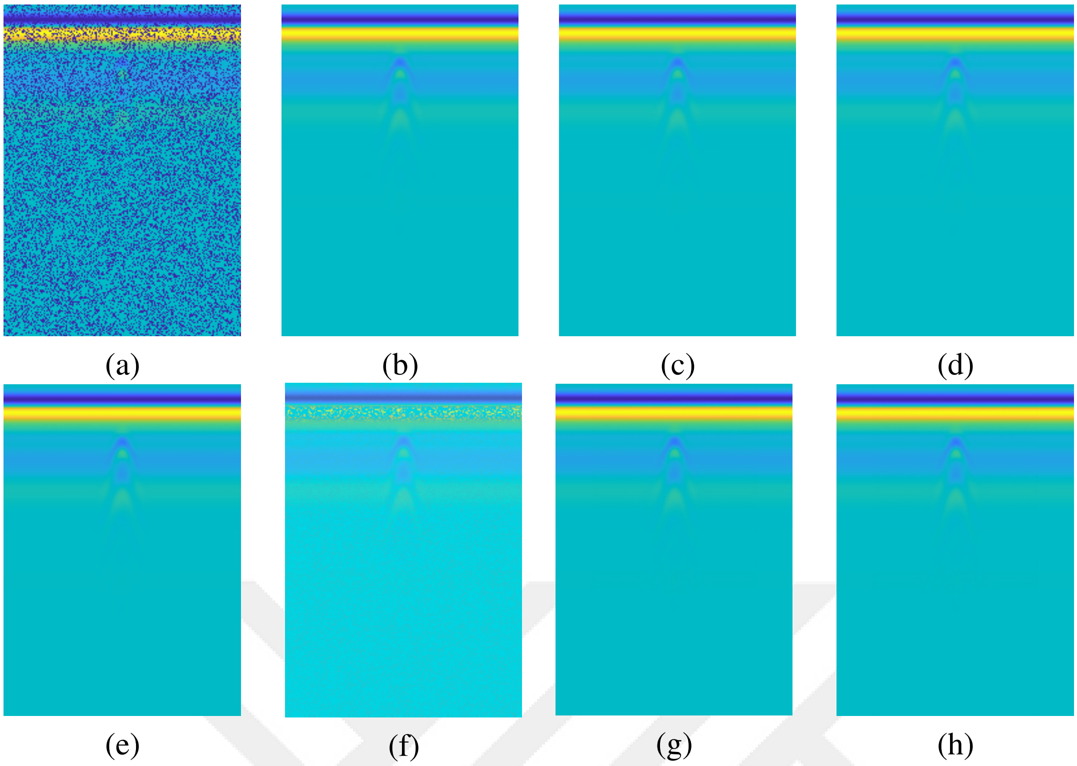


Figure 4.8 : Pixel-wise Missing Data Recovery. (a)-(f) 30% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

DIP performs better in numerical analysis, however, still no distinct difference is observed between the given figures, as depicted in Fig. 4.7.

A similar situation is observed also between the different results for 30% and 50% missing data case. Specifically, SSIM values are obtained with a slight difference. On the other hand, PSNR values vary between the conventional methods and deep learning-based methods results with the highest PSNR value, except the GI method. 30% and 50% pixel-wise missing data images are depicted in Fig. 4.8 and Fig. 4.9, respectively. The recovered images with each method depicted in Fig. 4.8b-4.8h.

Based on the SSIM and PSNR values, given in Table 4.2, occluding the GI method, it can be seen that GoDec and NMC methods results are not at a competitive level with LmaFit and NNM methods. And PEN-Net and DIP methods outperform all the conventional methods, under the circumstances of DIP resulted in the highest values in both PSNR and SSIM. On the other hand, keeping the GI method aside, slight visual

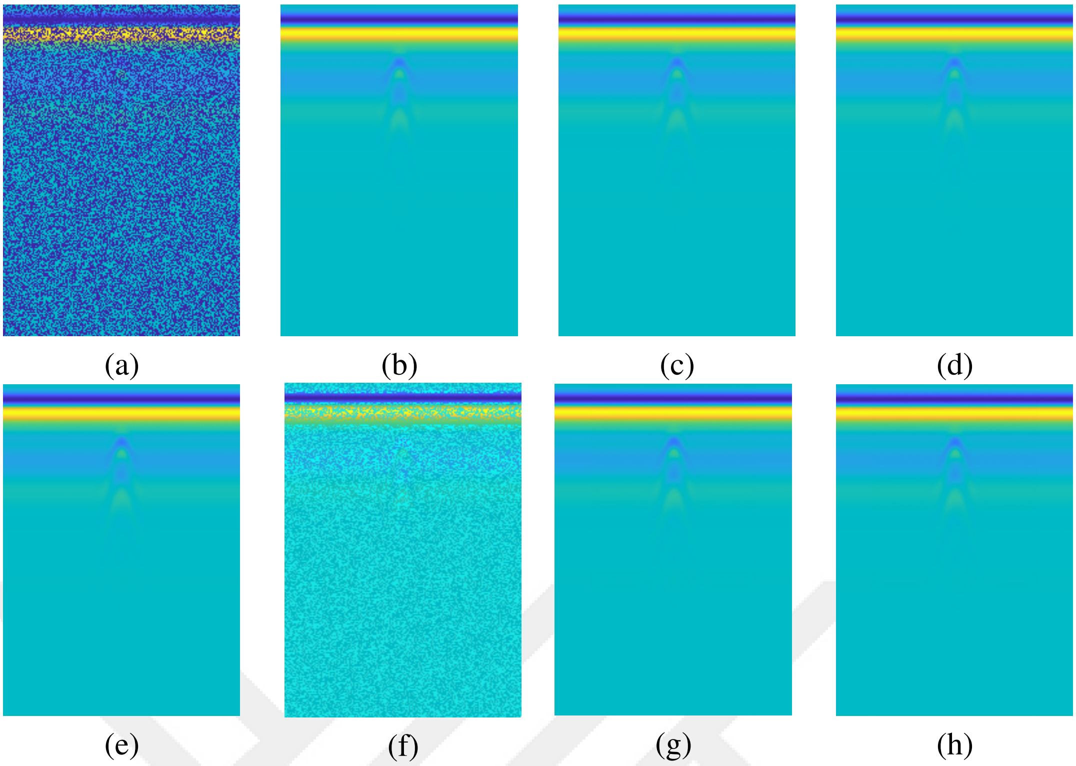


Figure 4.9 : Pixel-wise Missing Data Recovery. (a)-(f) 50% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

differences are observed between the recovered images of each method, as illustrated in Fig. 4.8.

And for 50% pixel-wise missing data case, even though the PSNR values vary, an observable difference can be hardly detected as seen in Fig. 4.9b-4.9h. Similar to the previous results, the GI method performs even worse with 50% pixel-wise missing data.

The predominant effect between the results is observed in 80% pixel-wise missing data case. Here, the variance between SSIM values increases, while still, the DIP method provides the highest SSIM value. On the other hand, differences between PSNR values decrease and conventional methods present similar performances. And the most crucial result is captured on the visual differences between the results of each method. In Fig. 4.10a, 80% pixel-wise masked image is depicted and results of conventional methods and deep learning-based methods are presented in Fig. 4.10b- 4.10h. Based on the observations, it can be said that 80% pixel-wise missing data can be regarded as a boundary condition and conventional methods suffer providing important details

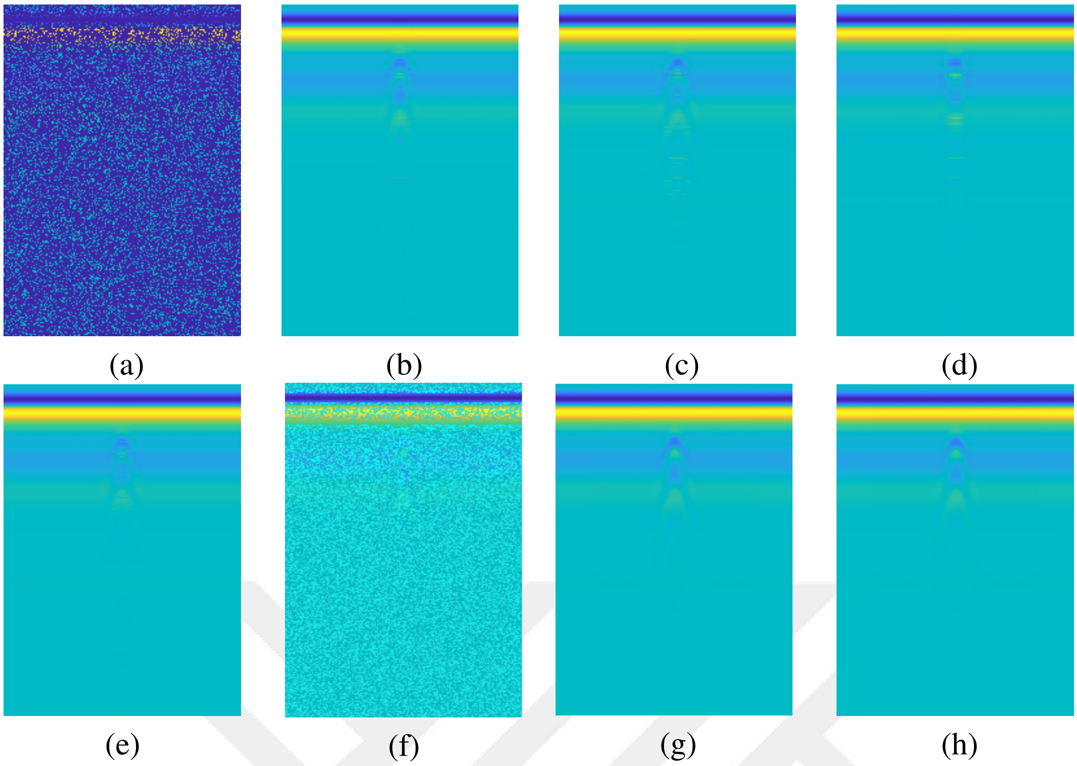


Figure 4.10 : Pixel-wise Missing Data Recovery. (a)-(f) 80% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

of the image. And we observe a decrease in the capability of the conventional method, whereas PEN-Net and DIP methods still can afford critical details in the image.

Furthermore, we aimed to present the effect of the level of data loss on recovering missing data. We plotted loss curves that we obtained by exploiting our proposed method. In Fig. 4.11, we observe different curvatures for different levels of missing data. With the higher corruption on the image, the loss function converges to the highest error value. Also, it requires dramatically longer iterations to converge to a steady value. The highest convergence is observed on the softest case, 10% pixel-wise missing data. In the long run, the error value converges to a lower value. Hence, for our proposed method, the direct relationship between the decreasing trend in PSNR values can be represented with the loss curves. And we obtain consistent curvatures with numerical and visual results.

