Low to High Dimensional Modality Reconstruction Using Aggregated Fields of View

by

Kausic Gunasekar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved June 2019 by the
Graduate Supervisory Committee:

Yezhou Yang, Chair
Qiang Qiu
Heni Ben Amor

ARIZONA STATE UNIVERSITY

August 2019

ABSTRACT

Autonomous systems that are out in the real world today deal with a slew of different data modalities to perform effectively in tasks ranging from robot navigation in complex maneuverable robots to identity verification in simpler static systems. The performance of the system heavily banks on the continuous supply of data from all modalities. These systems can face drastically increased risk with the loss of one or multiple modalities due to an adverse scenario like that of hardware malfunction, inimical environmental conditions, etc. This thesis investigates modality hallucination and its efficacy in mitigating the risks posed to the autonomous system. Modality hallucination is proposed as one effective way to ensure consistent modality availability thereby reducing unfavorable consequences. While there has been a significant research effort in high-to-low dimensional modality hallucination, like that of RGB to depth, there is considerably lesser interest in the other direction(low-to-high dimensional modality prediction). This thesis serves to demonstrate the effectiveness of this low-to-high modality hallucination in reducing the uncertainty in the affected system while also ensuring that the method remains task agnostic.

A deep neural network based encoder-decoder architecture that aggregates multiple fields of view in its encoder blocks to recover the lost information of the affected modality from the extant modality is presented with evidence of its efficacy. The hallucination process is implemented by capturing a non-linear mapping between the data modalities and the learned mapping is used to aid the extant modality to mitigate the risk posed to the system in the adverse scenarios which involve modality loss. The results are compared with a well known generative model built for the task of image translation, as well as an off-the-shelf semantic segmentation architecture re-purposed for hallucination. To validate the practicality of hallucinated modality, extensive classification and segmentation experiments are conducted on the University of Washington's depth image database (UWRGBD) database and the New

York University database (NYUD) and demonstrate that hallucination indeed lessens the negative effects of the modality loss.

ACKNOWLEDGMENTS

The journey through my master's program in the past two years has had a significant positive impact on me as a person as well as on my technical acumen and for that, I would like to thank all those who have been part of it. I owe a great deal to them, for it is they who have helped me grow from weakness to strength and helped instill in me the confidence to get through tough times.

I would like to begin by thanking my admirable advisor, Dr. Yezhou Yang for believing in me and my work when I have been skeptical. His constant push and words of encouragement kept me on track and not lose focus of my thesis. I am greatly thankful to Dr. Yang for guiding me and helping me understand the nature of research in this field. Thanks to him I saw myself grow as a researcher, learning to conduct experiments in an ethical as well as science approved way. He has been very generous in helping me get the resources when I desperately needed them so that I can focus only on my thesis and he has been kind enough to cut me some slack during times of incredible academic pressure. For all these reasons and more, I consider myself extremely lucky to have had him as my advisor and mentor. I would also like to thank Dr. Qiang Qiu for his guidance and patience during the entire period of my thesis. He has been an invaluable member of this thesis. His insightful suggestions have helped me direct my thesis in the right direction and complete it on time. Thanks to Dr. Heni Ben Amor for agreeing to be a presiding member of the thesis defense committee.

I would like to extend my thanks to the members of the Active Perception Group (APG). Most of my master's days were spent in the lab sharing the space and servers with my fellow group mates. They have been always available when I needed them and ensured that I had fun even when I worked. A special thanks to Zhiyuan Fang, who helped me during my paper submissions and the many rides from home to the lab. I would also like to thank Tejas Gokhale, Varun Jammula, Mohammad Farhadi, Xin Ye, Shibin Zheng, Aadhavan

Sadasivam, Anshul Rai, Rudra Saha and Sree Gowtham Josyula for being amazing lab mates lending a helping hand when needed.

I would be remiss if I did not thank my roommates and friends who have made this journey a very pleasurable and unforgettable experience. Thanks to Aravind Manoharan, Madhu Venkatesh, Bharathi Gunari Chandrashekar, Pravin Kumar Ravi,Preethi Venkateshan, Pavithra Ranganathan, Vignesh Namasivayam, Chandini Radhakrishnan, Balarupini Rajendran, Kevin J Thomas, Nidhi Dubey, Anuhya Nudurupatti for their unwavering support and company and many thanks to , Vishal Vaidyanathan and Sahan Vishwas for sharing their evenings over the two years in football which has been and will always be an integral part of my life. I'd like to thank Natalie Pollet as well for inspiring me to do better.

Finally, I would like to thank my mother, father, and sister for everything that's good in my life and for believing in me at all times.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Overview

Contemporary robotic systems and intelligent agents such as autonomous ground or aerial vehicles, smartphones and nimble security systems heavily rely on processing information from multiple sensory data streams to yield accurate and reliable decision-making results. Regardless of whether the system is complex or elementary, usually the systems are subjected to correlated data from numerous streams. One best way to ensure the efficacy and efficiency of these systems in terms of performance is to add redundancy into the system by incorporating data from all possible streams.

