Uncertainty-Aware Convolutional Neural Networks for Vision Tasks on Sparse Data

Abdelrahman Eldesokey
Uncertainty-Aware Convolutional Neural Networks for Vision Tasks on Sparse Data

Abdelrahman Eldesokey

Linköping University
Department of Electrical Engineering
Computer Vision Laboratory
SE-581 83 Linköping, Sweden

Linköping 2021
Tidiga datorseendealgoritmer arbetade med täta 2D-bilder som spelats in i gråskala eller med färgkameror. Dessa är passiva bildsensorer som under gynnsamma ljusförhållanden ger en begränsad scenrepresentation baserad endast på ljusflöde. Dessa begränsningar hämmade utvecklingen av de många datorseendealgoritmer som kräver information om scenens struktur under varierande ljusförhållanden. Utvecklingen av aktiva sensorer såsom kameror baserade på Time-of-Flight (ToF) bidrog till att lindra dessa begränsningar. Dessa gav emellertid istället upphov till många nya utmaningar, såsom bearbetning av gles data komma av flervägsinterferens samt ocklusion.

Man har försökt tackla dessa utmaningar genom att förbättra insamlingsprocessen i TOF-kameror eller genom att efterbearbeta deras data. Tidigare föreslagna metoder har dock varit sensor- eller till och med modellspecifika där man måste ställa in varje enskild sensor. Ett attraktivt alternativ är inlärningsbaserade metoder där man istället lär sig förhållandet mellan sensordata och en förbättrad version av ditto. Ett kraftfullt exempel på inlärningsbaserade metoder är neurala faltningssnät (CNNs). Dessa har varit extremt framgångsrika inom datorseende, men förutsätter tyvärr tät data och kan därför inte på ett effektivt sätt bearbeta ToF-sensorernas glesa data.

I denna avhandling föreslår vi en ny variant av faltningssnät som vi kallar normaliserade faltningssnät (eng. Normalized Convolutional Neural Networks) och som direkt kan arbeta med gles data. Först skapar vi ett deriverbart faltningssnät baserat på normaliserad faltning som tar in gles data samt en konfidenskarta. Konfidenskartan innehåller information om vilka pixlar vi har mätningar för och vilka som saknar mätningar. Modulen interpolerar sedan pixlar som saknar mätningar baserat på närliggande pixlar för vilka mätningar finns. Därefter föreslår vi ett kriterie för att propagera konfidens vilket tillåter oss att bygga en kaskad av normaliserade faltningssnät baserade på faltningssnät (CNs). Dessa har varit extremt framgångsrika inom datorseende, men förutsätter tyvärr tät data och kan därför inte på ett effektivt sätt bearbeta ToF-sensorernas glesa data.

Som ett andra bidrag undersökte vi sammanslagningsmetoden av normaliserade faltningssnät med konventionella faltningssnät som arbetar med vanliga färgbilder. Vi undersöker olika sätt att slå samman nätet och ger en grundlig analys för de olika nätverksdelarna. Den bästa sammanslagningsmetoden uppnår state-of-the-art-prestanda på scendjupkompletteringsproblemet med färgbilder och uppnådde ett mycket lågt nätverk.

Som ett tredje bidrag försöker vi sammanslagningsmetoden av normaliserade faltningssnät med konventionella faltningssnät som arbetar med vanliga färgbilder. Vi undersöker olika sätt att slå samman nätet och ger en grundlig analys för de olika nätverksdelarna. Den bästa sammanslagningsmetoden uppnår state-of-the-art-prestanda på scendjupkompletteringsproblemet med färgbilder och uppnådde ett mycket lågt nätverk.

Slutligen försöker vi använda vårt ramverk för en uppgift man ofta löser med vanliga faltningssnät, nämligen uppsampling. Vi formulerar uppsamplingproblemet som om vi fått in gles data och löser det med normaliserade faltningssnät. Jämfört med befintliga metoder är den föreslagna metoden både effektiv och lättattic. Vi testar vår uppsamplingsmetod på olika optisktflödesnät och visar att den konsekvent ger förbättrade resultat. När vi integrerar den med ett nytta faltningssnät slår vi all till och med vidare förbättrande resultat.

Slutligen försöker vi använda vårt ramverk för en uppgift man ofta löser med vanliga faltningssnät, nämligen uppsampling. We formulerar uppsamplingproblemet som om vi fått in gles data och löser det med normaliserade faltningssnät. Jämfört med befintliga metoder är den föreslagna metoden både effektiv och lättattic. Vi testar vår uppsamplingsmetod på olika optisktflödesnät och visar att den konsekvent ger förbättrade resultat. När vi integrerar den med ett nytta faltningssnät slår vi all till och med vidare förbättrande resultat.
ABSTRACT

Early computer vision algorithms operated on dense 2D images captured using conventional monocular or color sensors. Those sensors embrace a passive nature providing limited scene representations based on light reflux, and are only able to operate under adequate lighting conditions. These limitations hindered the development of many computer vision algorithms that require some knowledge of the scene structure under varying conditions. The emergence of active sensors such as Time-of-Flight (ToF) cameras contributed to mitigating these limitations; however, they gave a rise to many novel challenges, such as data sparsity that stems from multi-path interference, and occlusion.

Many approaches have been proposed to alleviate these challenges by enhancing the acquisition process of ToF cameras or by post-processing their output. Nonetheless, these approaches are sensor and model specific, requiring an individual tuning for each sensor. Alternatively, learning-based approaches, i.e., machine learning, are an attractive solution to these problems by learning a mapping from the original sensor output to a refined version of it. Convolutional Neural Networks (CNNs) are one example of powerful machine learning approaches and they have demonstrated a remarkable success on many computer vision tasks. Unfortunately, CNNs naturally operate on dense data and cannot efficiently handle sparse data from ToF sensors.

In this thesis, we propose a novel variation of CNNs denoted as the Normalized Convolutional Neural Networks that can directly handle sparse data very efficiently. First, we formulate a differentiable normalized convolution layer that takes in sparse data and a confidence map as input. The confidence map provides information about valid and missing pixels to the normalized convolution layer, where the missing values are interpolated from their valid vicinity. Afterwards, we propose a confidence propagation criterion that allows building cascades of normalized convolution layers similar to the standard CNNs. We evaluated our approach on the task of unguided scene depth completion and achieved state-of-the-art results using an exceptionally small network.

As a second contribution, we investigate the fusion of a normalized convolution network with standard CNNs employing RGB images. We study different fusion schemes, and we provide a thorough analysis for different components of the network. By employing our best fusion strategy, we achieve state-of-the-art results on guided depth completion using a remarkably small network.

Thirdly, to provide a statistical interpretation for confidences, we derive a probabilistic framework for the normalized convolutional neural networks. This framework estimates the input confidence in a self-supervised manner and propagates it to provide a statistically valid output confidence. When compared against existing approaches for uncertainty estimation in CNNs such as Bayesian Deep Learning, our probabilistic framework provides a higher quality measure of uncertainty at a significantly lower computational cost.

Finally, we attempt to employ our framework in a common task in CNNs, namely upsampling. We formulate the upsampling problem as a sparse problem, and we employ the normalized convolutional neural networks to solve it. In comparison to existing approaches, our proposed upsampler is structure-aware while being light-weight. We test our upsampler with various optical flow estimation networks, and we show that it consistently improves the results. When integrated with a recent optical flow network, it sets a new state-of-the-art on the most challenging optical flow dataset.
ACKNOWLEDGMENTS

My goal from pursuing a PhD was mainly to gain autonomy in conducting research, and solving problems. I can undoubtedly say that this was not going to be possible without the help and support from my colleagues at the Computer Vision Laboratory (CVL). First, I would like to genuinely thank my supervisor Michael Felsberg for his continuous support and guidance throughout this journey. His unique supervision allowed me to make my own assessments for every research problem I encountered, and unrestrictedly attack them. I would also like to thank my co-supervisor Fahad Shahbaz Khan for his mentoring, especially on writing manuscripts. The thank is extended to all of my colleagues at CVL, Mikael Persson for the interesting discussions, Karl Holmquist for the fruitful collaboration, Felix Järemo-Lawin for the spontaneous discussions when we shared an office, Andreas Robinson for the exciting collaboration, and to all other colleagues who benefited me through any kind of interaction. Many thanks to Fahad, Mikael, Andreas, Felix, Karl, and Gustav for helping me revise this manuscript; and to Joakim Johander for correcting my poor translation of the Swedish Abstract.

I am also thankful to my parents for the emotional support; my father who planted the love of science in me and always encouraged me to pursue my PhD, and my mother for boosting my confidence and valuing my self-worth. Last but not least, I am grateful to my beloved life-companion May Abdellatif for taking this journey with me, being my friend/mentor/partner, and providing me with all kind of support that I needed. And not to forget, the apple of my eyes, my daughters Joury and Laila who enlightened my life, especially during the dark days of the Swedish winter (and autumn).

Finally, I would like to thank the Swedish Research Council for supporting this work through grant 2018-04673, and the Wallenberg AI, Autonomous Systems and Software Program (WASP) for the educational support.

Abdelrahman Eldesokey
Linköping, May 2021
About the Cover The cover mimics the famous record cover "Abbey Road" by The Beatles. It shows a conceptualization for a LiDAR point cloud that is processed by methods developed in this thesis. From left to right: a raw point cloud with overlapping points at the lower part, a densified version using our approach in Paper I with distortions at the legs due to using binary input confidences, a refined input using our approach in Paper III with learned confidences, and finally a densified version with the learned confidences. The idea of the cover is mine, and Andreas Robinson suggested to do it on the The Beatles poster (which is definitely cooler).
Contents

Abstract iii
Acknowledgments vi
Contents vii

I Background 1
1 Introduction 3
  1.1 Sparse Data in Computer Vision ......................... 3
  1.2 Contributions ........................................... 6
  1.3 Thesis Outline ......................................... 8
  1.4 Included Publications ................................ 9
  1.5 Additional Publications ............................... 13

2 From Signal Spaces to Normalized Convolution 15
  2.1 Signals as Functions ............................... 16
  2.2 Scalar Products and Norms ............................ 17
  2.3 Signal Space Representation ......................... 19
  2.4 Normalized Convolution ............................ 20
  2.5 Normalized Averaging ............................... 21
  2.6 Confidence Propagation .............................. 24

3 Deep Guided Filtering for Vision Tasks 27
  3.1 Disturbances in LiDAR-Camera Setups .................. 28
  3.2 The Correspondence Problem .......................... 30
  3.3 Deep Bilateral Filtering ............................. 31
  3.4 Adaptive CNN Filters ............................... 32
  3.5 Pixel-Adaptive Convolution ......................... 33
4 Uncertainty in Neural Networks from a Bayesian Perspective 35
  4.1 Uncertainty in Machine Learning .......................... 35
  4.2 Bayesian Inference ........................................... 37
  4.3 Approximate Variational Inference ......................... 39
  4.4 Bayesian Neural Networks ................................... 40

5 Upsampling in CNNs 43
  5.1 Classical Interpolation Methods ............................ 44
  5.2 Non-Parametric Upsampling ................................ 46
  5.3 Parametric Upsampling ....................................... 46
  5.4 Hybrid Upsampling Approaches .............................. 47

6 Concluding Remarks 49

Bibliography 51

II Publications 59

Paper I 63
Paper II 77
Paper III 93
Paper IV 105
Part I

Background
Chapter 1

Introduction

1.1 Sparse Data in Computer Vision

Computer vision is a multidisciplinary domain, which attempts to mimic the visual system in humans and animals. The biological eye is typically superseded by cameras or other types of visual sensors, while computers act as the brain. The emergence of silicon-based image sensors, such as CCD and CMOS cameras [60], around 1970 revolutionized computer vision as they facilitated capturing images. These cameras classify as passive sensors comprising 2D arrays of light-sensitive receptors capturing the reflected light and producing dense 2D images on a regular grid. As a result, the majority of early computer vision algorithms were designed to operate on dense images captured by passive sensors, e.g., features extraction [22, 23], optical flow [46, 26], and stereo matching [1, 36]. Unfortunately, the limited scene representation cues from passive sensors hindered the development of these algorithms. Furthermore, passive sensors can only operate properly under good lighting conditions.