4.4.1.2 Column-wise missing data

In column-wise missing data scenario, randomly selected columns are occluded from the image with levels of $10\%, 30\%, 50\%$ and 80%, as given in Fig.4.3. Recovered

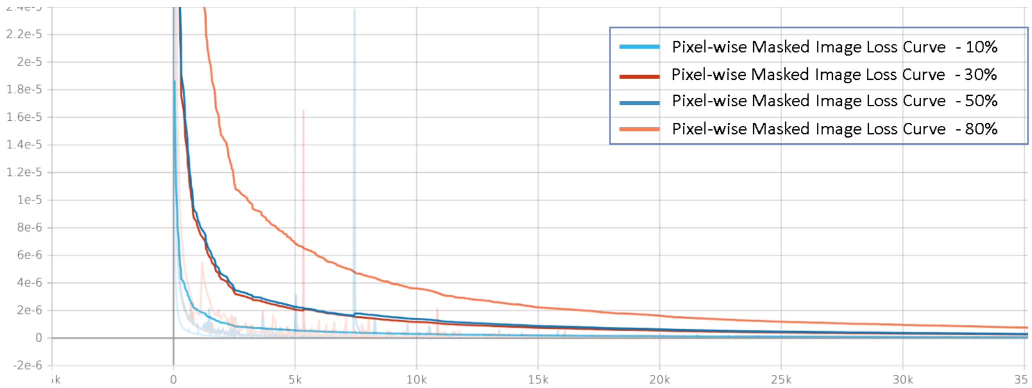


Figure 4.11 : Pixel-wise Missing Data Recovery Loss Curves

images are examined in quantitative and visual aspects. Based on the visual interpretations of the corrupted images in Fig. 4.3a- 4.3d, as the level of the missing data increases, objects become more and more obscure and the missing data recovery becomes harder for data recovery methods. The comparisons are conducted both visually and quantitatively. Since visual results are subjective and can be analyzed by trained eyes, the quantitative results will present objective comparison. This observation also can be verified with the descending trend in the calculated PSNR values for the DIP method, as given in Table 4.3. Another observation can be stated when we compare our findings as given in the Table.4.3, PSNR values are obtained less than pixel-wise missing data recovery scenarios. When we want to summarize the results listed given in Table 4.3,

- In 10% missing data case, average DIP result is 10.8% better than the most successful conventional method (NMC)
- In 30% missing data case, average DIP result is 6.4% better than the most successful conventional method (NMC)
- In 50% missing data case, average DIP result is 14.3% better than the most successful conventional method (NMC)
- In 80% missing data case, average DIP result is resulted with the highest SSIM and PSNR values, since conventional methods could not recover from this high level of missing data

On the contrary to the pixel-wise masking results, in the column-wise masking case, we observe lower SSIM and PSNR values. Since the information on the column

Table 4.3 : SSIM and PNSR comparison of missing data recovery methods for column-wise information loss.

	Missing rate (%)	GoDec	NMC	LmaFit	NNM	GI	PEN-Net	DIP
SSIM	10	0.9993	0.9999	0.9985	0.9998	0.5478	0.9986	**0.9998**
	30	0.9979	0.9996	0.9984	0.9984	0.5298	0.9981	**0.9997**
	50	0.9982	0.9984	0.9982	0.9969	0.5041	0.9981	**0.9986**
	80	–	–	–	–	0.3684	0.7245	**0.9913**
PSNR	10	53.4921	64.4189	50.5649	61.8392	16.3721	55.0267	**71.3785**
	30	46.8300	59.6113	50.3825	54.4345	17.1214	53.4962	**63.4336**
	50	49.5139	50.4303	49.5313	49.2968	17.0896	54.6790	**57.6261**
	80	–	–	–	–	18.0046	30.5452	**31.1949**

is lost completely, it is harder to recover from column-wise missing data for the methods. SSIM and PSNR values given in first row of the Table 4.3, 10% missing data image is depicted in 4.12a and recovered images with each method are given in Fig. 4.12b-4.12h. Since the data loss is not at a critical level for 10%, an observable difference does not occur between the recovered images with NMC, NNM, PENT-Net, and DIP methods. However, GoDec and LmaFit methods produce recovered images with low-resolution. Hence, based on the visual observation it can be said that even for a very low missing data rate, some of the matrix completion methods are not suitable for column-wise missing data recovery. On the other hand, in the first run other than the GI method, both PEN-Net and DIP methods perform well and produce satisfactory recovered images. Only the GI method can not afford to recover from also 10% column-wise missing data, similar to the pixel-wise missing data. Also when we compare the deep learning-based methods, we see that the DIP method outperforms PEN-Net.

Similarly, we observed slight differences in SSIM and PSNR results between the methods for 30% and 50% missing data cases. SSIM and PSNR values tend to decrease through 30% to 50%. Parallel to the measured PSNR values, conventional methods start to fail to recover column-wise missing data as depicted in Fig. 4.13b-4.13h and in Fig. 4.14b-4.14h, respectively. Even though both of the NMC and NNM methods yield satisfactory results in visual aspects of previous cases, these methods start to output low-resolution recovered images at the 30% missing data level. In this scenario, also GoDec and LmaFit methods produce images with poor resolution. And when we employ conventional methods with 50% column-wise missing data, we observe deteriorated recovery results. While the object is still somehow recovered with

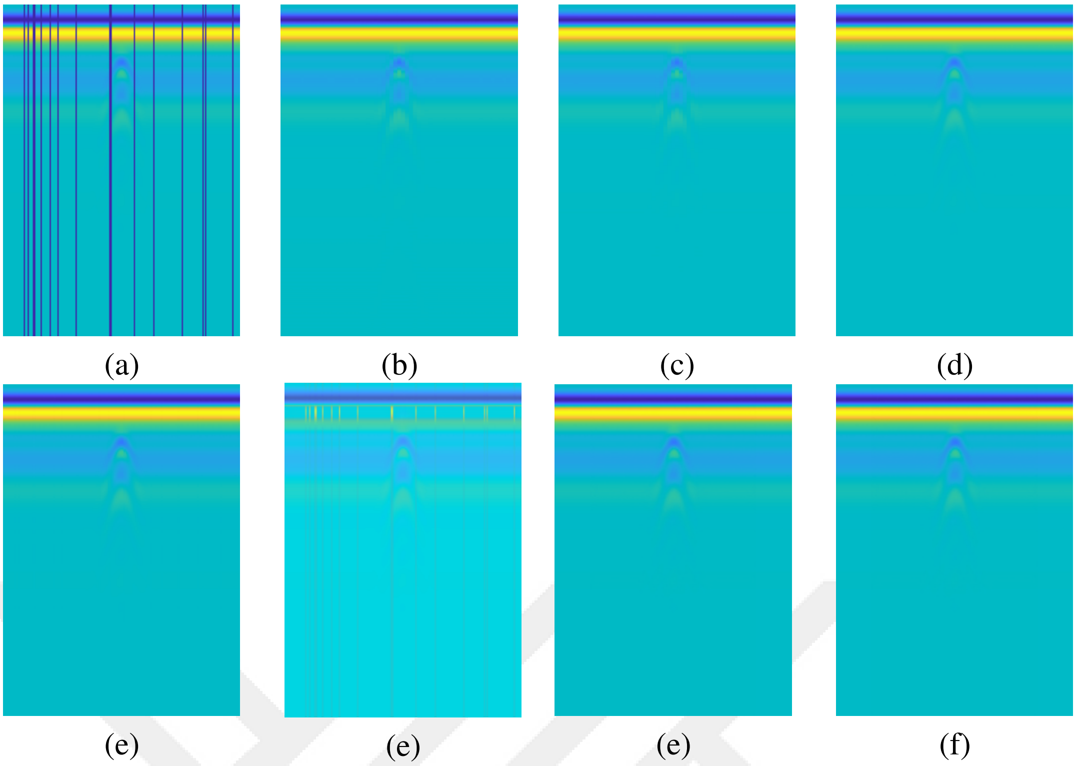


Figure 4.12 : Column-wise Missing Data Recovery. (a)-(f) 10% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

conventional methods, PEN-Net and DIP produce smoothly recovered images in both of the 30% and 50% missing data cases. Similar to the experiences from previous cases, the GI method continues to perform poorly for column-wise recovering missing data.

The most dramatic effect is observed in the 80% column-wise missing data case. In this case, masked input image is illustrated in Fig. 4.15a and as given in Fig. 4.15b-4.15h, conventional methods fail to produce a considerable result. Yet, the DIP method is capable of producing an adequate result in quantitative and visual aspects. Here, we also observe unsatisfactory results in both quantitative and visual results for the PEN-Net method. In a conclusion, it can be said that with the DIP method missing data recovery can be even succeeded at the boundary conditions or when a conventional method fails to recover missing data.

Moreover, since the DIP method produces the most successful results, we aimed to present the effect of the different levels of data loss on recovering missing data with DIP. We plotted loss curves that we obtained by exploiting the DIP method. In Fig. 4.16, we observe different curvatures for different levels of missing data. With

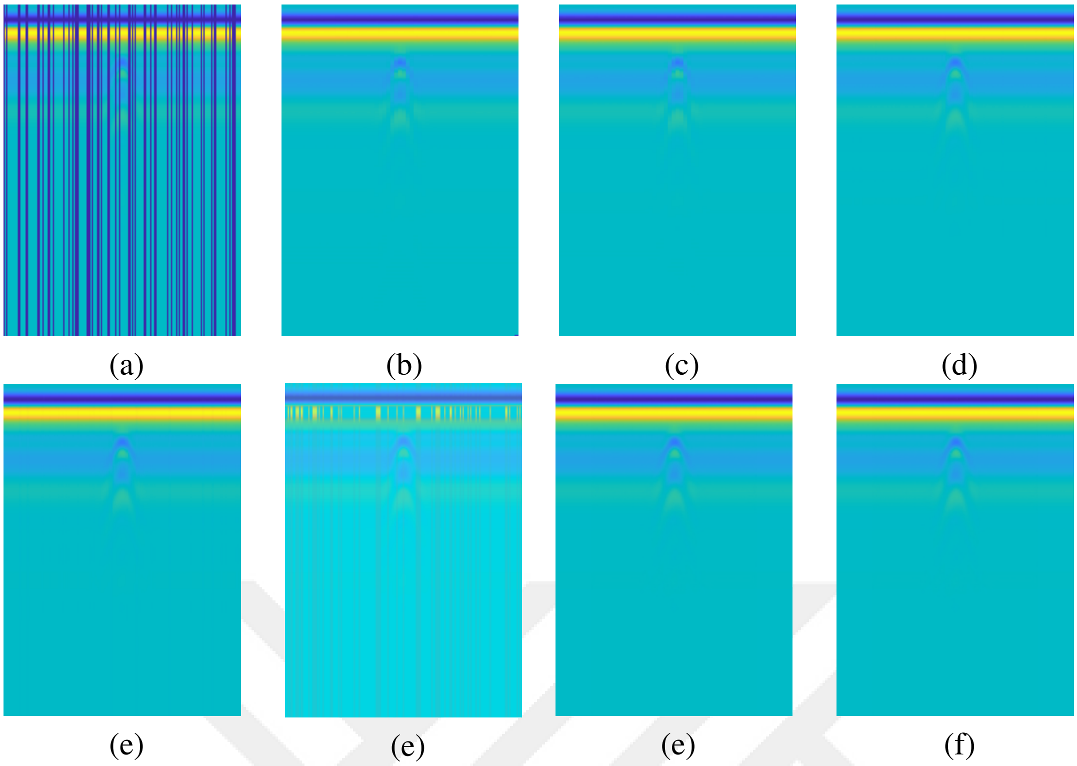
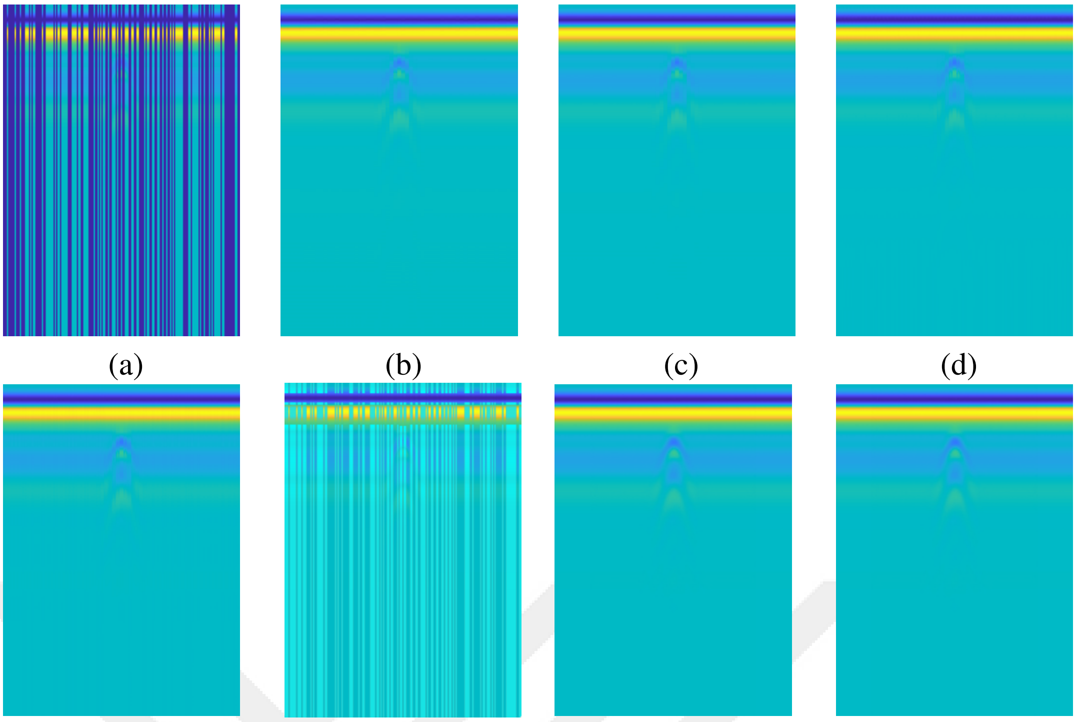