Recent advances in multi-modal information fusion mechanisms (Lahat, Adali, and Jutten 2015; Atrey et al. 2010; Khaleghi et al. 2013) have made it possible to widely adopt these techniques and incorporate them within the systems to ensure the best performance. Thus, it would be ideal for a system to access as many modalities as possible. When the decision system makes use of all these different streams of data, it is a necessary precaution to have more than one sensor for each modality just in case one of the fails. In practice, however, various constraints like the system budget, physical form factor, power budget, etc make it problematic to integrate the required redundancy. For example, lidar sensors used on top of self-driving cars can go up to $85,000 (Lidar cost) making it highly impractical to have more than one sensor.

One effective alternative that will help in beating the constraints and difficulties men-

tioned above is hallucinating[1] the data of the desired modality from another modality. For example, predicting what the depth map of an image would look like given its RGB image, with a trustworthy predicting method, we can replace the need of the redundant sensor with the predictor. An illustration of this has been portrayed in Fig. 1

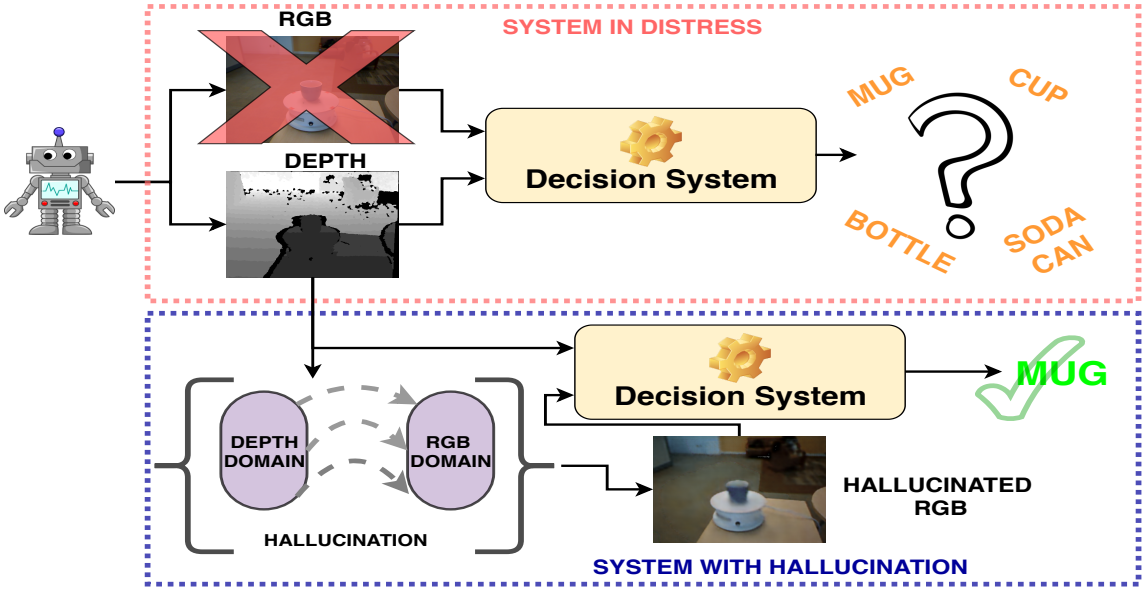


Figure 1: This illustration depicts a system in distress which leads to degradation of the sytem's ability to identify the object whereas a sytem that can hallucinate from its correlated modality identifies correctly.

1.2 Motivation

Biological organisms including humans have been shown to demonstrate this ability to hallucinate information from one modality to another in the event of losing a particular modality. As a human, if you're given the task to identify and move towards someone, you

[1]Hallucination in this paper means predicting but since we are predicting from a low to high dimensional space, hallucination is more apt for it has to imagine the information in higher dimensional space.

can use your eyes to navigate around the room and achieve the task. Here, you're using your eyes as your primary modality to achieve the task. Now in a scenario where you lose your primary modality i.e. if you go blind as a human you can still navigate around the room by listening carefully to the different voices in the room and follow the voice that you can recognize as the target. Studies (Bach-y-Rita, Tyler, and Kaczmarek 2003; White et al. 1970; Roe et al. 1992) have also provided evidence towards this demonstrated ability in living organisms to hallucinate which can be applied to rehabilitative purposes. We take inspiration from this biological capability and apply to the digital realm to improve the reliability and performance of the system.

Although sensors can generally function reliably for a long period of time, the lingering risk still exists that certain channels of the sensor array may fail at a critical time. The notorious case, where an autonomous car hits a pedestrian recently happened in Arizona US (Sacks 2018), there is speculation that the LIDAR sensor on the vehicle failed to function before the tragedy actually happened, and it is believed to be one of the crucial factors that caused the accident. In this thesis we simulate the case of loss of a higher dimensional modality and ask ourselves, are there any backup approaches (just like the hallucination capability of biological beings) an intelligent system can take to mitigate the risks or lower the likelihood of failure? We lay emphasis on utilizing the sensor channel with less information to hallucinate the information-rich sensor channel? Here, we put forward the first approach, to the best of our knowledge, that increases the reliability of a decision system involving multi-modal data. We consider a system that takes in two channels of sensory inputs: an RGB image channel and a depth channel. In this scenario, during normal conditions, the system has access to both the RGB as well as depth information and is considered as the training phase. The adverse scenario is treated as the testing phase where we the system has access only to one modality.