The aforementioned drawback of passive sensors can be alleviated using active sensors, which can produce richer information such as scene depth, and reflectively to reinforce the scene understanding. In addition, they can operate efficiently even under poor lighting conditions due to their active nature. Figure 1.1 shows an illustration for how active sensors operate in comparison to passive sensors. An example for active sensors are the time-of-flight (ToF) cameras, e.g., Microsoft Kinect 1, and LiDARs. They essentially emit light-modulated beams towards the scene, receive the reflected beams, and process them to produce depth maps based on phase differences [21]. Due to their active nature, they can provide accurate scene depth information that is prominently valuable for tasks that require scene-awareness such as autonomous driving and robotics. However, ToF cameras emerged with their own challenges such as noisy and missing measurements (sparsity), irregularity.

1https://developer.microsoft.com/en-us/windows/kinect/
1. Introduction

![Figure 1.1: An illustration for how passive and active cameras operate. Passive cameras reflect incident light from an external light source, while active cameras concurrently emit modulated light beams, and receive the reflected ones.](image)

of the signal grid, and other challenges that stems from the dynamics of the scene such multi-path interference [13], object boundary ambiguity [53], occlusion, and motion blur [21].

Multiple model-based solutions has been proposed in the literature to address these challenges on the sensor level [21], i.e., enhancing the acquisition and the post-processing pipelines. Nonetheless, these solutions are sensor-specific and need to be customized for each sensor, and application. Another attractive category of solutions for addressing these challenges is the data-driven approaches, i.e., machine learning and ultimately deep learning. Essentially, if a sufficient amount of data captured by a ToF sensor is available, it is possible to learn a mapping to a cleaner version of the data (groundtruth) without any prior knowledge about the sensor. Among these data-driven approaches, Convolutional Neural Networks (CNNs) have demonstrated remarkable success on many computer vision task utilizing dense images such as, object classification [40, 25], object detection [17, 43], optical flow estimation [6, 59], and depth estimation [7]. Unfortunately, data from active sensors such as ToF cameras suffer from sparsity and uncertainty limiting their usability within CNNs.

The fundamental operation in CNNs is the spatial convolution that is defined for dense signals. In a situation where parts of the signal are missing, the outcome from convolution becomes invalid. As an example, assume a grayscale image that is sparsified by removing a number of pixels randomly. When convolving this sparse image, there is no way to inform the convolution operator that the missing pixels are invalid. Figure 1.2 illustrates this scenario, where a low-pass filter is used to convolve a grayscale image and its sparsified variation. The output for the sparse image is clearly corrupted as shown in Figure 1.2e since the missing pixels were treated as valid zero grayscale pixels attenuating their neighboring pixels. It is worth noting how
The grayscale levels of the convolved sparse image are severely reduced because of the missing pixels in comparison with the convolved dense image in Figure 1.2d. This problem is even more prominent in CNNs with cascades of convolutional layers, making the learning from sparse data notably challenging. Consequently, huge CNNs are needed to learn directly from sparse data, imposing computational overhead [47, 64, 33].

An appealing solution to the sparsity and uncertainty problems is the normalized convolution operator proposed by Knutsson and Westin [38]. The key idea is to accompany the signal with a confidence map of the same dimensionality to indicate which parts of the signal are valid/reliable. During convolution, only valid parts of the signal are included in the computations, while the missing/noisy parts are interpolated from their neighbors. The simplest form for these confidence maps are binary masks, which have ones where data is present and zeros otherwise. Figure 1.2f shows the previous example when using the normalized convolution instead of the standard convolution. The figure clearly demonstrates how normalized convolution produces undistorted output that is almost identical to the convolved dense image in Figure 1.2d. Moreover, the missing values are interpolated and the grayscale levels are maintained in comparison to the standard convolution.
1. Introduction

In this thesis, we revisit the classical normalized convolution operator and investigate how to effectively integrate its sparse data handling capabilities into standard CNNs. In contrast to the standard convolution layer, a normalized convolution layer is expected to efficiently handle missing/uncertain values by discarding them from computations, and inferring them from their vicinity. This eliminates the need for huge CNNs to learn how to discriminate between valid and missing data points. Furthermore, incorporating confidences in CNNs contribute to their reliability and interpretability. This is especially important for safety-critical applications such as, robotics, autonomous driving and surveillance.

1.2 Contributions

The main focus of this thesis is introducing a novel sparsity and uncertainty-aware convolution operator that can be incorporated in existing CNNs. As explained earlier, the normalized convolution framework possesses appealing properties for handling sparse data as it employs confidences as a measure of uncertainty. Therefore, we study the classical normalized convolution in [38], and we develop a novel CNN-compatible counterpart. We first introduce a normalized convolution layer (NConv) that operates on confidence-accompanied signals in Paper I. This layer receives the sparse input, a confidence mask, and it employs a naïve basis for efficiency and differentiability. We assume binary input confidence masks indicating where data points are present or missing. Each normalized convolution layer outputs a denser version of the sparse input that is convolved with a trainable filter. Moreover, input confidences are propagated to the next layer through an efficient criterion that we derive. This allows constructing cascades of normalized convolution layers that can directly accept sparse data along with their confidences, and produce a denser version with an output confidence map indicating how reliable each location in the output is.

To evaluate our proposed layer on a real problem, we apply it to the problem of unguided depth completion. In this problem, depth point clouds captured by a LiDAR sensor are projected into a 2D image plane to facilitate fusion with monocular images. However, this process introduces several challenges such as sparsity, irregular grids, and false projections due to camera pose or occlusions. The aim of this task is to densify the projected point clouds, while rectifying faulty measurements. We develop a U-Net [44] shaped network using our proposed NConv layers for this purpose, and evaluate it on the KITTI-Depth dataset [63]. Our network achieves state-of-the-art results with only 480 parameters, which is three times fewer than any other approach in comparison. These results clearly demonstrate how normalized

\[This work was supported by the Swedish Research Council grant 2018-04673.\]
convolution can efficiently handle sparse data in comparison to the standard convolution.

Since the task of depth completion comprises densifying the sparse point cloud and rectifying the faulty points as well, we investigate achieving the latter by exploiting structural consistency from the RGB modality, i.e., guided depth completion. This task is readily feasible as the projected LiDAR point clouds lie on same plane as the RGB images. We employ this advantage in Paper II, and we investigate different fusion schemes between depth predictions from our unguided network in Paper I, and their corresponding RGB images. This includes early and late fusion schemes for a multi-stream as well as an encoder-decoder architectures. Furthermore, we show how the output confidence from our unguided network can be beneficial for improving the fusion. All networks are evaluated on the KITTI-Depth dataset as well as the NYU-Depth-v2 [49], which was captured using a Microsoft Kinect RGB-D sensor. We achieve state-of-the-art results using a significantly smaller network with at least one order of magnitude fewer parameters than any other competing CNN approach.

A major limitation of normalized convolution networks is the unavailability of a proper input confidence. In Paper I and II, the input confidence was assumed to be binary indicating that all present input points are reliable. However, it was shown in [54] that some of the input points in the KITTI-Depth datasets are erroneous and they do not match the groundtruth; therefore, the binary input confidence assumption becomes invalid. Solving this problem is challenging since input confidences have no groundtruth and can not be learned under supervision. We exploit the confidence propagation capabilities of our NConv layers, and we propose to learn the input confidence that minimizes the prediction error in a self-supervised fashion in Paper III. Experiments on the task of unguided depth completion on the KITTI-Depth datasets shows that by estimating confidences using our proposed approach, we outperform all other unguided approaches. Moreover, we show that learning input confidences leads to remarkably sharp predictions along object boundaries as it acts similar to guided filtering [24], but without requiring any guidance modality. We also test our approach on sparse optical flow rectification and ToF data and it performs remarkably well, which demonstrates its generalizability to other sparse problems.

Another crucial aspect when dealing with sparse and noisy data within CNNs, is examining how reliable the predictions are. NConv networks fulfill this requirement by providing output confidence maps, however, when these maps were assessed as a measure of uncertainty using metrics from the Bayesian Neural Networks literature [29], they performed sub-optimally. Ideally, a high-quality measure of uncertainty should be correlated with the prediction error, i.e., high uncertainty at high errors and vice versa. Hence, we establish a probabilistic framework for normalized convolutional networks in Paper III that aims to produce accurate predictions as well as reliable con-
1. Introduction

We evaluate our probabilistic variation against existing probabilistic approaches for depth completion in [20], and we outperform all of them both in terms of prediction accuracy and the quality of the output uncertainty. This contribution paves the way for implementing a fully Bayesian variation based on the output confidences.

Finally, we employ NConv layers in a widely used task in CNNs, upsampling. In Paper IV, we formulate the problem of upsampling as a sparse problem and we design an NConv network to solve it. Our proposed upsampler is light-weight and can use guidance from RGB images or intermediate deep features to learn an adaptive upsampling that respects object boundaries. We test the upsampler within a state-of-the-art optical flow estimation network [61] to upsample the final coarse prediction by a factor of eight. This modification sets a new state-of-the-art on the Sintel dataset [3], while reducing the number of network parameters by 7.5%. We also test our upsampler with popular optical flow networks such as FlowNetS [6], PWCNet [59], and it boosts the results up to 15% compared to the bilinear interpolation.

1.3 Thesis Outline

This thesis comprises two parts: background, and publications. The first part includes a number of chapters explaining relevant theory that facilities the understanding of the subject and the corresponding publications. These chapters are organized as follows:

- **Chapter 1 - From Signal Spaces to Normalized Convolution:** derives the classical normalized convolution formulation starting from signal spaces representation, and explains different confidence propagation criteria in the literature.

- **Chapter 2 - Deep Guided Filtering for Vision Tasks:** describes some sources for disturbances in common vision tasks, and several deep guided filtering approaches to address them.

- **Chapter 3 - Uncertainty in Neural Networks from a Bayesian Perspective:** provides an overview on modelling uncertainty in machine learning from a Bayesian perspective with a reflection on neural networks.