Figure 4.13 : Column-wise Missing Data Recovery. (a)-(f) 30% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

the higher corruption on the image, the loss function converges to the highest error value. Also, it requires dramatically longer iterations to converge to a steady value. The highest convergence is observed on the softest case, 30% column-wise missing data. In the long run, the error value converges to a lower value. However, since column-wise missing data is a more challenging recovery case, loss curves of the 50% and 80% overlap and converges to their minimum levels after 15000 iterations. Also, a decreasing trend in PSNR values is interrelated with the loss curves. And we obtain consistent curvatures with numerical and visual results.

4.4.1.3 Block-wise missing data

In this scenario, masks are applied on images as formulated in (3.12). Here, the main goal is to examine the effect of uniform data loss that can be occurred randomly. In order to explore the performance of the proposed method on the edge levels, the width of the mask is kept as ten continuous column-wide. Images with 5.78% continuous data loss are depicted in Fig. 4.17a- 4.17d. Similar to other scenarios, obtained results are evaluated in visual and quantitative aspects. Loss curves for the sample masked



(a) (b) (c) (d)

(e) (e) (e) (f)

Figure 4.14 : Column-wise Missing Data Recovery. (a)-(f) 50% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

images that are depicted in Fig. 4.17a-4.17d, are plotted in Fig. 4.18. Also recovered images are depicted in Fig. 4.17e-4.17h. In average, PSNR and SSIM values of recovered images are measured as 37.4558 and 0.9945, respectively.

In order to examine the obtained results also with the visual aspect, block-wise masked images and recovered images are presented in Fig. 4.17a- 4.17d and in Fig. 4.17e- 4.17h, respectively. Based on the visual analysis point of view, it can be said that satisfactory results are obtained. Even when the object in the image is partially masked, successful recovery results are observed, i.e. in Fig. 4.17h.

We also plot the loss curves of the sample recovered in Fig. 4.17a- 4.17d since we aimed to present a thorough interpretation. As depicted in Fig. 4.18, when the image contains less complex patterns or objects or block-wise mask coincides with a low-profile part of the image, the loss curve converges rapidly. Similarly, when the block-wise mask coincides with the most complex pattern or objects on the image, the curve converges the latest.

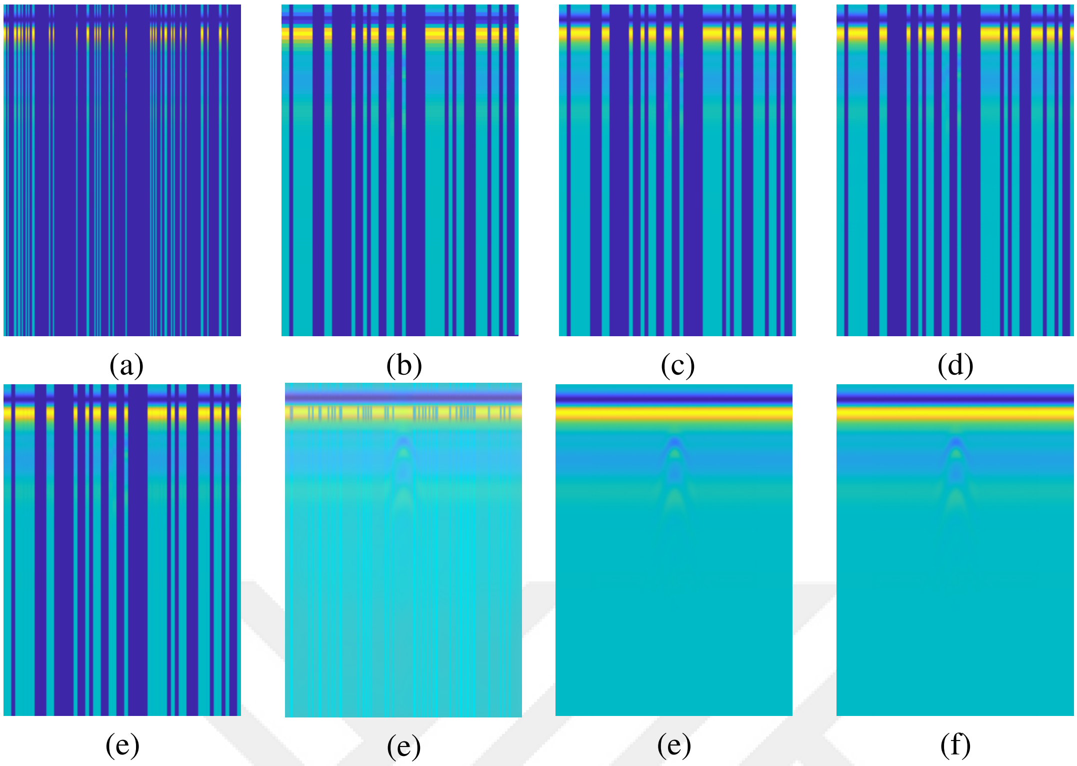


Figure 4.15 : Column-wise Missing Data Recovery. (a)-(f) 80% Masked Image, GoDec, LmaFit, NMC, NNM, GI, PEN-Net and DIP method results, respectively.

4.4.2 Real datasets

In this section, analyses that are performed by using real GPR data are thoroughly explained. These analyses include pixel-wise missing data and column-wise missing data scenarios.

4.4.2.1 Real dataset - 1

To further test the performance of the methods, two field test results are provided. The first real data is obtained from Vrije Universiteit Brussel. The setup of the experiment is given in Fig. 4.5a and the raw data is presented in the first row of Figure. This scenario contains four targets buried in the same direction. These are PMA-3, PMA-1, stone, and copper strip targets are buried at 5 cm depth (the PMA-3 and the PMA-1 refer to the plastic case antipersonnel landmines). The soil type is dry clay soil and it is mixed with small rocks. The surface type is rough and irregularities are a maximum of 10 cm between the highest and the lowest point. An antenna is located 5 cm above the highest point of the surface and a 1 GHz GPR antenna is used for the experiment.

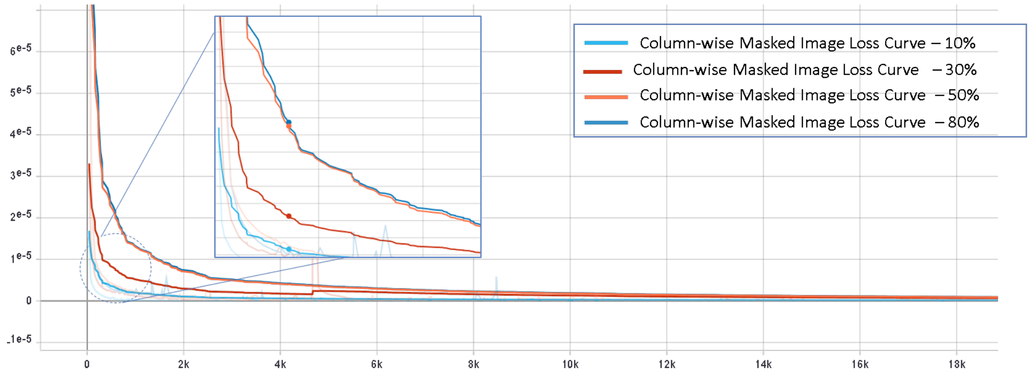


Figure 4.16 : Column-wise Missing Data Recovery Loss Curves

Table 4.4 : SSIM and PNSR comparison of missing data recovery methods for pixel-wise information loss on real GPR image.

	Missing rate (%)	GoDec	NMC	LmaFit	NNM	DIP
SSIM	30	0.9982	0.9987	0.9979	**0.9993**	**0.9993**
	80	0.9918	0.9920	0.9884	0.9919	**0.9934**
PSNR	30	54.8025	56.7148	54.2867	59.5118	**60.8923**
	80	45.9468	45.6161	44.9762	45.8010	**51.1180**

The scanned area is 50 cm on the X-axis and 196 cm on the Y-axis and the scanning step of 1 cm in each direction.

In the analysis of the first real GPR image, both pixel-wise mask and column-wise mask applied images with 30% and 80% missing data. In order to exhibit the performance of the missing data recovery of the methods, we designated two cases with a low and high level of missing data.

For pixel-wise missing data, 30% and 80% missing data images are illustrated in Fig. 4.19a and Fig. 4.20a, respectively. Measured SSIM and PSNR values are given in Table 4.4. In both of the missing data levels, our proposed method results with the highest SSIM and PSNR values. For 30% missing data case, as depicted in Fig. 4.19b-4.19f, recovery of the missing data is achievable. Similarly, our proposed method produces satisfactory results in visual aspects, as given in Fig. 4.19f. Hence for the 30% pixel-wise missing data case, it is ambiguous to present a distinction between the conventional methods and our proposed method.