1.3 Challenges

Hallucinating from one modality to the other has its own set of challenges. Generally, hallucination is done between modalities that are highly correlated so that a mapping can be learned to extract the information of the lost modality. One can obtain very little or no practical benefit by hallucinating between unrelated modalities. In addition to this, we consider the scenario of hallucinating a higher dimensional modality from a lower dimensional modality. This makes it an ill-posed or underconstrained problem to be solved. To put this in perspective consider the scenario in Fig. 2. If the green point represents the starting position of the object and it has been instructed to move two units in the one-dimensional case the solutions are finite with just two solutions represented by the blue points. But in the case of 2D space, the solution becomes infinite as there are infinite points in the circle. Thus in this case hallucinating the solution from 1D to 2D, there could exist infinite solutions and so it becomes intractable. Hallucination or prediction from higher to lower dimensional modality has been studied widely and a good example of research in that direction would be RGB to depth prediction also known as single camera or monocular depth estimation (Eigen, Puhrsch, and Fergus 2014; Laina et al. 2016). We investigate the relatively less trodden path of hallucinating a information-rich RGB modality from a lower dimensional depth modality. This task is further constrained by the fact that we aim to produce a generic task agnostic method that can be used to generate the hallucinated data points instead of embedding representations with other networks.

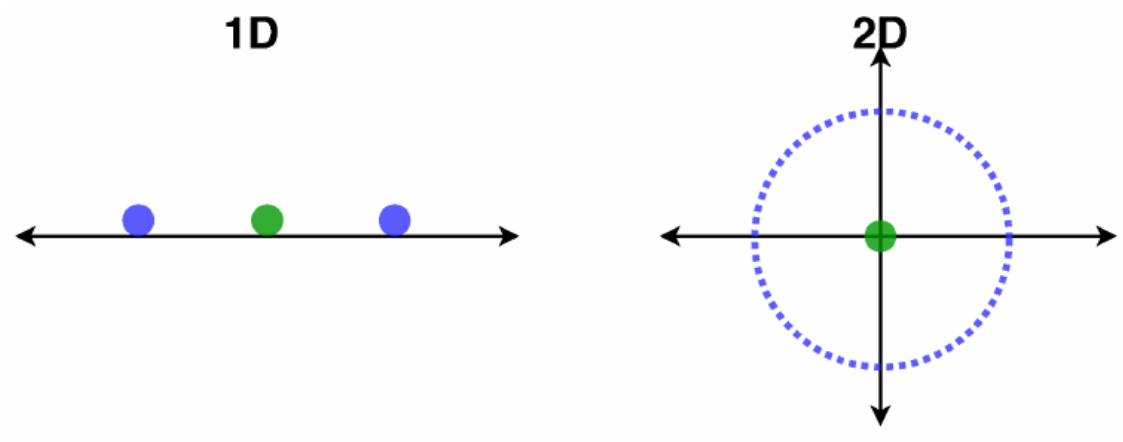


Figure 2: Illustration of the dilemma that exists in finding solutions in higher dimensional spaces.

1.4 Contributions

Following the insight of biological perception system as well as addressing the challenges, we model a custom neural network-based architecture that takes incorporates multiple fields of view to take into account the neighboring information at each layer of its network. Recent success in various computer vision tasks (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; Szegedy et al. 2015; K. He et al. 2016; Kaiming He et al. 2017; Redmon et al. 2016) encouraged us to go for a CNN-based encoder-decoder structure to model the mapping. We further constraint the learning process with an edge-aware smoothness loss term. The proposed hallucination scheme is generic enough to be used as a bridge for multiple tasks using correlated data modalities and doesn't have the requirement to be embedded into a task-specific network.

To summarize our contributions:

1. We investigate state of the art discriminative and generative neural architectures for the purpose of hallucination and we treat them as our baseline.
2. We propose a novel encoder-decoder based neural network architecture that incor-

porates multiple fields of view into its base encoder and decoder blocks to learn the low-to-high dimensional modality mapping. We compare the results of our architecture with that of the baseline results.

3. We design and conduct experiments on the well-known UW-RGBD and the NYUD datasets, and empirically show the advantage in hallucinating with our architecture over the baseline architectures. We further validate the usefulness of the hallucinated data by subjecting the hallucinated data to two fundamental vision task, namely, image classification and semantic segmentation.
4. We also experimentally show an added advantage that we observed with the hallucinated data. By incorporating the hallucinated data into the original system it can further improve the performance of the original system.

1.5 Outline

In **chapter 2** we will explain some recent work that has been of active interest in this domain as well other work that has applications in this field.

The **chapter 3** will elucidate the network architecture used for the baseline experiments as well as our proposed custom architecture. It will further delineate the details of the architecture used for the classification as well as segmentation. This chapter will describe the additional efforts undertaken to further better the performance of the networks as well as the explanation for the loss formulation.