- **Chapter 4 - Upsampling in CNNs:** describes common approaches for upsampling in CNNs, their strengths, and weaknesses.

- **Chapter 5 - Concluding Remarks:** summarizes the contributions of this thesis.

The second part includes the publications that led to this thesis.
1.4 Included Publications

Paper I: Propagating Confidences through CNNs for Sparse Data Regression

Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. “Propagating Confidences through CNNs for Sparse Data Regression”. In: The British Machine Vision Conference (BMVC), Northumbria University, Newcastle upon Tyne, England, UK, 2018.

Abstract: In most computer vision applications, convolutional neural networks (CNNs) operate on dense image data generated by ordinary cameras. Designing CNNs for sparse and irregularly spaced input data is still an open problem with numerous applications in autonomous driving, robotics, and surveillance. To tackle this challenging problem, we introduce an algebraically-constrained convolution layer for CNNs with sparse input and demonstrate its capabilities for the scene depth completion task. We propose novel strategies for determining the confidence from the convolution operation and propagating it to consecutive layers. Furthermore, we propose an objective function that simultaneously minimizes the data error while maximizing the output confidence. Comprehensive experiments are performed on the KITTI depth benchmark and the results clearly demonstrate that the proposed approach achieves superior performance while requiring three times fewer parameters than the state-of-the-art methods. Moreover, our approach produces a continuous pixel-wise confidence map enabling information fusion, state inference, and decision support.

Author’s Contribution: The main ideas was initiated by Michael Felsberg through the VR project 2018-04673. The author developed the method, performed the experiments, and was responsible for writing the manuscript in collaboration with Fahad Shahbaz Khan.
1. Introduction

Paper II: Confidence Propagation through CNNs for Guided Sparse Depth Regression

Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. “Confidence Propagation through CNNs for Guided Sparse Depth Regression”. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019.

Abstract: In this paper, we propose an algebraically-constrained normalized convolution layer for CNNs with highly sparse input that has a smaller number of network parameters compared to related work. We propose novel strategies for determining the confidence from the convolution operation and propagating it to consecutive layers. We also propose an objective function that simultaneously minimizes the data error while maximizing the output confidence. To integrate structural information, we also investigate fusion strategies to combine depth and RGB information in our normalized convolution network framework. In addition, we introduce the use of output confidence as an auxiliary information to improve the results. The capabilities of our normalized convolution network framework are demonstrated for the problem of scene depth completion. Comprehensive experiments are performed on the KITTI-Depth and the NYU-Depth-v2 datasets. The results clearly demonstrate that the proposed approach achieves superior performance while requiring only about 1-5% of the number of parameters compared to the state-of-the-art methods.

Author’s Contribution: The idea originated from discussions with Michael Felsberg. The author developed the method, performed the experiments, and was responsible for writing the manuscript in collaboration with the coauthors.
Paper III: Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End

Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Mikael Persson. “Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 12014–12023.

Abstract: We propose a novel approach to identify disturbed measurements in the input by learning an input confidence estimator in a self-supervised manner based on the normalized convolutional neural networks (NCNNs). Further, we propose a probabilistic version of NCNNs that produces a statistically meaningful uncertainty measure for the final prediction. When we evaluate our approach on the KITTI dataset for depth completion, we outperform all the existing Bayesian Deep Learning approaches in terms of prediction accuracy, quality of the uncertainty measure, and the computational efficiency. Moreover, our small network with 670k parameters performs on-par with conventional approaches with millions of parameters. These results give strong evidence that separating the network into parallel uncertain and prediction streams leads to state-of-the-art performance with accurate uncertainty estimates.

Author’s Contribution: The author initiated the idea, improved it through discussions with the coauthors, and was the main contributor to development, implementation, evaluation of the method, and writing the manuscript.
1. Introduction

Paper IV: Normalized Convolution Upsampling for Refined Optical Flow Estimation

Abdelrahman Eldesokey and Michael Felsberg. “Normalized Convolution Upsampling for Refined Optical Flow Estimation”. In: Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications: VISAPP. 2021. (Best Paper Award)

Abstract: Optical flow is a regression task where convolutional neural networks (CNNs) have led to major breakthroughs. However, this comes at major computational demands due to the use of cost-volumes and pyramidal representations. This was mitigated by producing flow predictions at quarter the resolution, which are upsampled using bilinear interpolation during test time. Consequently, fine details are usually lost and post-processing is needed to restore them. We propose the Normalized Convolution UPsampler (NCUP), an efficient joint upampling approach to produce the full-resolution flow during the training of optical flow CNNs. Our proposed approach formulates the upsampling task as a sparse problem and employs the normalized convolutional neural networks to solve it. We evaluate our upsampler against existing joint upsampling approaches when trained end-to-end with a coarse-to-fine optical flow CNN (PWCNet) and we show that it outperforms all other approaches on the FlyingChairs dataset while having at least one order fewer parameters. Moreover, we test our upsampler with a recurrent optical flow CNN (RAFT) and we achieve state-of-the-art results on Sintel benchmark with \( \sim 6\% \) error reduction. Finally, our upsampler shows better generalization capabilities than RAFT when trained and evaluated on different datasets.

Author’s Contribution: The author initiated the idea, developed the method, conducted experiments, and was the main contributor to the writing of the manuscript.
1.5 Additional Publications

Here, we list other peer-reviewed publications produced by the author during this thesis, but were not included in this manuscript due to their irrelevance to the topic.

- **Abdelrahman Eldesokey**, Michael Felsberg, and Fahad Shahbaz Khan. “Ellipse Detection for Visual Cyclists Analysis “In the Wild””. In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2017, pp. 319–331.

- Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Gutav Häger, Alan Lukezic, **Abdelrahman Eldesokey**, et al. “The visual object tracking VOT2017 challenge results”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017.

- Adam Nyberg, **Abdelrahman Eldesokey**, David Gustafsson, and David Bergström. “Unpaired Thermal to Visible Spectrum Transfer using Adversarial Training”. In: *Multimodal Learning AND Applications Workshop (MULA) - ECCV Workshops in Munich, Germany, 2018*, PP. 657–669.

- Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, **Abdelrahman Eldesokey**, et al. ”The sixth visual object tracking VOT2018 Challenge Results”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

- **Abdelrahman Eldesokey**, Mikael Persson, Michael Felsberg, and Fahad Shahbaz Khan. ”Tackling Disturbed Depth Maps by Learning Input Data Confidence”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2019. An early version for Paper III.

- Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni Kristian Kamarainen, Luka Cehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, **Abdelrahman Eldesokey**, et al. ”The seventh visual object tracking VOT2019 challenge results”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2019.
CHAPTER 2

From Signal Spaces to Normalized Convolution

The predominant operation in Convolutional Neural Networks (CNNs) is, as the name suggests, convolution. As demonstrated in section 1.1, convolution can only operate on complete signals with no missing or uncertain parts. Therefore, we are interested in replacing the standard convolution in CNNs with the normalized convolution [38], that has a machinery for handling incomplete signals.

The key concept for normalized convolution is based on the theory of confidence-equipped signals [18, 12, 66, 37]. Accompanying signals with confidence maps is beneficial to mark valid and invalid regions of the signals. Taking this into account, only valid parts are considered when processing the signal. This way, disturbances caused by missing or corrupted regions can be mitigated. Normalized convolution employs this notion by forming an image of the incomplete signal in a predefined subspace using only the valid parts of the signal as indicated by the accompanying confidence map. Eventually, the complete signal can be reconstructed from the subspace where the missing parts are implicitly induced from their vicinity.

This chapter covers some background material for Paper I, II, and it aims to provide a comprehensive derivation for the classical normalized convolution framework [38]. It starts with a formal definition for the signal space, and then it employs some linear algebra concepts to derive the normalized convolution frameworks as a weighted least-squares problem. Finally, we provide some illustrative examples for
the normalized convolution in action as well as a review for confidence propagation criteria in the literature.

2.1 Signals as Functions

Signals can be defined in many ways, but in general, they can be conceived as an electric representation for a physical phenomenon. One source of signals is sensors, which can be realized as devices that map physical stimuli to electric signals. For instance, cameras comprise grids of visual sensors that are sensitive to light. Internally, these sensors produce electric signals that are proportional to light influx at each location in the grid. Cameras, eventually, produce 2D signals (images) encoding light variations within their field of view. To be able to analyze these signals, they first need to be modeled mathematically.

A common approach has been to model signals as functions that map sets of variables to measurements. In case of cameras, the output signal (images from the camera) can be depicted as a function that receives two variables: the horizontal and the vertical displacements along the sensor grid, and outputs a measurement of light intensity at this location. For a $w \times h$ image $f$, this function can be defined as:

$$f : X \to Y, \quad X \subset \mathbb{R}^2, \quad Y \subset \mathbb{R}$$  \hspace{1cm} (2.1)

where $X$ is the set of all possible permutations between horizontal and vertical displacements $\{x = (u, v) \in X \mid 0 < u \leq w, \quad 0 < v \leq h\}$, and $Y$ is the set of possible grayscale intensity values $\{y \in Y \mid 0 \leq y \leq 255\}$. Note that the definition above assumes a continuous signal, but in reality, image signals are discretized by sampling.

Signal Space

Viewing signals as functions allows defining function spaces\(^1\) that contain certain types of signals, e.g. audio, grayscale images, or 3D point clouds, which we denote as the signal space\(^2\). In the previous example of the $w \times h$ image $f$, we can define a signal space $V = \mathbb{R}^{w \times h}$, where the image $f$ is a single point in this space. When processing images, the signal space $V$ is naturally huge, and can be computationally infeasible.
to model. For instance and image of size $800 \times 600$, would constitute a signal space $V = \mathbb{R}^{800\times600}$, which has $800 \times 600 = 480000$ dimensions. This indicates that each pixel in the image corresponds to a dimension or a basis function in this signal space. Therefore, it is more common to process images locally over a smaller window. This way, the dimensionality of the signal space, i.e., the number of basis functions in $V$, is computationally manageable.

In practice, independent image pixels do not provide a powerful representation of image that qualify them to be used as basis. Instead, the basis functions are chosen as more representative functions depending on the application. This gives more flexibility and capacity when processing signals as the signal space can be constructed in a way that facilitates the occurring task. For example, if studying frequency responses of a signal are of interest, the basis functions can be chosen as complex exponential functions of different frequencies as in the Fourier transform. Alternatively, if processing the actual values of the signal is desired, the naïve basis can be utilized. Next, we describe some fundamental features of the signal space such as the scalar product, and the norm that are essential for exploiting the space.

### 2.2 Scalar Products and Norms

The general definition of the scalar product is any mapping from two equal-length vectors to a scalar. There is infinite ways for defining such mapping, but it can be considered as a feature of the signal space, i.e., two identical signal spaces with different scalar products are considered different. For a signal space $V$, a scalar product $\langle .| . \rangle$ can be defined as $\langle .| . \rangle : V \times V \to \mathbb{R}$. For any arbitrary signals $a, b, c \in V$, this scalar product is calculated as:

$$\langle a|b \rangle = b^\top a = \sum_i b_i a_i \quad (2.2)$$

where $\top$ is the transpose operation.