In 80% pixel-wise missing data case, conventional methods result in similar PSNR values. However, with a deep learning-based approach, an increase in PSNR is observed. PSNR value obtained with DIP is 11.3% greater than the result obtained

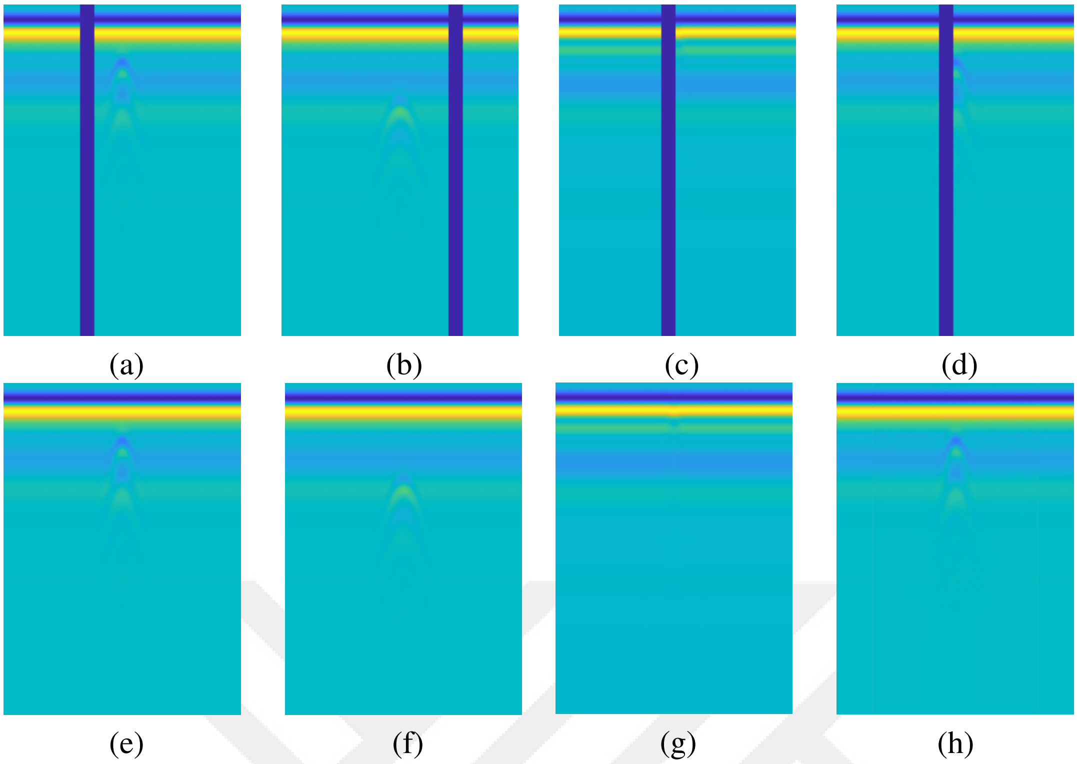


Figure 4.17 : Block-wise Mask Scenario (a)-(d) Images with masks (e)-(h) Recovered images with DIP.

with the highest conventional method (GoDec). Also, observation of visual results concludes satisfactory recovery of the missing data. Even with an 80% data loss, it can be said that data recovery with our proposed method yields successful results, as depicted in Fig. 4.20a. And we observe that conventional methods yield low resolution recovered images, as given in Fig. 4.20b-4.20e. On visual results stand of point, slight differences between results are observed, whereas our method provides output with higher PSNR values.

In column-wise masking, a higher variation between conventional methods is observed as given in Table 4.5. According to the results, conventional methods can recover from 30% column-wise missing data. Nevertheless, our proposed method outperforms conventional methods and the PSNR value obtained with our method is 7.7% greater than the result obtained with the highest conventional method (NMC). For visual evaluation, 30% column-wise missing data is given in Fig. 4.21a. Also recovered images with conventional methods and our proposed method are depicted in Fig. 4.21b-4.21e and Fig. 4.21f, respectively. Also for the visual examination, slight differences between outputs are observed.

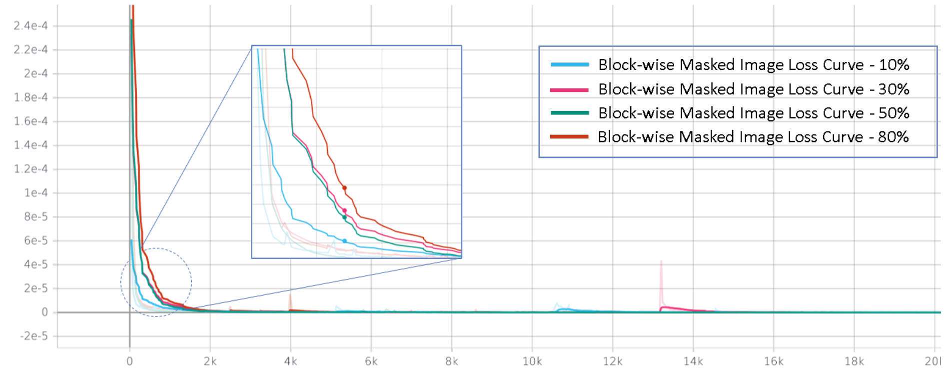


Figure 4.18 : Loss curves of sample images

Table 4.5 : SSIM and PNSR comparison of missing data recovery methods for column-wise information loss on real GPR image.

	Missing rate (%)	GoDec	NMC	LmaFit	NNM	DIP
SSIM	30	0.9964	0.9984	0.9964	0.99973	**0.9993**
	80	0.1023	0.1023	0.1050	0.1052	**0.9755**
PSNR	30	51.3586	54.7924	51.4660	52.7910	**58.9900**
	80	8.0566	8.0567	8.0579	8.0577	**33.6373**

On the contrary, when we employ all models with 80% column-wise missing data, both quantitative and visual results differ greatly. A big difference is measured SSIM and PSNR values, as given in Table 4.5. In this scenario, 80% column-wise missing data applied image is illustrated in Fig. 4.22a. Recovered images with conventional method are given in fig.4.22b-4.22e, and recovered image with deep image prior is given in Fig. 4.22f. Here, our model outperforms all conventional methods and produces an output with 3$\times$ higher PSNR. Similarly, visual interpretation of the recovered image with our proposed method yields solid results, whereas conventional methods fail to produce a considerable output.

Consequently, based on the PSNR values and visual outputs that we presented, it can be said that our proposed method has effective results on missing data recovery with real GPR images. Even with the high data loss levels such as conventional methods can not sustain to produce an output, loss data can be recovered nobly with our method.

4.4.2.2 Real dataset - 2

The experimental design of the second real scenario is given in Fig. 4.6a and the raw data result is presented in Fig. 4.6b. Field test results are obtained from landmine

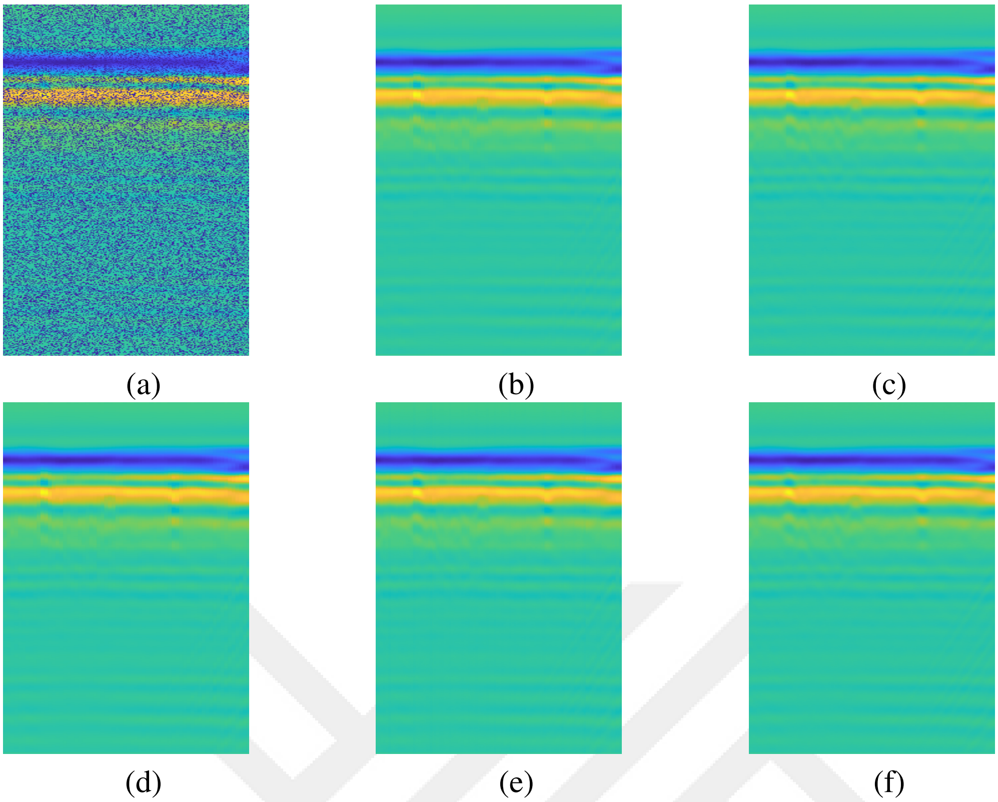


Figure 4.19 : Pixel-wise missing data recovery on real GPR image. (a)-(f) 30% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

clearance operations conducted in Germany as a project of the International Test and Evaluation Program for Humanitarian Demining (ITEP) in 2009. In this particular test, a commercial GPR antenna (SIR-3000 with 1.5-GHz antennas, GSSI) was used and it is located immediately over the targets and the antenna was moved over the center of the landmines with a broadside perpendicular orientation. Five rendered-safe landmines (PPM-2) which had a diameter of approximately 13 cm were buried in the same directions with positions of 0.6, 1.4, 2.2, 3.0, and 3.8 m. The burial depths of the objects are 25, 20, 15, 10, and 5 cm, respectively. The soil type is magnetic sand which is an artificial mixture of calcareous sand and engineered magnetite in order to replicate soil with high magnetic susceptibility. The texture is a coarse sand with a small amount of fine gravel (2% to 5%).

In the analysis of the second real GPR image, a similar structure of the first real GPR image is followed. Both pixel-wise mask and column-wise mask applied images with 30% and 50% missing data.

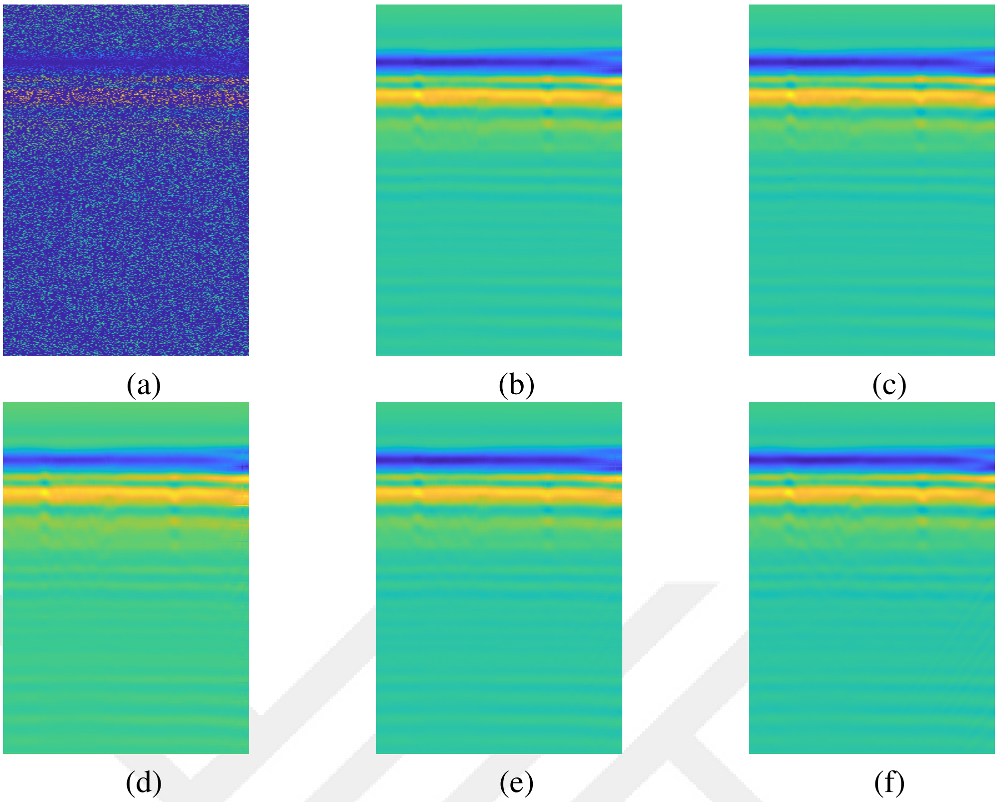


Figure 4.20 : Pixel-wise missing data recovery on real GPR image. (a)-(f) 80% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

For pixel-wise missing data, 30% and 80% missing data images are given in Fig. 4.23 and Fig. 4.24 and SSIM and PSNR values are given in Table 4.6. In both of the missing data levels, our proposed method results better than the conventional methods in SSIM and PSNR values. For 30% missing data case, as depicted in Fig. 4.23b-4.23e, recovery of the missing data can not be performed with all conventional methods. Unlike this situation, our proposed method produces satisfactory results in the visual aspect, as given in Fig. 4.23f. Hence for 30% pixel-wise missing data case, other than the NNM method, conventional methods and our proposed method produce considerable output.