The **chapter 4** will talk about the datasets used, the statistics regarding the same, the visual results from each of the baseline networks as well as our proposed networks, and the test results from the classification and segmentation networks which demonstrate the effectiveness of our proposed solution.

Chapter 2

RELATED WORK

2.1 Modality hallucination related:

Learning combined space representations and hallucinating data from different modalities is an active field of research (Lezama, Qiu, and Sapiro 2017; Hoffman, Gupta, and Darrell 2016; Christoudias et al. 2010; Srivastava and Salakhutdinov 2012) as it provides many advantages to the system that incorporates it. The work done in (Lezama, Qiu, and Sapiro 2017) is a very good example where the authors hallucinate RGB versions of the image from infra-red images and that is used in the face verification task. This, however, is targeted at a domain adaptation setting where the face verification model trained for RGB images is adapted to near infra-red images. The work by Hoffman et al. in (Hoffman, Gupta, and Darrell 2016) also deals with modality hallucination, although their method is used to learn mid-level abstractions and that is further used to enhance the performance of the detection network. The learning of their hallucination network is embedded as part of their object detection module and it learns by loss function paired with the depth stream. They do not produce hallucinated data and are also restricted to a specific task. Other works have used a mapping between modalities to help better the performance of a system (Christoudias et al. 2010) or use a generative model to learn the distribution of modalities and sample from them when needed (Srivastava and Salakhutdinov 2012). In (Christoudias et al. 2010), the authors learned a mapping using the unlabeled data with the help of Gaussian processes and that is leveraged for the object recognition task of objects previously not seen in the training data. The scenario is quite different as they propose to tackle missing data instances for their

task while our work is focused on tackling missing modality using data hallucinated with the help of CNNs. The work done in (Srivastava and Salakhutdinov 2012) uses generative models, specifically deep Boltzmann machines to help with the missing data. Generative models come with their own disadvantages and an important one is that they do not produce a one-to-one mapping as required in these tasks, instead, it learns the distribution which may not well describe our missing data modality. Our work is substantially different as ours is not a generative model and we tackle the case of an entire modality missing and not missing instances of data.

2.2 Multi-modal information processing:

Multi-modal systems are becoming more common recently and consequently, there has been increased interest in this field of research (Castrejon et al. 2016; Vrečko et al. 2009; Salvador et al. 2017; Li et al. 2003; Socher et al. 2013; Gupta, Hoffman, and Malik 2016; Xu et al. 2015; Karpathy, Joulin, and Li 2014; Huang, Peng, and Yuan 2017). Work such as (Socher et al. 2013; Gupta, Hoffman, and Malik 2016) deal with the learning of cross-modal data but differ in their learning process in the sense that they do not hallucinate the data in any manner similar to ours. While the former transfers images to the semantic text space and uses it to help in classification the latter uses a learned model to transfer its learning for the same task on the other modality. (Xu et al. 2017) work deals with RGB and thermal data modalities for the case of pedestrian detection. But in their work, they are using RGB images to reconstruct thermal images and using them on their detection network. Unlike their task, we are hallucinating RGB information which is relatively widely used modality and more information-rich for conventional vision tasks compared to corresponding depth or thermal modality thus making the task harder. Also, our hallucination scheme is not

task-specific and neither does it require to be embedded alongside a task-specific network to be used. A lot of work has been done that deals with multi-modal information processing. Work done in (Castrejon et al. 2016; Vrečko et al. 2009; Salvador et al. 2017; Li et al. 2003) involves learning cross-modal representations and associating the modality embeddings to learn the relationship between the modalities. This learned information about the modalities is used to perform a specific task on a given data modality as in (Salvador et al. 2017) or used together as in (Vrečko et al. 2009).Recent work done in (Xu et al. 2015; Karpathy, Joulin, and Li 2014; Huang, Peng, and Yuan 2017). also involves learning a common subspace that incorporates information from all the modalities, and knowledge transferring or association of semantic information is done at this space. The above methods don't use the learned representation to hallucinate lost modalities neither do they talk about handling such events that involve losing a complete data modality.

2.3 GANs using multimodal data:

Generative Modelling has been an active field of research even before the advent of deep learning to model the distribution of data. (Sutskever, Hinton, and Taylor 2009; Dayan et al. 1995). Recent advances in the theory of deep learning, as well as increased availability of cheap, compute and abundant data, has catalyzed the research in this domain. Deep learning methods (Kingma and Welling 2013; Goodfellow et al. 2014) have shown promise of better modeling the data distribution. GANs, in particular, have become a hotbed in the academic community as well as the industry. GANs have been proven to be particularly good with multi-modal data such as image and text (Zhang et al. 2017; Mansimov et al. 2015; S. Reed et al. 2016; S. E. Reed et al. 2016) or even between different image domains (Isola

et al. 2017; Zhu et al. 2017; Karras, Laine, and Aila 2019), which makes it a literature of our interest.