The following properties must be fulfilled for any scalar product:

1. Positive-definiteness:
   $$\langle a|a \rangle \geq 0 \ , \ \text{with equality iff} \ a = 0 \quad (2.3)$$
2. From Signal Spaces to Normalized Convolution

2. Linearity: over the first argument:\[\alpha \langle a | b \rangle = \langle \alpha a | b \rangle, \quad \alpha \in \mathbb{R}\] (2.4)

\[\langle a + c | b \rangle = \langle a | b \rangle + \langle c | b \rangle.\] (2.5)

3. Conjugate Symmetry:
\[\langle a | b \rangle = \langle b | a \rangle = \langle b | a \rangle^* (\text{for real-valued vectors})\] (2.6)

Note that the definitions above can be adapted to complex signal spaces over \(\mathbb{C}\) as well.

Norms

For a predefined scalar product, a norm can be induced for all signals \(v \in V\):
\[\|v\|^2 = \langle v | v \rangle = v^\top v\] (2.7)

The immediate question is: why is the formulation of a scalar product and its norm central in defining a signal space? The scalar product is broadly used as a projection operator within the signal space to find coordinates of a signal with respect to different basis, and to check for orthogonality. Give a signal \(v \in V = \mathbb{R}^n\), and the set of \(m\) orthonormal basis \(\{b_i \in \mathbb{R}^n\}_{1}^{m}\) that spans \(V\), the coordinate of the signal \(v\) under the basis \(b_k\) can be obtained as:\[r_k = \langle v | b_k \rangle\] (2.8)

In case that the coordinate \(r_k = 0\), this indicates that the signal is orthogonal to the basis \(b_k\) with no projection. Similarly, norms can be used to indicate the length of a signal \(v\) by projecting it on itself. This is particularly beneficial when calculating distances between signals.

Weighted Scalar Products and Norms

In some applications where the signal should not be uniformly weighted spatially, it is possible to introduce a weighting matrix \(W\), which is invertible: The definition in (2.2) becomes:
\[\langle a | b \rangle_W = \langle Wa | Wb \rangle = b^\top W^\top Wa = b^\top W^2 a\] (2.9)

\(^3\)If it applies over the second term, it is called conjugate linear.

\(^4\)In case of non-orthonormal basis, (2.8) gives the dual coordinate.
2.3. Signal Space Representation

The scalar product then acts as a spatially weighted sum using positive scaling factors. Equivalently, the norm in (2.7) can be redefined:

\[ \| v \|_W^2 = \langle Wv|Wv \rangle = v^\top W^\top Wv = v^\top W^2 v \quad (2.10) \]

With these definitions at hand, it is now possible to formulate a signal space in the next section.

2.3 Signal Space Representation

In this section, we use the definitions from earlier to formulate a signal space \( V = \mathbb{R}^n \) for a general set of basis functions \( \{ b_i \in \mathbb{R}^n \}_{i=1}^m \) that spans \( V \). Those basis functions can be arranged in the columns of an \( n \times m \) matrix \( B \), and any arbitrary signal \( f \in V \) can be represented in this space as:

\[ f = Br \quad (2.11) \]

where \( r \) are the coordinates of the signal \( f \) under the basis \( B \). The coordinates \( r \) can be obtained directly as \( r = B^{-1}f \) if and only if \( m = n \) and the basis functions in \( B \) are independent, i.e., \( B \) is square and non-singular.

Least-Squares Minimization

In practice, \( n \gg m \) leading to an overdetermined system, which is not guaranteed to have a solution. Therefore, we strive to minimize the discrepancy between the signal and its reconstruction from the signal space:

\[ \arg \min_{r \in \mathbb{R}^n} \| Br - f \|^2 \quad (2.12) \]

By applying the norm definition in (2.7), differentiating with respect to \( r \), and setting to zero to obtain the minima:

\[ \frac{\partial}{\partial r} (Br-f)^\top (Br-f) = 0 \ , \]

\[ \frac{\partial}{\partial r} (r^\top B^\top Br - 2r^\top B^\top f + f^\top f) = 0 \ , \]

\[ 2B^\top Br - 2B^\top f = 0 \ , \]

\[ B^\top Br = B^\top f \ , \]

If the basis matrix \( B \) has full column rank, then \( (B^\top B)^{-1} \) is invertible, and the solution to the minimization above reads:

\[ r = (B^\top B)^{-1}B^\top f \quad (2.14) \]
Otherwise, if $B$ is rank deficit, then it forms a frame or a subspace frame, which is not within the interest of this thesis. We refer the readers to [12] for more details about other scenarios.

**Weighted Least-Squares**

Now we consider the case where the signal is not uniformly weighted. We replace the norm in (2.12) with a weighted norm generated by an invertible weighting matrix $W$:

$$\arg \min_{r \in \mathbb{R}^n} \| Br - f \|_W$$

(2.15)

By employing the weighted norm definition in (2.10):

$$\frac{\partial}{\partial r} (Br - f)^\top W^2 (Br - f) = 0 ,$$

$$\frac{\partial}{\partial r} (r^\top B^\top W^2 Br - 2r^\top B^\top W^2 f + f^\top W^2 f) = 0 ,$$

$$2B^\top W^2 Br - 2B^\top W^2 f = 0 ,$$

$$B^\top W^2 Br = B^\top W^2 f ,$$

(2.16)

which has a unique solution if $B$ has full column rank:

$$r = (B^\top W^2 B)^{-1} B^\top W^2 f$$

(2.17)

**2.4 Normalized Convolution**

Given a finite discrete signal $f \in V = \mathbb{R}^n$, where $V$ is spanned by the set of basis in the columns of $B$, we assume an accompanying confidence map $c \in \mathbb{R}^n$. The confidence map describes the validity of each sample in the signal, where zero indicates missing/invalid, and a positive values indicate the level of confidence. In a similar manner, we can introduce a weighting function $a \in \mathbb{R}^n$ for the basis functions in $B$, which we denote as the applicability function. This function can be used to regulate focus on specific parts of the signal during processing.

We aim to use the formulation for weighted least-squares from (2.15) to minimize the discrepancy between the signal $f$, and its reconstruction from the subspace spanned by $B$, regulated by the confidence map $c$ and the applicability function $a$. To achieve this, we can employ the weights of the least-squares norm to encode the confidence maps and the applicability. Let $W_c = \text{diag}(c)$, $W_a = \text{diag}(a)$, where $\text{diag}()$ is a
diagonalization operator that places a column vector on the diagonal of a $n \times n$ matrix. Then, we set $W^2 = W_aW_c$ in (2.17), which leads to the following least-squares solution:

$$r = (B^\top W_aW_cB)^{-1}B^\top W_aW_cf$$  \hspace{1cm} (2.18)

This rightmost term of the right hand side shows that first, the invalid parts of the signal $f$ are discarded by $W_c$, then $W_a$ acts as some attention mechanism for specific parts of the signal, and finally the signal is projected to the subspace spanned by $B$. The resulting coordinates $r$ can then be used to reconstruct the signal as $\hat{f} = Br$. Note that in practice, the normalized convolution is applied at different points of the signal similar to the standard convolution, where $f$ refers to the neighborhood centered around each point of the signal.

The Applicability

As explained earlier, the applicability function allows regulating the importance of different parts of the signal. Usually, the choice of the applicability depends on the application and the nature of signals, however, there are no guidelines on how to choose the applicability. Most commonly, it is chosen as an isotropic bell-shaped function that has its peak at the center of the signal, and is monotonically decreasing towards the boundaries, i.e., assigning more importance at the center of the signal. Figure 2.1 shows two examples of such functions in case of 1D and 2D. Nonetheless, neither the isotropy assumption nor the choice of a bell-shaped function are guaranteed to be optimal consistently. Luckily, for data-driven approaches such as CNNs, it is possible to learn the optimal applicability function that minimizes the reconstruction error.

2.5 Normalized Averaging

To demonstrate the aforementioned concepts with a visual example, we show an example of the normalized convolution with the naïve basis on a synthetic 1D sinusoidal signal, and a 2D grayscale image. The naïve basis can be thought as a basis which yields the original signal value. First, we modify the normalized convolution definition in (2.18)
2. From Signal Spaces to Normalized Convolution

To incorporate the naïve basis \( \mathbf{B} = \mathbf{1} \), where \( \mathbf{1} \) is a column of ones:

\[
\mathbf{r} = (\mathbf{1}^\top \mathbf{W}_a \mathbf{W}_c \mathbf{1})^{-1} \mathbf{1}^\top \mathbf{W}_a \mathbf{W}_c \mathbf{f} \\
= \frac{\langle \mathbf{c} \cdot \mathbf{f} | \mathbf{a} \rangle}{\langle \mathbf{c} | \mathbf{a} \rangle} = \frac{\sum_i c_i f_i a_i}{\sum_i c_i a_i}
\]  

(2.19)

where \( f_i, a_i, c_i \) are elements of their corresponding vectors, \( \cdot \) is pointwise multiplication, and \( \mathbf{r} \) has now become a scalar that corresponds to a real signal value since we use a naïve basis.

We first apply this definition to a synthetic 1D finite signal \( \mathbf{f} = \sin(5t) + 2t, \ t \in [0, 2\pi] \) with additive noise. Figure 2.2a shows a plot of this signal that is artificially sparsified by randomly removing 30\% of the signal samples. When convolving the original signal with the Gaussian filter in 2.1a, we obtain a smoother signal in 2.2d. However, a border effect is produced at the tail of the signal because of the zero padding. The sparsified signal is also convolved with the same filter, and Figure 2.2e shows how corrupted the output signal is due to the missing samples. Finally, the sparse signal is convolved using the normalized convolution with the same filter as an applicability function, and a binary input confidence that is shown in Figure 1.2c. The output from normalized convolution shown in Figure 2.2f is almost identical to the convolved full signal. Nevertheless, it does not suffer from any boundary affects as the standard convolution.

The same can be performed on a 2D signal (an image), where the signal, confidence, and the applicability are flattened to vectors. Figure 2.3 shows the 2D example, where the same observations and conclusions can be made.
2.5. Normalized Averaging

Figure 2.2: An example for convolving a synthetic 1D signal and its sparsified version using the standard (std.) convolution and the normalized (norm.) convolution.

Figure 2.3: An example for convolving a 2D image and its sparsified version using the standard (std.) convolution and the normalized (norm.) convolution.
2. From Signal Spaces to Normalized Convolution

Boundary Effects in CNNs

Padding is widely applied in CNNs to maintain the spatial dimensionality of tensors. Typically, the padded values are chosen to be either zeros, replications of the boundary values, or reflections of values from the other boundary. However, all these approaches insert artificial values to the signal at the boundary, which was shown to degrade the performance of CNNs [31, 30]. Figure 2.4 shows an illustration for how padding is propagated between consecutive convolution layers. It is apparent that padding is progressively propagates towards the center of the image, the more convolution operations are applied. Of course, a CNN can learn to ignore the boundaries, but this could lead to a waste of valuable information. Normalized convolution can provide an appealing solution to this problem as the padding is assigned zero confidence, and instead, the network learn to predict the values at the boundaries that minimize the reconstruction error.