In 80% pixel-wise missing data case, conventional methods result in similar yet too low PSNR values. However, with a deep learning-based approach, an output with an acceptable PSNR is produced. PSNR value obtained with DIP is 8.6% greater than the result obtained with the highest conventional method (NNM). However, observation of visual results shows the unsatisfactory recovery of the missing data. In the case of 80% pixel-wise data loss, it can be said that data recovery with our proposed method

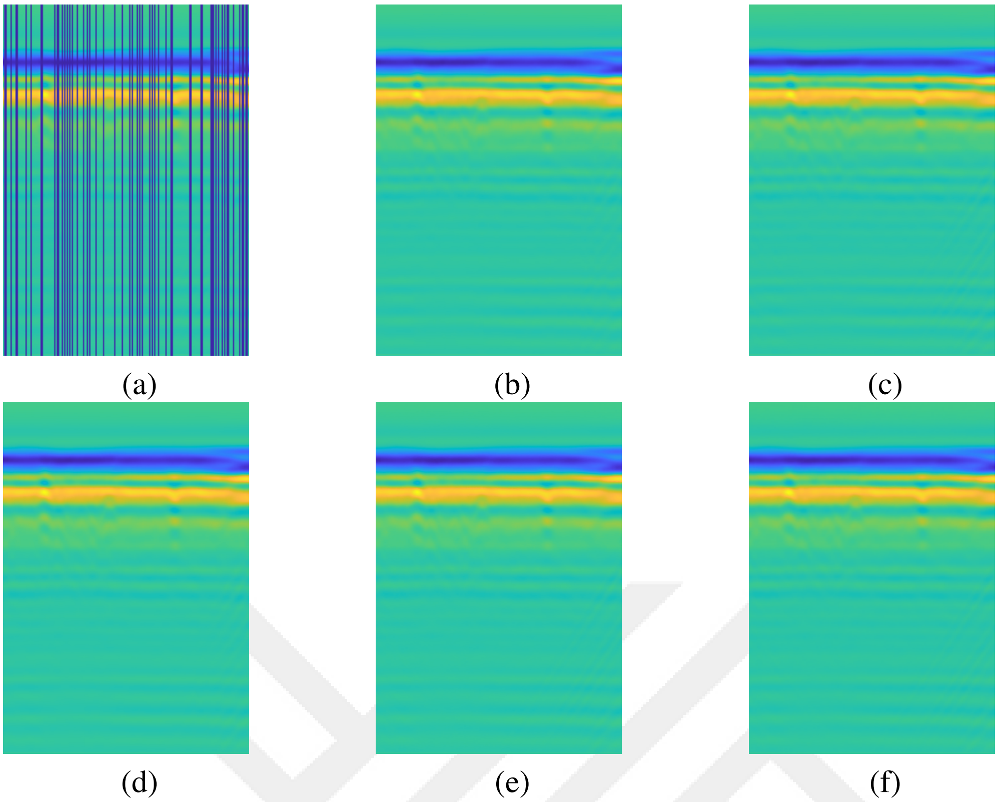


Figure 4.21 : Column-wise missing data recovery on real GPR image. (a)-(f) 30% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

yields successful results, as depicted in Fig. 4.24f. And we observe that conventional methods yield low resolution recovered images, as given in Fig. 4.24b-4.24e. On visual results stand of point, major differences between results are observed, and our method provides output with higher PSNR values.

In column-wise masking, a higher variation between conventional methods is observed as given in Table 4.7. According to results, while conventional methods can hardly recover from 30% column-wise missing data, our proposed method outperforms conventional methods with 7.3% greater than the result obtained with the highest conventional method (NMC). For visual evaluation, 30% column-wise missing data is given in Fig. 4.25a and recovered images with conventional methods and our proposed method are depicted in Fig. 4.25b-4.25e and Fig. 4.25f, respectively. In regards to the visual examination, slight differences between outputs are observed.

In real GPR image evaluation, we observe the strongest difference in 80% column-wise missing data scenario similar to the simulation GPR images. 80% column-wise missing data is depicted in Fig. 4.26a. According to the measured PSNR results,

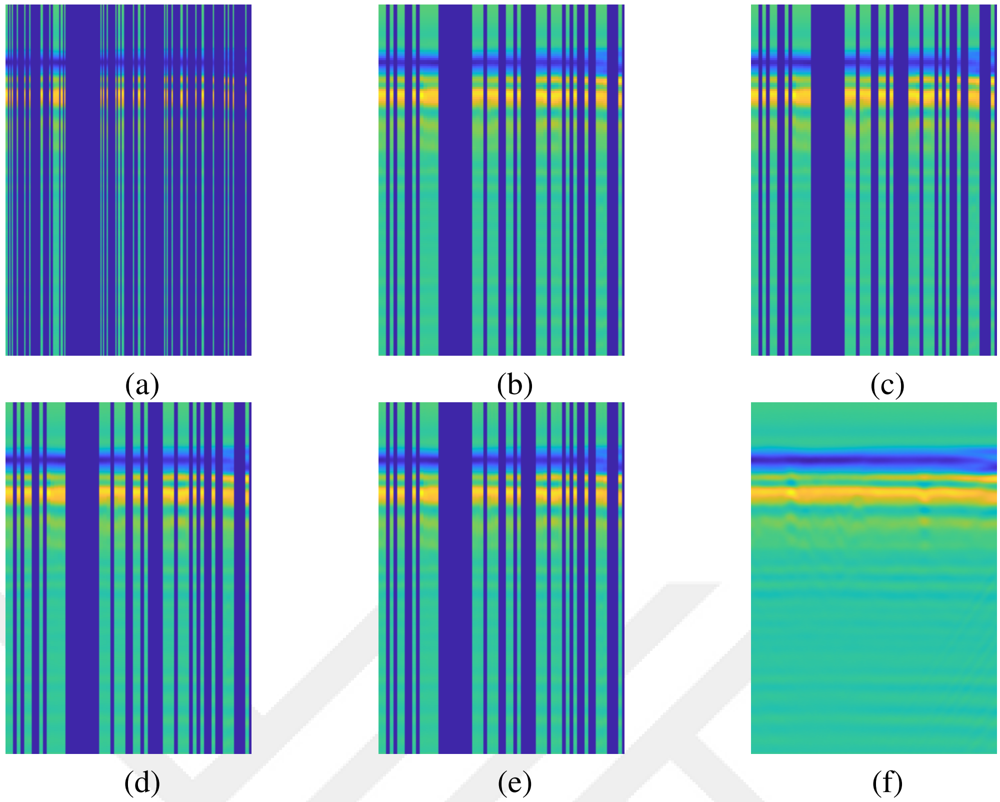


Figure 4.22 : Column-wise missing data recovery on real GPR image. (a)-(f) 80% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

given in Table 4.7, deep image prior's values is nearly $4\times$ greater than conventional methods. As an indication of this difference, we observe that results obtained with conventional methods deteriorate and an output can not be produced, as illustrated in Fig. 4.26b-4.25e. On the other hand, our method can handle even 80% column-wise missing data as given in Fig 4.25f.

4.4.3 Object detection over recovered images

In this thesis, in order to present the efficiency of the deep image prior data recovery method, we exploited the approach in [56] for object detection on both original images and recovered images. In [56], it is proposed that detecting objects on GPR images can be performed by using a neural network model, specialized for object detection. One of these models is given as Faster Region-based Convolutional Neural Network (Faster R-CNN). In [57], authors propose Faster R-CNN by concatenating Region Proposal Network (RPN) and Fast R-CNN as a single network, as depicted in Fig. 4.27.

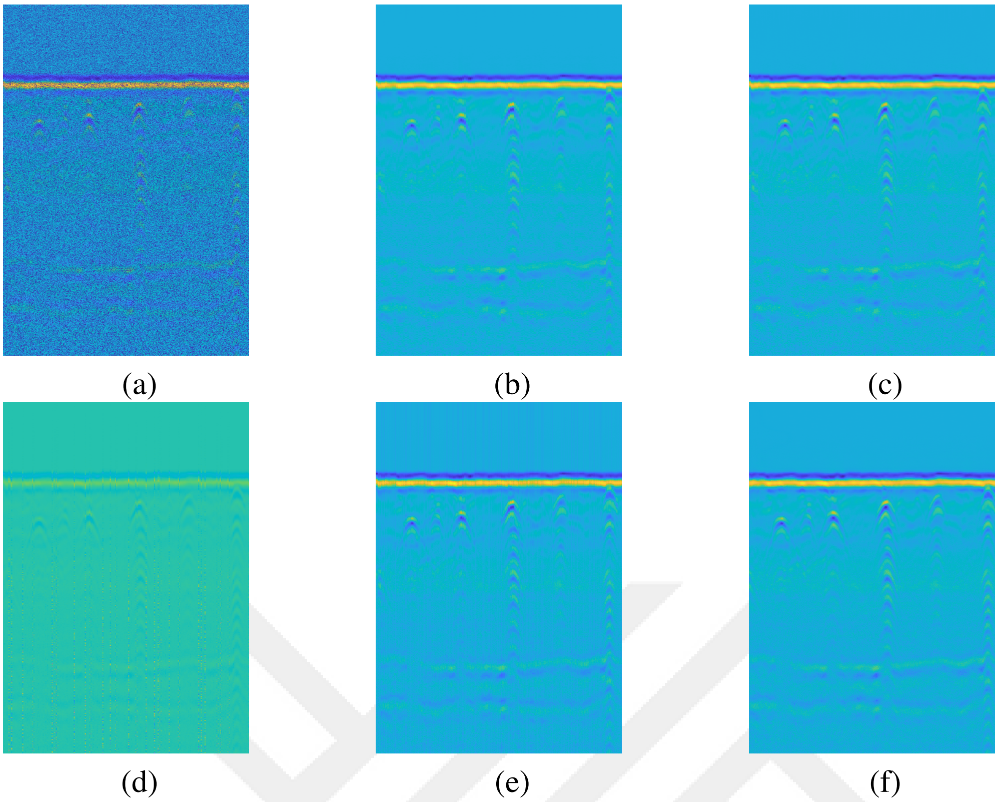


Figure 4.23 : Pixel-wise missing data recovery on real GPR image-2. (a)-(f) 30% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

In [58], buried objects are detected pm GPR data by exploiting a deep learning-based model which is named as Faster Region based Convolutional Neural Network (Faster R-CNN). Also in [56], it is proposed that detecting objects on GPR images can be improved when it is combined with clutter removal.

Faster R-CNN is an optimized version of object detection models such as R-CNN and Fast R-CNN. In this model, a search selective algorithm is conducted with RPN. Here, a feature map is created with a pre-trained neural network and in the next step, these features are forwarded to RPN. This step is regarded as feature extraction. Afterward, RPN tries to find a defined number of bounding boxes by using the feature map that is produced by the pre-trained network. With RPN, a common problem of generating a variable-length list of bounding boxes is resolved by substituting anchors. These anchors resolve this problem by questioning, the anchor contains a relevant object, or not. With the information of relevant bounding boxes, Region of Interest (RoI) pooling takes place and corresponding features are converted into RoI feature vector. Finally, on the R-CNN part, bounding boxes are classified based on the content (Softmax),

Table 4.6 : SSIM and PNSR comparison of missing data recovery methods for pixel-wise information loss on real GPR image-2.