GANs learn by transforming a latent variable which could be a random noise vector or data from one input domain into a data in the other target domain. This is done using a generator network that produces the image of the target domain and a discriminator that is used to identify between the fake and real image. The objective function in most of these methods have adversarial loss as one of the main components. The work done in Isola et al. 2017 is a supervised method for image translation that uses paired images from both domains to learn the distribution implicitly and then is used to generate. This can be used in our work by setting ours' as an image translation problem as well.

Chapter 3

HALLUCINATION AND VALIDATION NETWORKS

3.1 Introduction

There has not been a lot of work in the domain of modality hallucination that particularly satisfies the constraint of being task agnostic and generic. Consequently, there are no established baselines in this domain which we can use or further develop on. Therefore, we created our own baselines by re-purposing existing networks that achieve state of the art results in other tasks for the purpose of modality hallucination. We empirically validate the hallucination performance of these networks against the hallucination performance of the custom architecture that we propose. In the following sections, we explain the different architectures we experimented on as well the architectures used for the validation experiments.

3.2 Crossspectral Hallucination Architecture:

We first started with an architecture used for the purpose of hallucination. We use a part of the architecture similar to the one used by (Lezama, Qiu, and Sapiro 2017). Their paper deals with hallucination from near infrared spectrum to the RGB spectrum for the purpose of face recognition. They use a patch-wise hallucination scheme where different parts of the face are taken in small patches hallucinated after applying an affine transformation. We applied the hallucination for the entire image instead of adopting their patchwise hallucination. In their work, the authors were concerned with hallucinating only

the face and so patch wise hallucination made sense. This method doesn't make sense when applied to datasets like the ones we are dealing with which has a myriad of different objects in them. The architecture we used is described in table 1. The network has 11 layers taking in a 3 channel depth image as the input to produce a 3 channel RGB image as output. Each layer has a ReLU activation function which follows a batch normalization layer. The final layer is not subjected to batch normalization or ReLU.

3.2.1 Coarse to Fine Modelling:

Since the hallucination procedure in the original paper by Lezama, Qiu, and Sapiro 2017 adopts a patchwise hallucination mechanism the network only had small images as input. (In the paper the size of each image input is only 32x32). As we adopt a full image hallucination our inputs are much bigger. Thus adopting their architecture with our image inputs leads to an impractical training routine. To overcome this shortcoming we propose a progressive coarse to fine modeling scheme as illustrated in Fig. 3 that can help to speed up the training procedure.

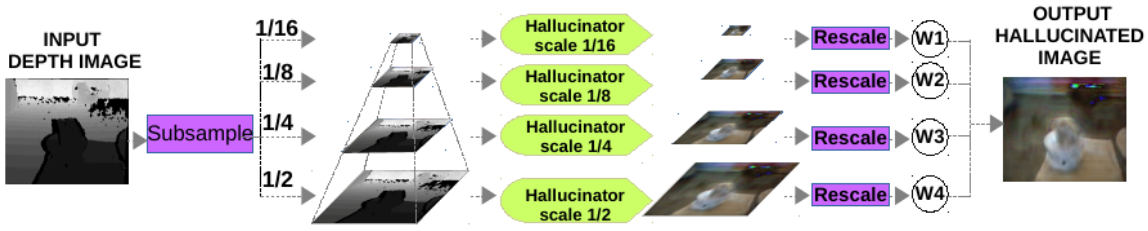


Figure 3: Coarse to fine model to speed up training routine. This happens at four scales.

We train the hallucinators at different scales making the model progressively learn

Layer	Specification
Layer 1	kernel size : 11 x11, Filters : 147 pad : zero, batch-normalization : yes, activation : ReLU, strides : [1,1]
Layers 2-10	kernel size : 11 x11, Filters : 36 pad : zero, batch-normalization : yes, activation : ReLU, strides : [1,1]
Layers 11	kernel size : 11 x11, Filters : 36 pad : zero, batch-normalization : no, activation : Linear, strides : [1,1]

Table 1: Cross spectral Hallucination Network Architecture from Lezama, Qiu, and Sapiro 2017 modified for hallucinating RGB from depth

to capture dominant patterns as the weaker patterns die out in the sampling process. A weighted linear combination of the different model outputs can be used to obtain the final output. These weights can be implemented as learnable parameters as well. Adopting the coarse to fine modeling scheme helps the network to gain over a 5-time speed up. The major bottleneck in the training routine is the number of operations from the convolutions. The proposed coarse to fine model reduces the number of convolution operations thereby leading to faster training. This has been explained below using equations 3.1,3.2,3.3.

$$\begin{aligned} C_{\text{total}} &= \frac{W}{S_c} \cdot \frac{H}{S_r} \\ &= \frac{W}{S_c} \cdot \frac{H}{S_r} \qquad (\textit{lower bound}), \end{aligned} \tag{3.1}$$

where W and H represent the width and height of the feature maps. S_c and S_r represent the strides across the columns and the rows and C_{total} is the total number of convoultion operations without the coarse to fine . We consider the case of convolutions with zero padding which is the same as in our architecture. . represents a ceiling function. In Eqn. 3.1 we consider the lower bound of the ceil function. Therefore, that equation represents the minimum number of convolution operations without the progressive modeling, per layer. The number of convolution operations after the modeling is adopted is shown below. Here

we consider the upper bound of the ceil function. Therefore, this represents the maximum number of convolution operations after the modeling is adopted.