2.6 Confidence Propagation

A key aspect for incorporating normalized convolution into CNNs is how to propagate confidences between layers. Several measures have
been proposed in the literature for computing the output confidence from normalized convolution [66, 37, 12]. The output confidence should be mainly proportional to the magnitude of the input confidence, the sensitivity to noise, and how good the choice of the basis is to describe the signal.

But first, to understand the intuition behind these confidence propagation measures, we define two matrices: \( G = B' W_a W_c B \), and \( G_0 = B'B \). The matrix \( G_0 \) encapsulates the scalar products between all possible combinations of basis functions, i.e., how correlated and dependent they are:

\[
G_0 = \begin{bmatrix}
\langle b_1 | b_1 \rangle & \langle b_1 | b_2 \rangle & \ldots & \langle b_1 | b_m \rangle \\
\langle b_2 | b_1 \rangle & \langle b_2 | b_2 \rangle & \ldots & \langle b_2 | b_m \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle b_m | b_1 \rangle & \langle b_m | b_2 \rangle & \ldots & \langle b_m | b_m \rangle
\end{bmatrix}.
\]  

Similarly, the matrix \( G \) encloses weighted scalar products between different basis, i.e., how locally dependent the basis are given the applicability and the input confidence at a specific neighborhood of the signal. In a sense, \( G_0 \) can be assumed to have full confidence and a uniform applicability function. The two matrices are also referred to as the G-metric and they represent the degree of non-orthogonality of the basis in case of full and partial confidence. With these definitions at hand, we can describe confidence propagation measures in the literature.

Westelius [66] proposed the following criterion:

\[
c_{\text{out}} = \left( \frac{\det G}{\det G_0} \right)^{\frac{1}{m}},
\]  

(2.21)

The determinant of the \( G \) and \( G_0 \) gives a measures of how non-orthogonal the basis are in case of partial and full confidence. The formula exploits this property to produce an output confidence measure that is large if the input confidence is large, or if \( \det G < \det G_0 \) for large \( m \), i.e., \( G \) encodes more orthogonality than \( G_0 \).

Another criterion for propagating confidences was proposed by Karlholm [37]:

\[
c_{\text{out}} = \frac{1}{\| G_0 \|_2 \| G^{-1} \|_2},
\]  

(2.22)

where the norm \( \| \cdot \|_2 \) gives the largest singular value of a matrix. The reciprocal of the norm of \( G_0 \) would penalize non-orthogonality of the basis, while that of \( G^{-1} \) will reflect the level of input confidence.
There are several aspects to consider regarding these two measures. Firstly, they both reflect the level of input confidence, and the non-orthogonality of the basis, i.e., sensitivity to noise. However, they do not consider the righteousness of the basis choice. To address this concern, Farnebäck [12] suggested the use of the reconstruction error in (2.15) as a measure of confidence. Secondly, these measures produce a single confidence value for all basis since both the determinant in (2.21) and the norm in (2.22) map to scalars. Ideally, we would like to have a confidence value per basis vector to be able to discard non-orthogonal basis only. Finally, the two measures does not depend on the signal $f$, but rather on the basis and the input confidence only. This separation between the signal and its confidence can be beneficial when predicting the output confidence. This is demonstrated in Paper III as the signal prediction accuracy is not notably degraded by the output confidence estimation process.
Chapter 3

Deep Guided Filtering for Vision Tasks

Vision algorithms usually undergo several disturbances that can degrade their performance, and it is crucial to make them robust against them. A common disturbance is data noise that originates from sensors acquisition problems. For instance, several types of noise can occur in monocular image sensors due to illumination, internal processing of the signal, and sensor heat. This type of noise can usually be mitigated by improving sensor design, enhancing the acquisition process, or using post-processing techniques. Other disturbances can also stem from intermediate steps in the algorithm due to the nature of the underlying problem. As an example, in optical flow and stereo matching problems, false matches between image pixels can resemble data noise, and possibly degrade the overall solution produced by the algorithm. Therefore, those intermediate disturbances need to be considered when seeking a robustness.

There are several model-based solutions that attempt to robustify algorithms against disturbances through heuristics, regularization, and post-processing steps. However, they are usually hand-crafted and tuned for specific algorithms and scenarios. Alternatively, learning-based approaches can provide more flexible solutions that are driven from the data regardless of the algorithm or the scenario. Guided filtering is an example for these data-driven approaches, where a guidance modality is exploited for mitigating disturbances. For example, edge maps can be used as a guidance modality to rectify disturbances along edges. Nonetheless, it can be challenging to choose a reliable guidance
modality for each problem. Deep learning addresses this problem by offering a machinery for learning a suitable guidance directly from the data, which is denoted as Deep Guided Filtering.

In this chapter, we consider deep guided filtering for addressing disturbances in several vision tasks. First we start with a brief description for some sources of disturbances in major vision tasks such as depth completion, stereo matching and optical flow. Afterwards, we review different deep guided filtering approaches in the literature. This chapter complements the background material in Paper II, III.

3.1 Disturbances in LiDAR-Camera Setups

The LiDAR-Camera setup is very common in many applications where both natural images and depth data are needed, e.g. autonomous driving, and robotics. The camera provides textural and color information, while the LiDAR produces the scene depth. To be able to exploit this setup, both sensors need to be well calibrated to align data from the different sources. Nevertheless, there are several sources of disturbances in this setup that should be considered when processing the data. Figure 3.1 illustrates this setup and possible sources of disturbances.

Sensors Miscalibration

When calibrating the two sensors, it is necessary to estimate a mapping between the coordinate systems of the two sensors, which is usually a rigid transformation $[R|t]$ encompassing rotation $R$, and translation $t$. Estimating this mapping is usually sub-optimal due to human measurement errors, manufacturing deficiencies, and rounding errors. There are also other sources of error from calibrating the intrinsic camera matrix $K_{rgb}$, and rectifying the lens distortions. Figure 3.1 illustrates this setup. These errors usually result in data misalignments that ranges from few pixels to sub-pixel discrepancies between the two sensor grids. The misalignment is typically more critical along edges, and at fine details, as depth values are assigned to wrong objects.

Occlusion

Another source of disturbances are occlusions, which can be categorized into two types with LiDARs: laser, and sensor occlusion. The former occurs when the laser beams can not reach a region of the scene, while
3.1 Disturbances in LiDAR-Camera Setups

Figure 3.1: LiDAR-Camera cameras setup where it is desired to calibrate the LiDAR with a monocular camera. Disturbances might arise from miscalibration, occlusion (the occluded penguin), or reflective objects scattering the LiDAR beams (the shiny trophy).

the latter happens when an object is occluded by another due to the placement of the sensor. Both types lead to missing or disturbed measurements that are most evident along edges. Figure 3.2 shows an example of this scenario where a sensor occlusion causes faulty projections of points from the foreground and the background. This phenomenon can cause severe degradation for vision algorithms as foreground objects can be confused with background, leading to hazardous consequences in safety-critical applications.

Surface Characteristics

Since LiDARs use pulsed laser beams, they are governed by the same laws as natural light. When laser beams are directed towards some surface, the angle of the reflected beams can vary based on the characteristics of the surface such as reflectively, roughness, and density. For instance, surfaces with high reflectively tends to diffuse the reflected beams, while transparent surfaces can refract the laser beams changing their direction. These two aforementioned situations can either cause the laser beams to be lost, or change their characteristics leading to false measurements. This problem is usually very challenging to fix in
3. Deep Guided Filtering for Vision Tasks

3.2 The Correspondence Problem

Given two or more images of the same 3D scene taken at different poses or times, the correspondence problem aims to ensure correct matches of different regions between all images. This problem represents a core challenge of many vision task comprising multi-view images or video sequences, e.g. stereo matching, structure-from-motion, and optical flow. Nonetheless, finding the right matches between images is not a simple task, and is mainly governed by the nature of the images. For instance, the matching becomes challenging if images have repeated patterns, textureless regions, varying lighting, or camera occlusion. Figure 3.3 shows some examples of these scenarios.

Typically, these challenges can be alleviated in two ways; either by enhancing the matching process, or rectifying the false matches. The matching can be enhanced by employing more robust feature representations, and possibly adopting a multi-scale matching scheme [61]. Alternatively, false matches can be rectified as a later step in the al-
3.3. Deep Bilateral Filtering

Classical bilateral filtering is a non-linear operation that aims to reduce noise, and preserve edges in images by replacing corrupted pixels with a weighted combination of their neighbors. Typically, the weights are calculated using a Gaussian distribution centered around a feature representation of each pixel. Those features are usually selected as the spatial pixel location, and the color intensity values. For an image $I$, the bilateral filtering $BF[\cdot]$ is defined at pixel coordinates $p$ as [52]:

$$BF[I]_p = \frac{1}{w_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_t}(|I[p] - I[q]|) I[q],$$  \hspace{1cm} (3.1)

where $S$ is the set of neighbors, $G_{\sigma_s}, G_{\sigma_t}$ are Gaussian distributions for spatial, and tonal intensity weighting with standard deviation $\sigma_s, \sigma_t$ respectively; and $w_p$ is a normalization factor defined as:

$$W_p = \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_t}(|I[p] - I[q]|).$$  \hspace{1cm} (3.2)

The incorporation of intensity values in the pixel representation allows identifying edges easily compared to only using the spatial location. By representing an image in a higher dimensional space, e.g. 2D for location, and 3D for color, the image becomes sparse, and identifying edges becomes simpler than in the original 2D image space. This behavior of bilateral filtering also enables filtering noisy pixels that are inconsistent with their surroundings. However, the handcrafted choice
3. Deep Guided Filtering for Vision Tasks

of Gaussian parameters and the features limit the capabilities and the
generalization of the bilateral filtering.

Deep learning allowed learning the parameters of the Guassian as well as the features, eliminating the need for manual tuning of the functions \( G_{\sigma_u}, G_{\sigma_t} \) in (3.1). Jampani et al. [32] proposed a high-dimensional sparse convolution to learn free parameterization of the Gaussian filter within a gradient-descent optimization. They applied their approach to joint bilateral upsampling that uses a high-resolution guidance grayscale image to upsample a low-resolution color image. They also applied it to the problem of depth maps upsampling, and semantic segmentation mask upsampling, outperforming the standard bilateral upsampling.

Gadde et al. [14] proposed Bilateral Inception layer that performs bilateral upsampling using a learned Gaussian filter. The proposed layer was employed for upsampling and refining segmentation masks in CNNs as a replacement for the standard upsampling and CRF [39]. Their modified CNN outperformed the baseline on all datasets in comparison, while being faster. Barron et al. [2] proposed a fast bilateral solver that formulates the bilateral filtering as a differential optimization problem allowing it to be used within deep networks. The solver was tested in depth/stereo upsampling, colorization, and semantic segmentation, where it consistently improved the performance of the baseline.