	Missing rate (%)	GoDec	NMC	LmaFit	NNM	DIP
SSIM	30	0.7121	0.7520	0.7382	0.9172	**0.9568**
	80	0.6507	0.5977	0.6168	0.7099	**0.7118**
PSNR	30	31.6770	31.9981	31.8770	35.9859	**39.5690**
	80	29.6588	27.8841	27.7967	30.4092	**33.0378**

Table 4.7 : SSIM and PNSR comparison of missing data recovery methods for column-wise information loss on real GPR image-2.

	Missing rate (%)	GoDec	NMC	LmaFit	NNM	DIP
SSIM	30	0.7923	0.8628	0.5552	0.8592	**0.9113**
	80	0.0896	0.0897	0.0899	0.0971	**0.8976**
PSNR	30	32.3113	33.9112	19.2242	33.3317	**36.3700**
	80	10.0645	10.0648	10.0649	10.0603	**39.6029**

and fine-tuning for the position of the bounding box is performed (Bbox Regressor). In [57], authors propose Faster R-CNN by concatenating Region Proposal Network (RPN) and Fast R-CNN as a single network, as depicted in Fig. 4.27.

We implemented the model by simply training over the objects in the training set, objects in various sizes and lucidity can be detected in the test set. We utilized the pipeline with simulation images and divided the data as 70% for training, 10% for validation, and 20% for the test. In Table 4.8; the first row represents the detection rates obtained with the test set of the original GPR images. In the second row, recovered GPR images are used as test sets and sample images with object detection are depicted in Fig. 4.28.

As can be seen in Table 4.8, it is observed that no major difference is measured between the detection rates of original data and recovered data. According to the results, hyperbola detection could be performed better. However, when we look at the Average Precision values of the different areas, we see that the Average Precision of medium size area and large size area are higher in the recovered images. And for the small-sized areas, the opposite way around applies. This suggests that even though the Average Precision of the detected objects is less successful, recovered images include adequate detail for correctly detecting the regions.

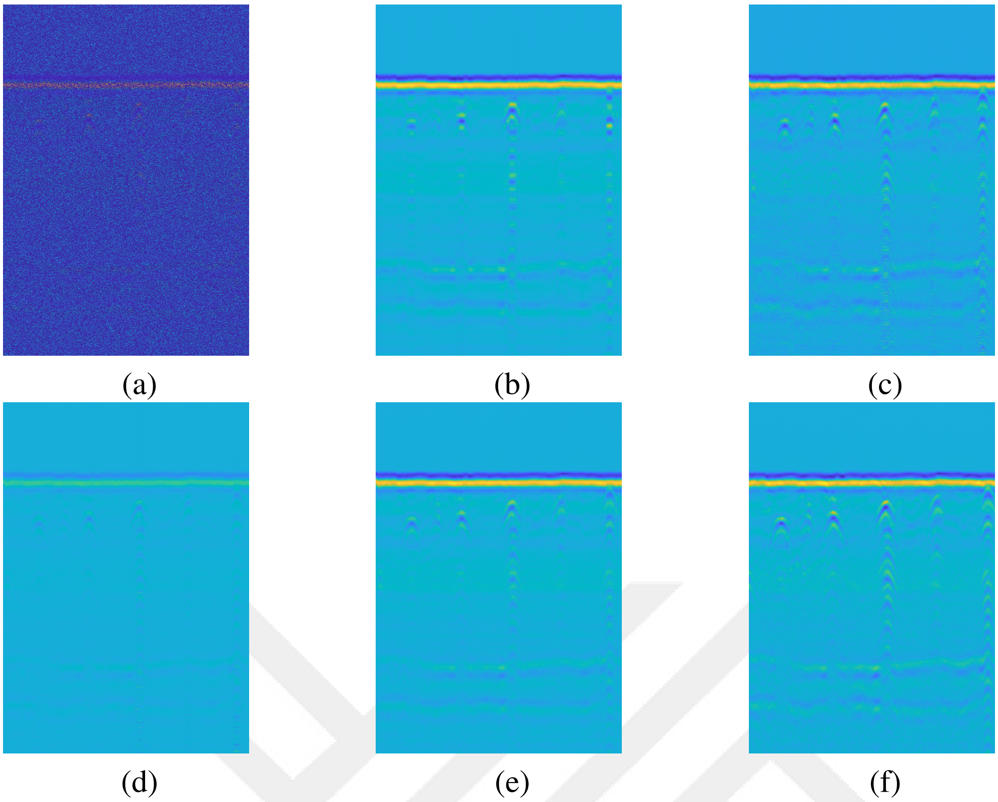


Figure 4.24 : Pixel-wise missing data recovery on real GPR image-2. (a)-(f) 80% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

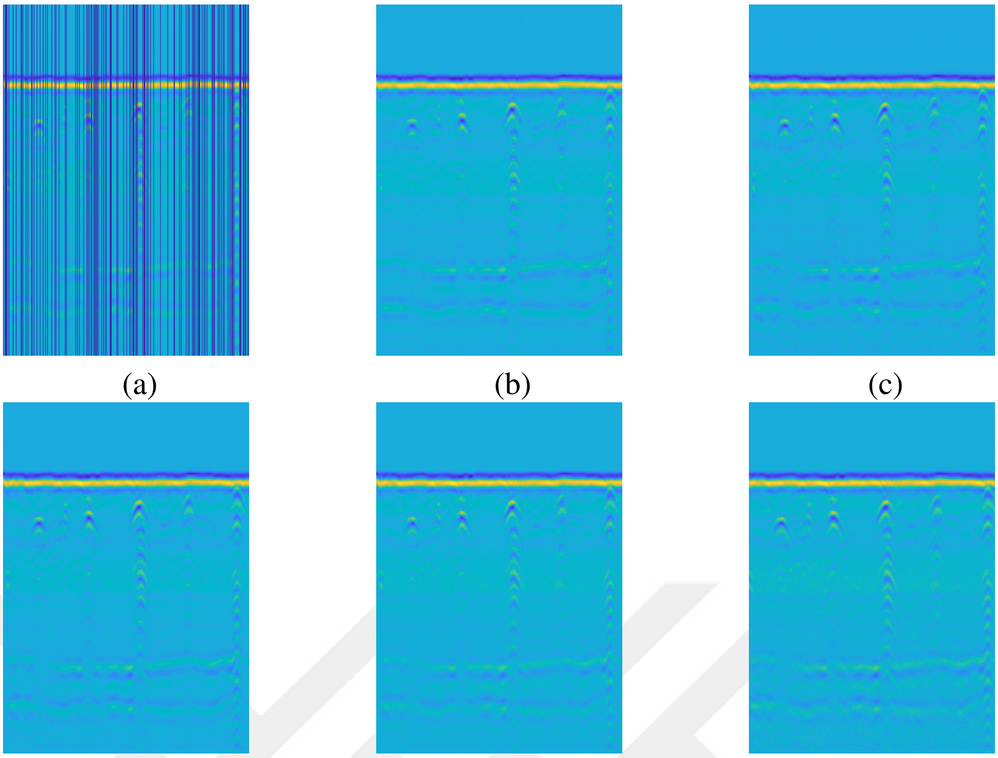
Table 4.8 : Object Detection COCO Metric Results.

	AP	AP-Hyperbola	AP50	AP75	APl	APm	APs
Original Images	23.843	23.843	69.307	12.241	18	22.847	5.297
Recovered Images with NNM Method	18.182	18.182	52.815	3.564	-	19.182	-
Recovered Images with Our Method	**18.5204**	**18.5204**	**55.1498**	**9.6310**	**20.0**	**27.4022**	**4.1311**

Based on the results given in Table 4.8, we observe a slight difference in performance. We used COCO metrics, given in Section 4.2 for performance comparison. In Fig. 4.28c, even though the target object has an indistinct appearance, the neural network still can successfully detect the object. Also, in Fig. 4.28d the target object is positioned in a downward position contrary to ordinary. However, still, object detection could be accomplished.

4.4.4 Comparison with existing deep learning based model

Missing data recovery with deep learning-based methods is a fairly new approach and up to our knowledge, only one alternative method is presented in the literature. In [59], U-Net architecture is considered as a starting point for missing data recovery.



(a) (b) (c) (d) (e) (f)

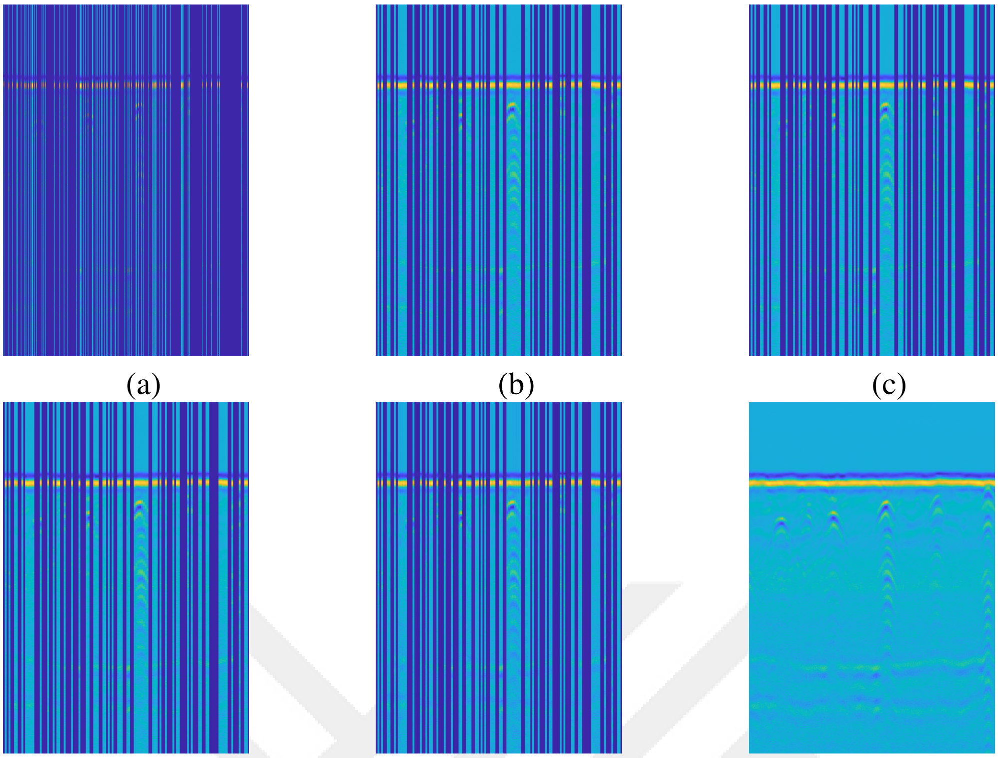
Figure 4.25 : Column-wise missing data recovery on real GPR image-2. (a)-(f) 30% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

Originally U-Net is proposed for image segmentation in biomedical images [46]. But this architecture is also commonly used for deep learning-based missing data recovery applications as well.

In the article, the authors perform column-wise missing data recovery by employing U-Net architecture and also propose a customized version of U-Net for GPR images. The authors name this customized version as Simplified U-Net (SU-Net), since the newer version has fewer convolutional layers, compared with the original U-Net architecture. And in the article, missing data recovery performances of both U-Net and SU-Net are given and compared, under different levels of column-wise missing data cases.