$$\begin{aligned} C_{\text{model}} &= \Sigma_{i=1}^{N} \frac{W}{2^i.S_c} \cdot \frac{H}{2^i.S_r} \\ &= \Sigma_{i=1}^{N} (\frac{W}{2^i.S_c} + 1).(\frac{H}{2^i.S_r} + 1) \\ &= \frac{W.H}{S_r.S_c} \Sigma_{i=1}^{N} \frac{1}{2^2 i} + \Sigma_{i=1}^{N} \frac{1}{2^i} (\frac{H}{S_r} + \frac{W}{S_c}) + \Sigma_{i=1}^{N} 1 \\ &= C_{total} \Sigma_{i=1}^{N} \frac{1}{2^2 i} + \Sigma_{i=1}^{N} \frac{1}{2^i} C_{total} (\frac{S_c}{W} + \frac{S_r}{H}) + N \\ &= C_{total} \frac{(1-(\frac{1}{4})^N)}{3} + C_{total} (1 - (\frac{1}{2})^N)(\frac{S_c}{W} + \frac{S_r}{H}) + N. \end{aligned} \tag{3.2}$$

C_{model} is the number of convolution operations after the progressive modeling has be adopted, N represents then number of levels of the pyramid in the sub-sampling process. We recommend, $2 < \text{N} < 6$ and we used N = 4. In practice, stride values are usually less than 4 and the spatial dimension of the image are significantly greater than the stride values. Thus, Eqn.3.2 can be approximated as follows,

$$\begin{aligned} C_{model} &\approx C_{total} \frac{(1-(\frac{1}{4})^N)}{3} \\ C_{model} &\approx 33\%\ C_{total} \quad when\ N = 4, \end{aligned} \tag{3.3}$$

Thus with 4 scales, we had 67% lesser operations which explains the speed up. We carry out our experiment with this procedure.

3.3 GANs for Hallucination:

We investigated the fit of generative adversarial networks for the purpose of modality hallucination. We explored the possibility of using a generative model to learn the distribution of the data and then to conditionally generate the output. Recently, GAN's have been shown to work well in implicitly learning the distribution. In particular, we are interested in conditionally learning the distribution of the data. This way, the output of the

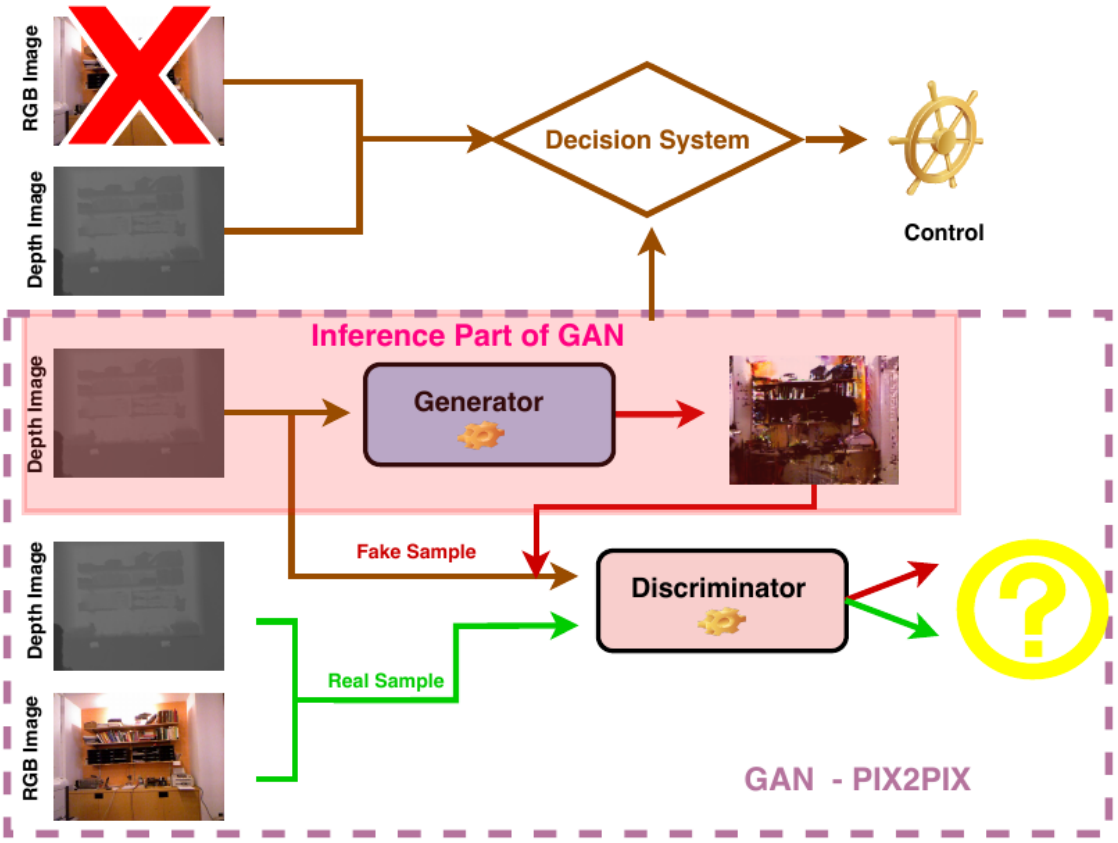


Figure 4: Illustration of the pix2pix GAN for modality hallucination.