3.4 Adaptive CNN Filters

CNN filters are by design spatially invariant, where the learning is globally averaged over the entire spatial dimensions. Furthermore, after training a network, the filters are fixed during inference. In order to make filters adaptive to the scene structure, it should be possible to change filters during inference based on the input. Braban-dere et al. [34] proposed the Dynamic Filter Networks that employs a filter-generating module to adapt the filters to the input sample during test time. The proposed module is both sample-specific, and position-specific adapting to local structural and photometric changes. Dai et al. [4] proposed Deformable Convolution Networks that learn a local offset for the CNN filters at each spatial location. This offset is predicted by an additional network during inference making the filter adaptive the the input during inference. They tested their approach on several ob-
3.5 Pixel-Adaptive Convolution

Inspired by the dynamic filter networks (DFN) [34], pixel-adaptive convolution (PAC) [58] learns an “adapting kernel” with the same dimensionality as the feature map, which is multiplied by the shared filter for local adaptation. In contrast to DFN, PAC can scale to huge networks with a large number of filter banks. The proposed approach was evaluated on upsampling depth, optical flow, and as a replacement for CRFs in semantic segmentation CNNs. It was also employed for monocular depth estimation in [19] to overcome the content-agnostic behavior of the standard convolution. Wannenwetsch and Roth [65] extended PAC to be confidence-aware, i.e., adapt filters based on how confident the network is about its predictions. This allows propagating high confidence regions to lower confidence ones, while respecting edges, and fine details.

Summary There have been many approaches in the literature for making CNNs content-adaptive. However, the majority of these approaches are either computationally expensive, e.g. PAC [58, 65], or they do not scale to big networks such as [34, 4]. In Paper II and III, we propose light-weight approaches that are content-adaptive, computationally very cheap, and scale to big networks. Furthermore, our approach in Paper III is fully unguided, i.e., it does not need a guidance modality and learns guidance directly from the target data.
Uncertainty in Neural Networks from a Bayesian Perspective

This chapter provides a brief overview of uncertainty modelling in machine learning from a Bayesian perspective, more specifically, in Neural Networks. We start by describing aspects of uncertainty in machine learning models, and how it can be captured. Afterwards, we outline the basic concepts of Bayesian inference, followed by an explanation for approximate variational inference. Next, we reflect these concepts on neural networks to formulate Bayesian Neural Networks (BNNs). This chapter aims to help the reader position our approach in Paper III with respect to existing BNN approaches. It is worth noting that it provides a gentle introduction into the topic, but for a more elaborate review, we refer the readers to [48, 15].

4.1 Uncertainty in Machine Learning

Machine learning algorithms are widely used in computer vision to exploit visual data, and find interesting patterns that can facilitate some underlying task. These algorithms can be categorized based on different characteristics, but they are commonly classified based on their learning paradigm. When labeled data with input/target correspondences is available, supervised learning approaches are commonly employed. These approaches essentially attempt to learn mappings between the input data and the corresponding target using specific model choices, e.g. linear, polynomial, or non-linear models. This mapping can be conceived as a function with a set of learnable parameters, and the ob-
4. Uncertainty in Neural Networks from a Bayesian Perspective

Figure 4.1: An illustration for epistemic and aleatoric uncertainty in machine learning problems. Epistemic uncertainty helps identifying out-of-distribution examples that lies outside the learned data manifold/distribution. Aleatoric uncertainty gives an indication for noise inherent in the data.

The objective of the algorithm is to capitalize on the available data to find the optimal parameters within a finite parameter space\(^1\). Consequently, the performance of these algorithms is governed by the magnitude and the diversity of the data, i.e., how representative the available data is for the underlying distribution that has generated it.

The dependency between performance and the characteristics of training data has been a practical dilemma in many machine learning algorithms. For instance, the exact same algorithm can produce completely different solutions based on the distribution of the provided data. In formal terms, the optimization process for finding the optimal model parameters can converge to different local minimas depending on the training data. Therefore, there is inevitable degree of uncertainty about the optimal parameters for a given machine learning model. This uncertainty is referred to as *epistemic uncertainty*, and it is primarily caused by the insufficiency of training data that can rule out all implausible parameters [15, 20]. As a result, the quality of predictions made by a model on unseen test data would highly depend on their proximity to the training data. Another source of uncertainty is noise inherent in the data, denoted as *aleatoric uncertainty*. The noise originates from sensor noise or measurement errors, and can mislead the model to learn false representations of the data. Figure 4.1 illustrates these two concepts. In pursuit of reliable machine learning models, both of the aforementioned sources of uncertainty need to be considered.

Probability theory offers the machinery needed for modelling uncertainty throughout several probabilistic interpretations. For instance,

\(^1\)If the parameter space is infinite, the machine learning algorithm defines as non-parametric and is not covered within this thesis.
Bayesian modelling strives to find the distribution that has most likely generated the data taking into account our beliefs about the origin of the data. Initially, those beliefs are realized as a prior distribution over the parameters space reflecting our knowledge. Afterwards, the data is observed and evaluated using a likelihood distribution at all possible choices of parameters. Eventually, the likelihood is used to update the prior through the Bayes’ theorem forming a posterior distribution. This posterior allows capturing the epistemic uncertainty within its parameters, while the aleatoric uncertainty is modelled by the likelihood distribution. Therefore, Bayesian modelling is indeed favorable whenever full uncertainty modelling is desired. In the next section, we will formally describe the basic concepts for Bayesian modelling/inference.

4.2 Bayesian Inference

Given a training dataset with inputs \( X = \{x_1, x_2, \ldots, x_N\} \), and their corresponding targets \( Y = \{y_1, y_2, \ldots, y_N\} \), it is desired to find a function \( f^\theta \), parameterized by \( \theta \) that can map the inputs to the targets; \( y = f^\theta(x) \). Before observing the data, we represent our knowledge about this function as a prior distribution \( p(\theta) \) over the space of parameters \( \Theta \). Afterwards, we start observing the data, and we define a likelihood distribution \( p(Y|X, \theta) \) that indicates how likely would we get the right target given the input and a set of parameters \( \theta \in \Theta \). The likelihood distribution is usually chosen as a Gaussian in regression problems\(^2\):

\[
p(y|x, \theta) = \mathcal{N}(f^\theta(x), \sigma^2 \mathbf{I}) ,
\]

where \( \sigma^2 \) is the variance of the noise disturbing the data. For classification problems, the most common choice for the likelihood is a categorical distribution, which can be represented by applying the Softmax function to the logits space:

\[
p(y = c|x, \theta) = \frac{\exp(f^\theta_c(x))}{\sum_{c'} \exp(f^\theta_{c'}(x))} ,
\]

where \( f^\theta_c \) is the function value for class \( c \).

To combine the prior and the likelihood, we derive a posterior distribution \( p(\theta|X, Y) \) that indicates which set of parameters that are the

\(^2\)In general, the likelihood can be chosen as any distribution, but Gaussian and Laplace distributions are the most widely used as they correspond to the L2 and the L1 error models in non-probabilistic setups.
most likely to map the inputs to the targets correctly. By invoking conditional probability and multiplication rules from probability theory:

$$
p(\theta | X, Y) = \frac{p(\theta, X, Y)}{p(X, Y)} = \frac{p(Y | X, \theta)p(X, \theta)}{p(Y | X)p(X)} = \frac{p(Y | X, \theta)p(\theta)}{p(Y | X)p(\theta)} ,
$$

which resembles Bayes’ theorem with a model evidence $p(Y | X)$. This formulation provides the dynamics for modelling our beliefs before observing the data as a prior. Afterwards, the model is updated based on the observed data yielding a posterior that allows predicting the plausibility of any choice of parameters. The question now is how can we employ this machinery for predicting an unknown target $y^*$ given an unseen test sample $x^*$?

**Predictive Distribution**

We want to predict the target value $y^*$ given the unseen input $x^*$, the training data $X, Y$, and taking into account the uncertainty about the parameters $\theta$. This can be achieved by marginalising the likelihood of the new input over the posterior through a posterior predictive distribution:

$$
p(y^* | x^*, X, Y) = \int_\theta p(y^* | x^*, \theta)p(\theta | X, Y) \ d\theta .
$$

This way, the epistemic uncertainty is captured through the marginalization, and the aleatoric uncertainty is modelled as a part of the likelihood distribution.

In practice, evaluating (4.4) is not always possible since the marginalization is only tractable for small models with a limited parameter space, where the posterior can be evaluated at all possible parameter values. The same applies to the model evidence $p(Y | X)$ in (4.3) that is calculated over the entire parameter space and dataset:

$$
p(Y | X) = \int_\theta p(Y | X, \theta)p(\theta) \ d\theta .
$$

Consequently, Bayesian inference, despite its great potential, is bounded by the model complexity and the dataset size.
4.3 Approximate Variational Inference

To scale Bayesian inference to larger models, we can replace the intractable marginalization over the true posterior distribution in (4.4) with an approximate distribution. This is referred to as variational inference and it aims to finding a tractable distribution \( q(\theta) \) that is as similar as possible to the true posterior. This can be achieved by formulating an optimization problem that minimizes the difference between the true and the approximate distribution. There are hypothetically infinite ways for comparing distributions, but we are interested in approaches that are analytically convenient. The Kullback–Leibler (KL) divergence [41] offers a suitable dissimilarity metric between probabilistic distributions, and can be employed for this purpose:

\[
\text{KL}(q(\theta) \mid \mid p(\theta|X, Y)) = \int \theta q(\theta) \log \frac{q(\theta)}{p(\theta|X, Y)} \, d\theta .
\] (4.6)

By invoking the Bayes’ theorem from (4.3) to expand the posterior:

\[
\text{KL}(q(\theta) \mid \mid p(\theta|X, Y)) = \int \theta q(\theta) \log \frac{q(\theta)p(Y|X)}{p(Y|X, \theta)p(\theta)} \, d\theta
= \int \theta q(\theta) \log \frac{1}{p(Y|X, \theta)} \, d\theta + \int \theta q(\theta) \log \frac{q(\theta)}{p(\theta)} \, d\theta + \int \theta q(\theta) \log p(Y|X) \, d\theta
= -\int \theta q(\theta) \log p(Y|X, \theta) \, d\theta + \text{KL}(q(\theta) \mid \mid p(\theta)) + \log p(Y|X) .
\] (4.7)

By rearranging terms:

\[
\int \theta q(\theta) \log p(Y|X, \theta) \, d\theta - \text{KL}(q(\theta) \mid \mid p(\theta)) = \log p(Y|X) - \text{KL}(q(\theta) \mid \mid p(\theta|X, Y)) . \tag{4.8}
\]

Since the KL divergence is always non-negative, we can get the lower-bound of the above equation by discarding the rightmost term:

\[
\int \theta q(\theta) \log p(Y|X, \theta) \, d\theta - \text{KL}(q(\theta) \mid \mid p(\theta)) \leq \log p(Y|X) . \tag{4.9}
\]

This formula is denoted as Evidence Lower Bound (ELBO) and it indicates that minimizing the KL-divergence to the posterior corresponds to maximizing the log evidence.