As a result, we intended to compare our proposed method with the approach given in [59]. And we implemented the delineated SU-Net with our GPR images and compared the two methods with the hyper-parameters given in Section 4.3.

The major drawback of the method proposed in [59] is that the model is utilized with a high volume of data-set for all of the implementations (18000 images for training, 4500



(a) (b) (c) (d) (e) (f)

Figure 4.26 : Column-wise missing data recovery on real GPR image-2. (a)-(f) 80% Masked Image, GoDec, NMC, LmaFit, NNM, DIP method results, respectively.

and 4500 images for validation and testing, respectively). However, in GPR images, creating a data-set with high volumes is mostly an ambiguous situation. Here, in order to present an impartial comparison between the two methods, we trained and tested both of the methods with the same data-set volume.

Unfortunately, SU-Net could not afford to produce a considerable result with our data-set. In a conclusion, we can not provide a sample output obtained with SU-Net. However, in order to present a comparison between SU-Net and our proposed method, we choose to examine the loss curvatures of both of the methods.

In comparison, loss curvatures are acquired by applying 50% column-wise missing data on the GPR images, as depicted in Fig. 4.3c. Consequently, in Fig. 4.29, we present loss curves that we obtained by exploiting SU-Net with 70 images for training and 30 images for testing and exploiting our proposed method with the same data-set partition.

Based on the results, we discover that our method converges in a higher value of iteration. On the other hand, our method converges to a drastically small error

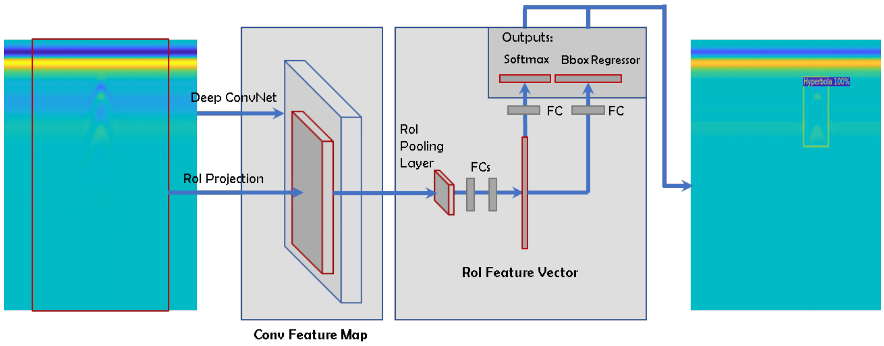
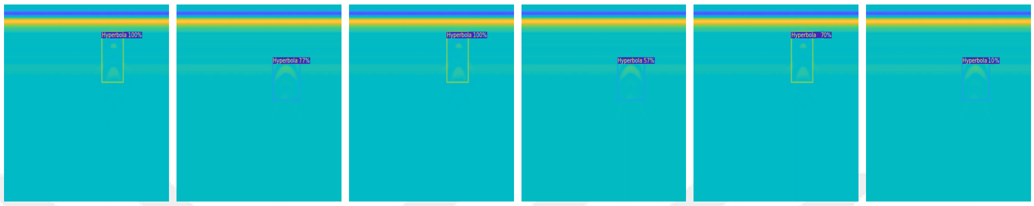


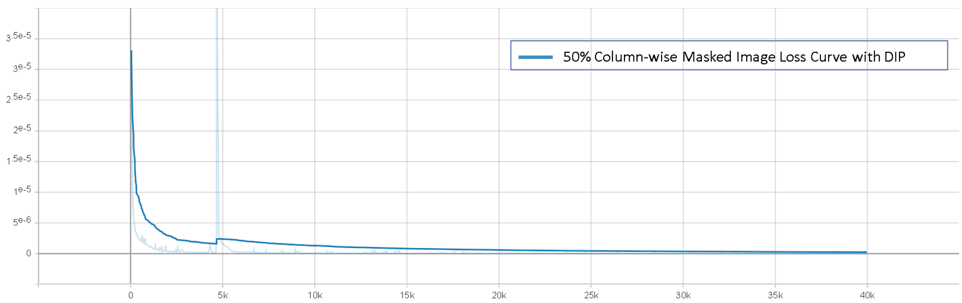
Figure 4.27 : Faster R-CNN Architecture for Object Detection



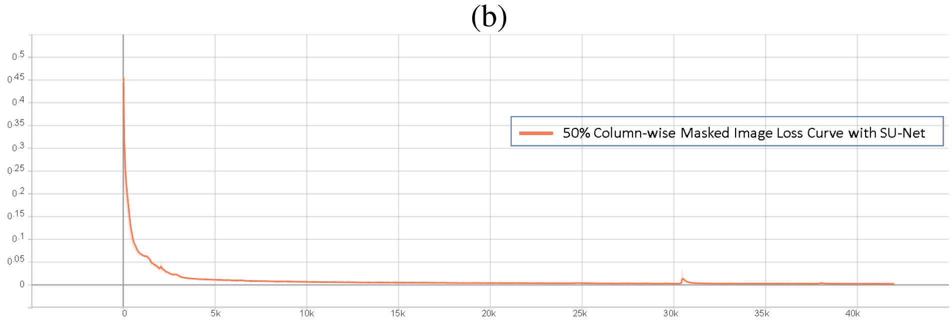
(a) (b) (c) (d) (e) (f)

Figure 4.28 : Sample recovered images with object detection results (a)-(b) Simulation GPR image (c)-(d) Proposed method. (e)-(f) NNM method.

level when compared with the loss curve of the SU-Net. And also by taking into consideration that, SU-Net fails to provide an adequate recovered image, we can conclude that our method outperforms SU-Net.



(b)



(a)

Figure 4.29 : Comparison between SU-Net and our proposed method for 50% column-wise missing data scenario (a) SU-Net (b) Proposed method

5. CONCLUSIONS

Missing data recovery on GPR data with a deep learning-based approach is the main objective that this thesis relies on. In the first part of the thesis, an extensive literature overview is presented on the missing data recovery applications, that can be considered as the conventional methods. Also, deep learning-based image inpainting methods are listed, which are exploited as missing data recovery applications. The studies in each method are listed along with their contributions and performance metrics.

In the second and third parts, we presented conventional and deep learning-based missing data recovery methods, that we utilized. We introduced principles of missing data recovery methods where GPR data is considered as a matrix and is recovered by completion of the missing values in the matrix. We benefit GoDec, LmaFit, NNM, and NMC methods as conventional methods for missing data recovery. Each method proposes a different approach to recover missing data, such as the NNM method focuses on low-rank optimization via SVD. In the LmaFit method, instead of rank minimization, the nuclear norm minimization is studied. NMC method tries to find the non-negative low-rank matrices from the input matrix. GoDec method decomposes the input matrix into the low-rank part and sparse part. Additional to the conventional missing data recovery methods, we performed data recovery on GPR images with three individual deep learning-based approaches, which are Generative Inpainting, PEN-Net, and Deep Image Prior. In Generative Inpainting, the model is constructed over a GAN architecture. PEN-Net proposes a context-encoder structure and exploits and AutoEncoder architecture. Finally, Deep Image Prior uses a SkipNet architecture and tries to grasp the details, via using skip connections towards a convolutional neural network.

In this thesis, we presented comparative results on conventional missing data recovery methods. In particular, the performance of the proposed methods is investigated under three datasets. The first and the following two datasets consisted of simulated and real GPR data, respectively. For the simulated dataset, we examined the effect of the different versions of missing data and we followed three distinctive scenarios by occluding pixels randomly as in column-wise missing data, pixel-wise missing data, and block-wise missing data on GPR data. In real datasets, we performed column-wise missing data and pixel-wise missing data scenarios. In each scenario, we increased the missing data level gradually.

We compared the performances of the proposed conventional and the deep learning-based recovery methods in quantitative and visual aspects. As quantitatively, we presented SSIM and PSNR values, that are obtained from the recovered images with each method. Based on the increase in PSNR values, it can be said that learning image semantics provides the highest support in case of the missing data ratio is too high for the recovery when compared with an approach without learning. This accomplishment is obtained by memorizing the image semantics and transferring both high-level textures and low-level features into the missing regions on the image. Also among the conventional methods, LmaFit and NNM methods are more suitable for missing data recovery on GPR images, when we compared LmaFit and NNM methods on their own merit.

Also based on the visual results that we obtained from each method, it can be seen that deep learning-based approaches provide the highest contribution specifically where the data loss is at critical levels. When the missing data level is less than 50%, conventional methods also can recover the missing data. The performance of the conventional methods decreases throughout the level of the missing data increases. On the other hand, the decrease in the performance of the deep learning-based methods is less than the conventional methods.

One contradicting observation towards deep learning-based methods is that the Generative Inpainting method performed poorly recovering missing data on GPR data, whereas PEN-Net and Deep Image Prior methods performed better. We observed that for each of the datasets, the Deep Image Prior method outperformed all of the proposed methods, in each case. In order to perform a fair comparison, we kept the hyper-parameters identical for each method. Hence Generative Inpainting method could not provide adequate results in the designated iteration limit, whereas the iteration limit was sufficient for PEN-Net and Deep Image Prior models. With Generative Inpainting, adequate results may be obtained however this requires a longer training interval which makes the comparison turned against Generative Inpainting. The necessity for the immense amount of time for the training makes Generative Inpainting an unpractical application for missing data recovery on GPR data.

In overall, with two out of three successfully adapted deep learning-based missing data recovery methods, it can be said that we proposed solution for different situations. In PEN-Net method, the pipeline can be deployed and testing of new data can be handled in every environment. However, in cases such as data acquisition is limited to very few samples, then DIP method can be exploited on a work station with enough computational power.

As future work, recovered GPR data can be used for post-processing applications, such as clutter removal or object detection. For simulated data, object detection performances are given in Chapter 4 and based on the results it can be said that our proposed method provides advantage for post-processes on GPR data. When a corrupted GPR data is received from the measurements, data can be recovered with our proposed method and post-processes give promising results. Hence our proposed method can be utilized for different post-processes as well.

Additionally in this thesis we applied deep learning-based missing data recovery methods on artificially corrupted data. But as future work, methods that we have proposed can be used on a real corrupted measurement.

Also the block-wise masking can be examined on the severely corrupted cases. As given in the Chapter 4, with the proposed method missing GPR data can be recovered under the condition of ten column-width portion of removed data. On the other hand, depending on the width of the column a limit value for the width of the missing data portion is required to be find, where the recovery can not be provided any longer.

REFERENCES

[1] **Carrick Utsi, E.**, (2017). Chapter 1 - Fundamentals of GPR Operation, **E. Carrick Utsi**, editor, Ground Penetrating Radar, Butterworth-Heinemann, pp.1–11, `https://www.sciencedirect.com/science/article/pii/B9780081022160000011`.

[2] **Zeng, Y.**, **Fu, J.**, **Chao, H. and Guo, B.** (2019). Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting, pp.1486–1494.

[3] **Nguyen, L.T.**, **Kim, J. and Shim, B.** (2019). Low-Rank Matrix Completion: A Contemporary Survey, *IEEE Access*, *7*, 94215–94237.

[4] **Candès, E. and Recht, B.** (2012). Exact Matrix Completion via Convex Optimization, *Commun. ACM*, *55*(6), 111–119, `https://doi.org/10.1145/2184319.2184343`.

[5] **Keshavan, R.**, **Montanari, A. and Oh, S.** (2010). Matrix Completion From a Few Entries, *Information Theory, IEEE Transactions on*, *56*, 2980 – 2998.

[6] **Jain, P.**, **Netrapalli, P. and Sanghavi, S.** (2013). Low-Rank Matrix Completion Using Alternating Minimization, *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, Association for Computing Machinery, New York, NY, USA, p.665–674, `https://doi.org/10.1145/2488608.2488693`.