learned model will produce an RGB image conditioned on the given input depth image thus producing an RGB image from a depth image. A particularly well-known work in this domain of conditional GAN is work done by Isola et al. 2017 and Zhu et al. 2017 who's work popularly known by their academic references as pix2pix and cycleGAN. While cycleGAN is completely self-supervised with unpaired image to image translation pix2pix is supervised with image pairs of the different modalities given together during training and does a better job at this task. An illustration of the pix2pix model for modality hallucination can be seen in Fig. 4.

The working of pix2pix model is similar to most GAN's. The Generator model takes input from the extant modality which is the depth image and tries to produce the corre-

sponding RGB image. The input depth image and the generated RGB image are given together to the discriminator network as the fake sample while the depth image and its corresponding RGB image is given as the real sample into the discriminator network. We use the authors' pytorch implementation of pix2pix to train on the same dataset keeping as many hyper-parameters of the experiment same to ensure a fair evaluation. Once trained we subject the results to the same validation experiments as the other hallucination mechanisms.

3.4 Hallucination using LinkNet:

Modality reconstruction networks described in our process takes in an image as input and gives an image as output which represents the reconstructed modality. Semantic segmentation networks, although built for another task also has images as input and output. Moreover, the state of the art semantic segmentation networks (Badrinarayanan, Kendall, and Cipolla 2017; Ronneberger, Fischer, and Brox 2015; Zhao et al. 2017; Chen et al. 2017) use an encoder based architecture that compresses the original image to smaller space which captures the semantic information of the image and a progressively expanding set of decoder layers to leverage the semantic information to segment them. Thus, semantic segmentation networks are ideal candidates that can be re-purposed for hallucination.

We chose linkNet (Chaurasia and Culurciello 2017) which is also an encoder-decoder architecture that performs better than most of the state of the art model. the architecture of linkNet is described in table 2. The architecture features four encoder and four decoder blocks. The encoder and decoder blocks are based on the Resnet architecture. Skip connections from each encoder layer to the corresponding decoder layer are provided to better incorporate early information from the encoder space. Like the previous architecture mentioned in section 3.2 each layer has batch normalization followed by ReLU activation.

((a)) Network Architecture

Layer #	Type	Kernel dimensions		Stride	Skip
		Spatial kw $\times kh$	Depth inp $\times out$		
1	conv	7 x 7	3 x 64	2	-
2	max pool	3 x 3	64 x 64	2	-
3	encoder	3 x 3	64 x 64	-	-
4	encoder	3 x 3	64 x 128	-	-
5	encoder	3 x 3	128 x 256	-	-
6	encoder	3 x 3	256 x 512	-	-
7	decoder	3 x 3	512 x 256	-	layer 5
8	decoder	3 x 3	256 x 128	-	layer 4
9	decoder	3 x 3	128 x 64	-	layer 3
10	decoder	3 x 3	64 x 64	-	-
11	FCN	3 x 3	64 x 32	0.5	-
12	conv	3 x 3	32 x 32	1	-
13	FCN	3 x 3	32 x 3	0.5	-

((b)) ResNet Blocks

Encoder - Decoder Block					
Layer #	Type	Kernel dimensions		Stride	Skip
		Spatial kw $\times kh$	Depth inp $\times out$		
Encoder Block					
1	conv	w x h	D_{in} x D_{out}	2	-
2	conv	w x h	D_{out} x D_{out}	1	1
3	conv	w x h	D_{out} x D_{out}	1	-
4	conv	w x h	D_{out} x D_{out}	1	2
Decoder Block					
1	conv	1 x 1	D_{in} x $D_{in}/4$	2	-
2	conv	w x h	$D_{in}/4$ x $D_{in}/4$	0.5	-
3	conv	1 x 1	$D_{in}/4$ x D_{out}	1	-

Table 2: (a) Linknet architecture is elaborated here. Here FCN represents full convolution layers that up-samples the given layer. A fractional stride represents an up-sampling while a integer stride represents down-sampling of that layer. (b) The encoder and decoder blocks used inside the linknet architecture is elaborated here. w and h are the width and height and D_{in} and D_{out} are input depth and output depth of the feature maps. They take the values from Table (a).

Since the application of this thesis is oriented towards the robotics and autonomous systems it is important to keep in mind the constraints faced by the above-mentioned systems. The linkNet architecture was specifically designed for fast inference and to keep the parameters low helping the time and memory budget. This made the linkNet an ideal choice to be used for the purpose of modality hallucination.