Accordingly, the optimization objective can be written as maximization for the log evidence lowerbound for a family of approximate dis-
4. Uncertainty in Neural Networks from a Bayesian Perspective

tributions $q$ parametrized by $\omega$:

$$\hat{L}(\omega)_{\text{ELBO}} = -\int_\theta q_\omega(\theta) \log p(Y|X, \theta) \, d\theta + \text{KL}(q_\omega(\theta) \| p(\theta)),$$

(4.10)

where $\omega$ are the parameters of the approximate distribution. The first term can be identified as the data term and it aims to maximizing the data log likelihood in conjunction with the approximate distribution. The second term can be thought as a regularization term by encouraging the approximate distribution to be as simple as possible, and ideally close to the true prior $p(\theta)$.

The objective can also be written for individual training samples as follows:

$$\hat{L}(\omega)_{\text{ELBO}} = -\sum_{n=1}^N \int_\theta q_\omega(\theta) \log p(y_n|f_\theta(x_n)) \, d\theta + \text{KL}(q_\omega(\theta) \| p(\theta)),$$

(4.11)

where the summation corresponds to the expected log likelihood over the data. This produces an approximate predictive distribution that reads as:

$$q(y^*|x^*, X, Y) = \int_\theta p(y^*|x^*, \theta) q_\omega(\theta) \, d\theta.$$

(4.12)

The choice of the approximate distribution $q(\theta)$ essentially depends on the underlying model. Since we are interested in uncertainty in CNNs, we briefly describe some choices for this distribution that are good approximators for CNNs.

4.4 Bayesian Neural Networks

Several approaches have been proposed in the literature attempting to find a good approximate distribution for neural networks. This pursuit is especially challenging in Convolutional Neural Networks (CNNs) due to the large number of filters, and non-linearities. Here, we describe some of the common approximations that were demonstrated to suffice with CNNs.

Monte-Carlo Dropout

In [16], it was shown that applying dropout before each layer with weights in neural networks is equivalent to performing variational inference. The dropout can be realized as multiplying the weights of each layer by binary vectors sampled from a Bernoulli distribution.
with a certain probability. For a neural network with $L$ layers, the network weights constitute the model parameters $\theta = \{W_i\}_{i=1}^L$. To apply dropout, we draw a set of $M$ binary vectors from a Bernoulli distribution $\{z^m_1, \ldots, z^m_L\}_{m=1}^M$, and multiply them with the weights, which gives $\{\bar{W}_1^m, \ldots, \bar{W}_L^m\}_{m=1}^M$. Given a test sample, the network prediction can be estimated as [16]:

$$\mathbb{E}_{q(y^*|x^*)}(y^*) \approx \frac{1}{M} \sum_{m=1}^M \hat{y}^*(x^*; W_1^m, \ldots, W_L^m).$$ (4.13)

This approach require running multiple forward passes with dropout, and then average over the predictions to get the final prediction, and the predictive uncertainty. However, this causes a major computational overhead for running multiple forward passes.

**Ensembles**

Despite that ensembles are non-Bayesian, they can offer a machinery for estimating the predictive uncertainty similar to Bayesian approaches. Lakshminarayanan et al. [42] argues that by training $M$ instances of the same networks that are initialized randomly on randomly shuffled data, we can capture uncertainty similar to Bayesian inference. During test time, the averaging of ensembles becomes identical to the Monte-Carlo Dropout averaging in (4.13). However, they argue that ensembles are easier to train, scale better to larger models, and require no major modifications to the training pipeline. But they share the same drawback with the Monte-Carlo dropout as they require storing and running multiple forward passes.

**Snapshot Ensembles**

A major drawback for ensembles is the need for training and storing $M$ instances of the model, which is a computational and storage burden. As a solution, Huang et al. [27] proposed to employ a cyclic learning rate schedule to reach several local minimas that are stored as instances of the ensemble. In contrast to the standard ensembles, they only require a single training session, and they still perform favorably in terms of the accuracy of predictions and the quality of uncertainty estimates.

**Summary** In this chapter, we described uncertainty modelling in neural networks from a Bayesian perspective starting from basic probabil-
4. Uncertainty in Neural Networks from a Bayesian Perspective

This derivation helps to understand different aspects and complications for capturing uncertainty in neural networks. Furthermore, it helps to position our probabilistic approach in Paper III in relation to existing Bayesian Neural Networks.
Chapter 5

Upsampling in CNNs

Upsampling is a fundamental task in Convolutional Neural Networks (CNNs), especially in problems where predictions are made at the input resolution. Essentially, CNN architectures tend to gradually reduce the spatial resolution of the input for computational efficiency, and to learn more abstract representations. For a simple image classification task, the prediction is a textual label, and there is no need for upsampling. On the other hand, tasks such as semantic segmentation, depth estimation, and optical flow, a CNN needs to produce a prediction at each point in the input. Therefore, many upsampling approaches are employed to restore the input resolution, and produce pixel-wise predictions. Figure 5.1 shows an example of the two cases.

(a) Classification network with a single prediction.

(b) Segmentation network with pixel-wise predictions.

Figure 5.1: Examples for single-output CNNs in (a), and CNNs with pixel-wise predictions in (b).
5. Upsampling in CNNs

A common CNN architecture is the encoder-decoder network (Figure 5.1b), where the resolution is sequentially decreased at the encoder, and increased at the decoder. Typically, increasing the resolution is achieved either using content-agnostic approaches such as the bilinear interpolation, content-adaptive approaches such as the transposed convolution, or a combination of the two. This combination is favorable since it produces a good trade-off between computational efficiency, and upsampling quality. In this chapter, we describe these different approaches, their strengths and weaknesses to motivate the need for our upsampler in Paper IV.

5.1 Classical Interpolation Methods

These approaches utilize fixed functions to modulate the interpolation process. In general, the interpolation can be represented as a dot product between a weighing function, and the interpolation support. The weighing function decides the contribution of each pixel within the support to the final interpolated value. However, it does not take into account the contents of the pixels, i.e., it is content-agnostic. The support is defined by mapping each pixel in the output grid to the input grid (inverse mapping), and then select a $2 \times 2$, or $4 \times 4$ neighborhood. Figure 5.2 shows an illustration for how the support is defined, and examples of the weighing function.
5.1. Classical Interpolation Methods

Figure 5.3: An example for resampling using classical interpolation methods. The nearest neighbor interpolation causes aliasing artifacts especially along edges, the bilinear produces a smoothed image, and the bicubic can produce some sharpening artifacts or invalid values along edges.

Nearest-Neighbor Interpolation

In this interpolation, the rectangular weighing function \( \Pi \) is employed, and it acts as a rounding operator that finds the closest pixel within the support to assign it to the output pixel solely. This method is quite simple, however, it can lead to several artifacts such as aliasing, and stair-case effect, especially along edges (See Figure 5.3).

Bilinear Interpolation

The bilinear interpolation employs the triangular function \( \Lambda \) to produce interpolation weights that are inversely proportional to the distance to neighboring pixels. It does not suffer from aliasing as the nearest-neighbor interpolation, but it can cause blurring artifacts along edges since the output is a weighted average of all pixel within the support.

Bicubic Interpolation

The bicubic interpolation performs similarly to the bilinear, but using a larger support of \( 4 \times 4 \), and a third-order polynomial as a weighing function. It produces sharper images than the bilinear interpolation as further pixels get lower weights, but it is computationally more expensive and can cause sharpening artifacts along edges due to the characteristics of the weighting function.
5. Upsampling in CNNs

5.2 Non-Parametric Upsampling

This category of approaches has no parameters, and they rely on inverting the downsampling process.

Unpooling

When the max pooling operation is used for downsampling in CNNs, there is a possibility for inverting this operation by storing the indices of the pooled pixels. For example, in the encoder-decoder network shown in Figure 5.1, if max pooling is used for downsampling in the encoder, the pooled pixels are replaced at their original positions in the decoder. However, other pixels that were not pooled are lost, and they are replaced with zeros in the decoder.

5.3 Parametric Upsampling

In content-agnostic approaches, such as the classical interpolation, and non-parametric upsampling, the upsampling does not adapt to the data. Contrarily, adaptive approaches can learn some upsampling function from the data that adapts to the contents.

Transposed Convolution

It is one of the most common learnable upsampling method in CNNs, which was originally designed to learn mid and high level feature representations [67]. Afterwards, it was widely employed for upsampling in encoder-decoder networks for semantic segmentation [51, 45], optical flow estimation [6], and Generative Adversarial Networks [55]. The key concept of transposed convolution is to propagate each pixel in the low-resolution data to a larger region in the upsampled data modulated by learnable filters. Figure 5.4 shows a simple illustration for transposed convolution. It is clear from the figure that if the filter size is not divisible by the stride, there is an overlap between different pixels. This overlap can lead to checkerboard artifacts which degrade the visual quality of the upsampled images. This can be mitigated by choosing a proper filter size, and stride that cause no overlapping. However, in practice, these artifacts can not be completely avoided since filters are shared spatially, and are optimized globally. Moreover, utilizing transposed convolution imposes additional computational and
5.4. Hybrid Upsampling Approaches

There are several hybrid upsampling methods, which combine both content-agnostic and adaptive approaches. The most straightforward approach is to perform the upsampling using any of the classical interpolation functions, followed by stacks of convolution layers for refinement. For instance, \cite{57, 56} upsamples the final prediction using the bicubic interpolation followed by convolution. However, this approach is not completely adaptive, but is rather considered as a refinement approach.

Alternatively, some of the guided filtering approaches described in Chapter 3 can be employed to perform content-adaptive upsampling. Nonetheless, these approaches are computationally expensive, and are usually trained separately from the main network as a post-processing step \cite{58, 65}. In Paper IV, we propose an adaptive upsampling approach based on the normalized convolutional neural networks that is adaptive, light-weight and trains end-to-end with any CNN. Figure 5.5 shows an example for the qualitative improvement in several optical flow estimation CNNs when attaching our upsampling to them during training.

The figure clearly shows the marginal improvement in the quality of predictions when employing our adaptive upsampler with a negligible overhead of 2000 parameters.
5. **Upsampling in CNNs**

![Figure 5.5: A qualitative example from several optical flow estimation CNNs on the FlyingChairs dataset [5]. Our upsampler produces sharper edges and manages to eliminate some flow errors at the orange chair.](image)
In this thesis, we investigated challenges associated with sparse data in computer vision tasks such as sparsity and uncertainty. Sparse data is typically generated by active sensors, such as LiDAR, RGB-D sensors, and other types of time-of-flight cameras, which are becoming key components in modern perception systems. Another cornerstone in these systems is deep learning, and more specifically, Convolutional Neural Networks (CNNs). CNNs provide a powerful learning paradigm that can exploit data from different modalities. Nonetheless, they were exclusively designed to operate on dense data on regular grids. When provided with sparse data on irregular grids, they tend to perform sub-optimally.

We proposed a novel sparsity-aware layer in Paper I denoted as the Normalized Convolutional Neural Network (NConv) that is capable of efficiently processing sparse data. The layer receives sparse data accompanied by a confidence map indicating the validity of each point in the input. The straightforward approach was to assume binary confidence mask, where zeros indicate the absence of data, and ones otherwise. When we tested our approach on the task of unguided depth completion, it achieved state-of-the-art results with a remarkably small network of ~500 parameters.