[7] **Toh, K.C. and Yun, S.** (2010). An Accelerated Proximal Gradient Algorithm for Nuclear Norm Regularized Least Squares Problems, *Pacific Journal of Optimization*, *6*.

[8] **Yang, J. and Yuan, X.** (2011). Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, *Mathematics of Computation*, *82*.

[9] **Goldfarb, D. and Ma, S.** (2009). Convergence of Fixed-Point Continuation Algorithms for Matrix Rank Minimization, *Computing Research Repository - CORR*, *11*.

[10] **Xu, Y.**, **Yin, W.**, **Wen, Z. and Zhang, Y.** (2011). An Alternating Direction Algorithm for Matrix Completion with Nonnegative Factors, *Computing Research Repository - CORR*, *7*.

[11] **Zhou, T. and Tao, D.** (2011). GoDec: Randomized Lowrank Sparse Matrix Decomposition in Noisy Case., *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 33–40.

[12] **Wen, Z.**, **Yin, W. and Zhang, Y.** (2012). Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, *Mathematical Programming Computation*, *4*.

[13] **Kang, Z.**, **Peng, C. and Cheng, Q.** (2016). Top-N Recommender System via Matrix Completion, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, AAAI Press, p.179–184.

[14] **Fan, J. and Cheng, J.** (2018). Matrix completion by deep matrix factorization, *Neural Networks*, *98*, 34–41, `https://www.sciencedirect.com/science/article/pii/S0893608017302502`.

[15] **Efros, A. and Leung, T.** (1999). Texture synthesis by non-parametric sampling, *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pp.1033–1038 vol.2.

[16] **Criminisi, A.**, **Perez, P. and Toyama, K.** (2004). Region filling and object removal by exemplar-based image inpainting, *IEEE Transactions on Image Processing*, *13*(9), 1200–1212.

[17] **Barnes, C.**, **Shechtman, E.**, **Finkelstein, A. and Goldman, D.B.** (2009). PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing, *ACM Trans. Graph.*, *28*(3), `https://doi.org/10.1145/1531326.1531330`.

[18] **Ružić, T. and Pižurica, A.** (2015). Context-aware patch-based image inpainting using Markov random field modeling, *IEEE Transactions on Image Processing*, *24*(1), 444–456.

[19] **Pathak, D.**, **Krahenbuhl, P.**, **Donahue, J.**, **Darrell, T. and Efros, A.** (2016). Context Encoders: Feature Learning by Inpainting, pp.2536–2544.

[20] **Telea, A.** (2004). An Image Inpainting Technique Based on the Fast Marching Method, *Journal of Graphics Tools*, *9*(1), 23–34, `https://doi.org/10.1080/10867651.2004.10487596`, `https://doi.org/10.1080/10867651.2004.10487596`.

[21] **Bertalmio, M.**, **Sapiro, G.**, **Caselles, V. and Ballester, C.** (2000). Image Inpainting, *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., USA, p.417–424, `https://doi.org/10.1145/344779.344972`.

[22] **Sun, J.**, **Yuan, L.**, **Jia, J. and Shum, H.Y.** (2005). Image Completion with Structure Propagation, *ACM Trans. Graph.*, *24*(3), 861–868, `https://doi.org/10.1145/1073204.1073274`.

[23] **Wexler, Y.**, **Shechtman, E. and Irani, M.** (2007). Space-Time Completion of Video, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(3), 463–476.

[24] **Keshavan, R.H.**, **Montanari, A. and Oh, S.** (2010). Matrix Completion from Noisy Entries, *J. Mach. Learn. Res.*, *11*, 2057–2078.

[25] **Shabat, G. and Averbuch, A.** (2012). Interest Zone Matrix Approximation, *ELA. The Electronic Journal of Linear Algebra [electronic only]*, *23*.

[26] **Wang, J.**, **Ma, J.**, **Han, B. and Cheng, Y.** (2015). Seismic data reconstruction via weighted nuclear-norm minimization, *Inverse Problems in Science and Engineering*, *23*(2), 277–291, `https://doi.org/10.1080/17415977.2014.890616`, `https://doi.org/10.1080/17415977.2014.890616`.

[27] **Yang, Y.**, **Ma, J. and Osher, S.** (2013). Seismic data reconstruction via matrix completion, *Inverse Problems & Imaging*, *7*(4), 1379.

[28] **Chen Xue**, (2016), The Application of "OPTSPACE" Algorithm and Comparison with "LmaFit" Algorithm in Three-Dimensional Seismic Data Reconstruction via Low-Rank Matrix Completion.

[29] **Kumlu, D. and Erer, I.** (2020). Ground penetrating radar clutter removal via randomized low rank and sparse decomposition for missing data case, *International Journal of Remote Sensing*, *41*(19), 7680–7699.

[30] **Yang, C.**, **Lu, X.**, **Lin, Z.**, **Shechtman, E.**, **Wang, O. and Li, H.** (2017). High-Resolution Image Inpainting Using Multi-scale Neural Patch Synthesis, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4076–4084.

[31] **Iizuka, S.**, **Simo-Serra, E. and Ishikawa, H.** (2017). Globally and Locally Consistent Image Completion, *ACM Trans. Graph.*, *36*(4), `https://doi.org/10.1145/3072959.3073659`.

[32] **Ulyanov, D.**, **Vedaldi, A. and Lempitsky, V.** (2020). Deep Image Prior, *International Journal of Computer Vision*, *128*(7), 1867–1888, `http://dx.doi.org/10.1007/s11263-020-01303-4`.

[33] **Liu, G.**, **Reda, F.A.**, **Shih, K.J.**, **Wang, T.C.**, **Tao, A. and Catanzaro, B.**, (2018), Image Inpainting for Irregular Holes Using Partial Convolutions, `1804.07723`.

[34] **Yu, J.**, **Lin, Z.**, **Yang, J.**, **Shen, X. and Lu, X.** (2018). Generative Image Inpainting with Contextual Attention, pp.5505–5514.

[35] **Li, H. and Huang, J.** (2019). Localization of deep inpainting using high-pass fully convolutional network, *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*(Iccv), 8300–8309.

[36] **Liao, L.**, **Hu, R.**, **Xiao, J. and Wang, Z.** (2019). Artist-Net: Decorating the Inferred Content with Unified Style for Image Inpainting, *IEEE Access*, *7*, 36921–36933.

[37] **Dosovitskiy, A. and Brox, T.** (2016). Generating Images with Perceptual Similarity Metrics based on Deep Networks, *arXiv:1602.02644 [cs.LG]*, `1602.02644`.

[38] **Zhang, S., He, R. and Tan, T.** (2016). DeMeshNet: Blind Face Inpainting for Deep MeshFace Verification, *arXiv:1611.05271 [cs.CV]*, `1611.05271`.

[39] **Li, Y., Liu, S., Yang, J. and Yang, M.H.** (2017). Generative Face Completion, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[40] **Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H. and Kuo, C.C.J.** (2018). Contextual-Based Image Inpainting: Infer, Match, and Translate, 3–18.

[41] **Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. and Huang, T.S.** (2018). Generative Image Inpainting With Contextual Attention, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[42] **Yan, Z., Li, X., Li, M., Zuo, W. and Shan, S.** (2018). Shift-Net: Image Inpainting via Deep Feature Rearrangement, *Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision.*

[43] **Papyan, V., Romano, Y., Elad, M. and Sulam, J.** (2017). Convolutional Dictionary Learning via Local Processing, pp.5306–5314.

[44] **Wang, X., Yu, F., Dou, Z.Y., Darrell, T. and Gonzalez, J.E.** (2018). SkipNet: Learning Dynamic Routing in Convolutional Networks, *Proceedings of the European Conference on Computer Vision (ECCV).*

[45] **He, K., Zhang, X., Ren, S. and Sun, J.** (2016). Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770–778.

[46] **Weng, W. and Zhu, X.** (2015). INet: Convolutional Networks for Biomedical Image Segmentation, *IEEE Access*, *9*, 16591–16603, `1505.04597`.

[47] **Hashimoto, F., Ohba, H., Ote, K., Kakimoto, A., Tsukada, H. and Ouchi, Y.** (2021). 4D deep image prior: dynamic PET image denoising using an unsupervised four-dimensional branch convolutional neural network, *Physics in Medicine & Biology*, *66*(1), 015006, `https://doi.org/10.1088/1361-6560/abcd1a`.

[48] **Eckart, C. and Young, G.** (1936). The approximation of one matrix by another of lower rank, *Psychometrika*, *1*, 211–218.

[49] **Xu, Y., Yin, W., Wen, Z. and Zhang, Y.** (2012). An alternating direction algorithm for matrix completion with nonnegative factors, *Frontiers of Mathematics in China*, *7*(2), 365–384, `http://dx.doi.org/10.1007/s11464-012-0194-5`.

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

[51] **Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A.** (2018). Image-to-Image Translation with Conditional Adversarial Networks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, `1611.07004`.

[52] **Goodfellow, I.J.**, **Pouget-Abadie, J.**, **Mirza, M.**, **Xu, B.**, **Warde-Farley, D.**, **Ozair, S.**, **Courville, A. and Bengio, Y.** (2014). Generative Adversarial Nets, *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, MIT Press, Cambridge, MA, USA, p.2672–2680.

[53] **Burger, H.C.**, **Schuler, C.J. and Harmeling, S.** (2012). Image denoising: Can plain neural networks compete with BM3D?, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp.2392–2399.

[54] **Ren, J.S.**, **Xu, L.**, **Yan, Q. and Sun, W.** (2015). Shepard Convolutional Neural Networks, *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, MIT Press, Cambridge, MA, USA, p.901–909.

[55] **Warren, C.**, **Giannopoulos, A. and Giannakis, I.** (2016). gprMax: Open source software to simulate electromagnetic wave propagation for Ground Penetrating Radar, *Computer Physics Communications*, *209*.

[56] **Koprucu, F.**, **Erer, I. and Kumlu, D.** (2021). Clutter Aware Deep Detection For Subsurface Radar Targets, *International Geoscience and Remote Sensing Symposium*, 3–6.

[57] **Ren, S.**, **He, K.**, **Girshick, R. and Sun, J.** (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, `1506.01497`.

[58] **Pham, M.T. and Lefèvre, S.** (2018). Buried Object Detection from B-Scan Ground Penetrating Radar Data Using Faster-RCNN, pp.6804–6807.

[59] **Rohman, B.P.A.**, **Member, S. and Nishimoto, M.** (2021). Reconstruction of Missing Ground-Penetrating Radar Traces Using Simplified U-Net, *IEEE Geoscience and Remote Sensing Letters*, 1–5.

CURRICULUM VITAE

Personal Information

NAME SURNAME | Kübra Taş

Experiences

2014-2019 | **Computer Vision Process Engineer**
Robert Bosch GmbH
Bursa, Turkey

2019-2020 | **Computer Vision Process Expert Engineer**
Robert Bosch GmbH
Bursa, Turkey

2020- | **Artificial Intelligence Engineer**
Tekhnelogos A.Ş.
Istanbul, Turkey

Education

2011-2016 | **Bachelor of Science**
Electrical Engineering
Istanbul Technical University, Istanbul, Turkey
Supervisor: Murat Yılmaz

Publications From This Thesis

CONFERENCE | K. Tas, D. Kumlu, I.Erer, "Pyramid-Context Encoder Network (PEN-Net) for Missing Data Recovery in Subsurface Images," *accepted*

JOURNAL | K. Tas, D. Kumlu, I.Erer, "Missing Data Recovery in GPR via Deep Image Prior," *in progress*