To investigate fusion with other modalities, we performed a thorough analysis for different fusion schemes in Paper II. We evaluated them both on LiDAR data with aligned RGB images, and RGB-D data. The top-performing scheme in both cases achieved state-of-the-art results with a small network that has at least one order of magnitude fewer parameters than other methods in the literature. This makes
our approach suitable for visual perception systems in robotics, autonomous driving, and surveillance. For instance, the NConv was used to perform aerial depth completion in drones in [62].

Another crucial aspect when dealing with sparse data is the uncertainty of the output. Therefore, we derived a probabilistic interpretation for our NConv in Paper III that is able to produce high-quality uncertainty maps with the output that are better or on-par with the existing expensive Bayesian Deep Learning approaches. In addition, it can learn the input confidence that minimizes the prediction error without using any auxiliary or guidance data. We evaluated this framework on the task of unguided depth completion and it was able to set a new state-of-the-art among both deterministic and probabilistic approaches in comparison.

Finally, we test our NConv on a popular task in CNNs, namely adaptive upsampling. We formulated the upsampling as a sparse problem, and employed our NConv to solve it in Paper IV. When applied to the problem of optical flow upsampling, we achieved state-of-the-art results on the most challenging optical flow estimation dataset. Moreover, the upsampler requires significantly lower number of parameters than the baseline. Generally, this upsampler can be employed with a regression problem such as stereo matching, or depth prediction.
Bibliography

[1] Stephen T Barnard and Martin A Fischler. “Computational stereo.” In: ACM Computing Surveys (CSUR) 14.4 (1982), pp. 553–572.

[2] Jonathan T Barron and Ben Poole. “The fast bilateral solver.” In: European Conference on Computer Vision. Springer. 2016, pp. 617–632.

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. “A naturalistic open source movie for optical flow evaluation.” In: European Conf. on Computer Vision (ECCV). Ed. by A. Fitzgibbon et al. (Eds.) Part IV, LNCS 7577. Springer-Verlag, Oct. 2012, pp. 611–625.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. “Deformable convolutional networks.” In: Proceedings of the IEEE international conference on computer vision. 2017, pp. 764–773.

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. “FlowNet: Learning Optical Flow with Convolutional Networks.” In: IEEE International Conference on Computer Vision (ICCV). 2015. URL: http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15.

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazırbaş, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. “FlowNet: Learning optical flow with convolutional networks.” In: Proceedings of the IEEE international conference on computer vision. 2015, pp. 2758–2766.

[7] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network.” In: Advances in neural information processing systems 27 (2014), pp. 2366–2374.
[8] Abdelrahman Eldesokey and Michael Felsberg. “Normalized Convolution Upsampling for Refined Optical Flow Estimation.” In: Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP. INSTICC. SciTePress, 2021.

[9] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. “Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 12014–12023.

[10] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. “Confidence propagation through cnns for guided sparse depth regression.” In: IEEE transactions on pattern analysis and machine intelligence (2019).

[11] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. “Propagating Confidences through CNNs for Sparse Data Regression.” In: The British Machine Vision Conference (BMVC). Northumbria University, Newcastle upon Tyne, England, UK. 2018.

[12] Gunnar Farnebäck. “Polynomial expansion for orientation and motion estimation.” PhD thesis. Linköping University Electronic Press, 2002.

[13] Stefan Fuchs. “Multipath interference compensation in time-of-flight camera images.” In: 2010 20th International Conference on Pattern Recognition. IEEE. 2010, pp. 3583–3586.

[14] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. “Superpixel convolutional networks using bilateral inceptions.” In: European conference on computer vision. Springer. 2016, pp. 597–613.

[15] Yarin Gal. “Uncertainty in Deep Learning.” PhD thesis. University of Cambridge, 2016.

[16] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning.” In: international conference on machine learning. PMLR. 2016, pp. 1050–1059.

[17] Ross Girshick. “Fast R-CNN.” In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Dec. 2015.

[18] Goesta H Granlund. “In search of a general picture processing operator.” In: Computer Graphics and Image Processing 8.2 (1978), pp. 155–173.

[19] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. “Semantically-Guided Representation Learning for Self-Supervised Monocular Depth.” In: International Conference on Learning Representations. 2020. URL: https://openreview.net/forum?id=BysxT7TNFvH.
[20] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schön. “Evaluating scalable Bayesian deep learning methods for robust computer vision.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. 2020.

[21] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Patrice Horaud. Time-of-flight cameras: principles, methods and applications. Springer Science & Business Media, 2012.

[22] Robert M Haralick. “Edge and region analysis for digital image data.” In: Computer graphics and image processing 12.1 (1980), pp. 60–73.

[23] Christopher G Harris, Mike Stephens, et al. “A combined corner and edge detector.” In: Citeseer. 1988.

[24] Kaiming He, Jian Sun, and Xiaoou Tang. “Guided image filtering.” In: European conference on computer vision. Springer. 2010, pp. 1–14.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778.

[26] Berthold KP Horn and Brian G Schunck. “Determining optical flow.” In: Techniques and Applications of Image Understanding, Vol. 281. International Society for Optics and Photonics. 1981, pp. 319–331.

[27] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. “Snapshot ensembles: Train 1, get m for free.” In: arXiv preprint arXiv:1704.00109 (2017).

[28] Junhwa Hur and Stefan Roth. “Iterative residual refinement for joint optical flow and occlusion estimation.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 5754–5763.

[29] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. “Uncertainty estimates and multi-hypotheses networks for optical flow.” In: Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 652–667.

[30] Carlo Innamorati, Tobias Ritschel, Tim Weyrich, and Niloy J. Mitra. “Learning on the Edge: Explicit Boundary Handling in CNNs.” In: Proceedings of the British Machine Vision Conference (BMVC). selected for oral presentation. BMVA Press, Sept. 2018.

[31] Carlo Innamorati, Tobias Ritschel, Tim Weyrich, and Niloy J. Mitra. “Learning on the Edge: Investigating Boundary Filters in CNNs.” In: International Journal of Computer Vision 128.4 (Apr. 2020), pp. 773–782. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01223-y. URL: https://doi.org/10.1007/s11263-019-01223-y.
[32] Varun Jampani, Martin Kiefel, and Peter V Gehler. “Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks.” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 4452–4461.

[33] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. “Sparse and dense data with cnns: Depth completion and semantic segmentation.” In: 2018 International Conference on 3D Vision (3DV). IEEE. 2018, pp. 52–60.

[34] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. “Dynamic filter networks.” In: NIPS. 2016.

[35] Younghyun Jo, Seoung Wng Oh, Jaeyeon Kang, and Seon Joo Kim. “Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation.” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2018.

[36] Takeo Kanade and Masatoshi Okutomi. “A stereo matching algorithm with an adaptive window: Theory and experiment.” In: IEEE transactions on pattern analysis and machine intelligence 16.9 (1994), pp. 920–932.

[37] Jörgen Karlholm. “Local Signal Models for Image Sequence Analysis.” PhD thesis. Linköping University Electronic Press, 1998.

[38] Hans Knutsson and C-F Westin. “Normalized and differential convolution.” In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE. 1993, pp. 515–523.

[39] Philipp Krähenbühl and Vladlen Koltun. “Efficient inference in fully connected crfs with gaussian edge potentials.” In: arXiv preprint arXiv:1210.5644 (2012).

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks.” In: Communications of the ACM 60.6 (2017), pp. 84–90.

[41] Solomon Kullback and Richard A Leibler. “On information and sufficiency.” In: The annals of mathematical statistics 22.1 (1951), pp. 79–86.

[42] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles.” In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6405–6416. isbn: 9781510860964.
Bibliography

[43] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. “Ssd: Single shot multi-box detector.” In: European conference on computer vision. Springer. 2016, pp. 21–37.

[44] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation.” In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 3431–3440.

[45] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation.” In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 3431–3440.

[46] Bruce D Lucas, Takeo Kanade, et al. “An iterative image registration technique with an application to stereo vision.” In: (1981).

[47] Fangchang Mai and Sertac Karaman. “Sparse-to-dense: Depth prediction from sparse depth samples and a single image.” In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2018, pp. 1–8.

[48] Kevin P. Murphy. Probabilistic Machine Learning: An introduction. MIT Press, 2021. URL: probml.ai.

[49] Pushmeet Kohli Nathan Silberman Derek Hoiem and Rob Fergus. “Indoor Segmentation and Support Inference from RGBD Images.” In: ECCV. 2012.

[50] Simon Niklaus, Long Mai, and Feng Liu. “Video Frame Interpolation via Adaptive Separable Convolution.” In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Oct. 2017.

[51] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation.” In: Proceedings of the IEEE international conference on computer vision. 2015, pp. 1520–1528.

[52] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédou Durand. Bilateral filtering: Theory and applications. Now Publishers Inc, 2009.

[53] Kaustubh Pathak, Narunas Vaskevicius, and Andreas Birk. “Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds.” In: 2009 IEEE International Conference on Robotics and Automation. IEEE. 2009, pp. 1631–1636.

[54] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. “Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image.” In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, pp. 3313–3322.
[55] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks.” In: arXiv preprint arXiv:1511.06434 (2015).

[56] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shabbaz Khan, and Michael Felsberg. “Learning fast and robust target models for video object segmentation.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 7406–7415.

[57] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.” In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 1874–1883.

[58] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. “Pixel-adaptive convolutional neural networks.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 11166–11175.

[59] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. “Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume.” In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp. 8934–8943.

[60] Stuart A Taylor et al. “CCD and CMOS imaging array technologies: technology review.” In: UK: Xerox Research Centre Europe (1998).

[61] Zachary Teed and Jia Deng. “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow.” In: Computer Vision – ECCV 2020. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Springer International Publishing, 2020, pp. 402–419. ISBN: 978-3-030-58536-5.

[62] Lucas Teixeira, Martin R. Oswald, Marc Pollefeys, and Margarita Chli. “Aerial Single-View Depth Completion with Image-Guided Uncertainty Estimation.” In: IEEE Robotics and Automation Letters (RA-L) (2020). doi: 10.1109/LRA.2020.2967296.

[63] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. “Sparsity invariant cnns.” In: 2017 international conference on 3D Vision (3DV). IEEE. 2017, pp. 11–20.

[64] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. “Sparse and noisy lidar completion with rgb guidance and uncertainty.” In: 2019 16th International Conference on Machine Vision Applications (MVA). IEEE. 2019, pp. 1–6.
[65] Anne S Wannenwetsch and Stefan Roth. “Probabilistic pixel-adaptive refinement networks.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 11642–11651.

[66] Carl-Johan Westelius. “Focus of attention and gaze control for robot vision.” PhD thesis. Linköping University Electronic Press, 1995.

[67] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. “Deconvolutional networks.” In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE. 2010, pp. 2528–2535.
Part II

Publications
Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-175307
Uncertainty-Aware Convolutional Neural Networks for Vision Tasks on Sparse Data

Abdelrahman Eldesokey