3.4.1 Regularizing Autoencoder:

From our experimental results (explained more in detail in chapter 4 the hallucinator network while it did a good job in predicting the structural information of the objects it did not preserve the color information and finer details well enough. Hence, to maintain the color information as much as possible we introduce a second stage into the hallucination

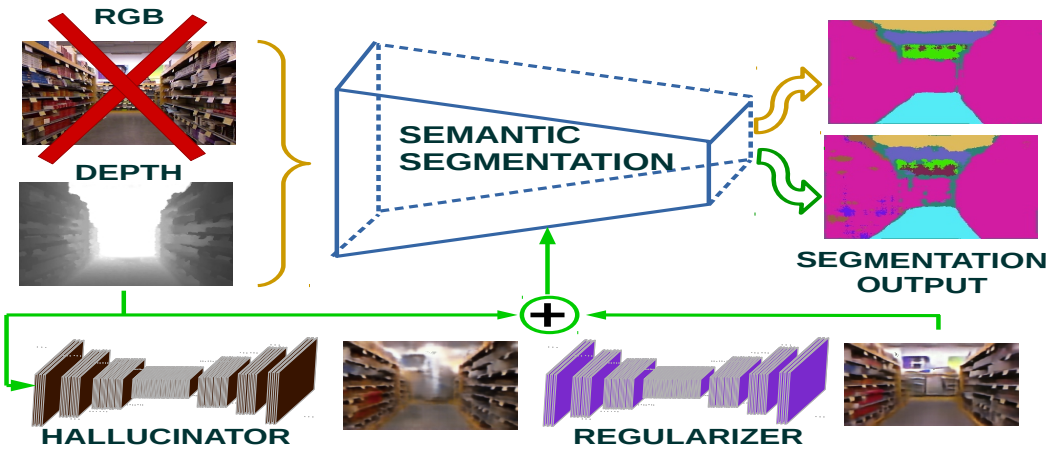


Figure 5: Illustration of how the two stage hallcination procedure is adopted to help a semantic segementation network in adverse scenario.

procedure. We use another model that has the same architecture as the linkNet architecture. it accepts the hallucinated results from the 1st stage and tries to hallucinate the original RGB image. Since this network accepts hallucinated RGB and only tries to preserve the original color information of the network, it is essentially a regularizing autoencoder. In practice to achieve higher convergence the regularizing autoencoder it is trained completely from scratch with randomly initialized weights as it helps to learn a hierarchical set of features solely responsible for correcting the image. Using the trained weights from the hallucinator network is an option but this will not help as the hallucinator kernels have a very different set of functions compared to the regularizers kernels and therefore they are trained from scratch. An example illustration of this pipeline is shown in Fig. 5.

3.5 Hallucination by Aggregating Multiple Fields of View:

The architectures explained in the previous sections experimented for the purpose of modality hallucination differed in the structure of the architecture or the way they learn to predict the information of the lost modality. All these networks focus on learning kernels with a single specific field of view determined by the kernel size. Since we are dealing with an ill-posed problem statement it would help the hallucinating network to make use of information obtained from different fields of view. Different fields of view encapsulate various degrees of information to pass on to the next feature map. For example, while hallucinating from the depth image of an apple a small 3x3 kernel would extract the features from a small patch of the apple. In a lower dimensional modality, this is not going to have much information due to the nature of how information is represented in lower dimensional modalities making the prediction task more difficult. Say, in the depth image if the 3x3 kernel is trying to predict from the center of the apple all it could see in the depth image is the distance and a box at the same distance could as well be predicted as an apple. On the other hand, in the higher dimensional modality which is RGB, it has access to the color information of the apple making it easier for the predictor. Thus if the hallucinator could have access to other fields of view the hallucinator can make predictions better. Considering the same example, if bigger fields of view exist the hallucinator can make out relationships of the object with its neighbors. Considering the same example, the apple could be associated with a fruit basket in the training distribution and now with the bigger fields of view it could incorporate the information from its neighbors to create the features thus adding more information progressively in the network

Based on the above-mentioned theory of the benefits of the added fields of view we

Network Architecture				
Name	Layer	Filters	Skip	Kernels
Enc_1	Encoder	48	-	-
Enc_2	Encoder	60	-	-
Enc_3	Encoder	192	-	-
Enc_4	Encoder	288	-	-
Dec_4	Decoder	96	-	-
Dec_3	Decoder	30	Enc_3	-
Dec_2	Decoder	24	Enc_2	-
Dec_1	Decoder	3	Enc_1	-
logits	Convolution	3	-	5x5
Hallucinated	Convolution	3	-	3x3

Table 3: This table describes the complete architecture used in our experiments.

Encoder - Decoder Blocks			
Layer	Filters	Stride	Skip Connection
Encoder block : Input -> Filters : d			
AggConv	d	1	NO
ReLU	-	-	-
AggConv	d	1	YES
ReLU	-	-	-
Convolution *kernel : 3x3* *dilation rate : 1*	d	2	NO
Decoder block : Input -> Filters : d			
AggTrConv	d	0.5	NO
ReLU	-	-	-
AggTrConv	d	1	YES
ReLU	-	-	-

Architecture Building Blocks				
Layer	Kernels	Filters	dilation rate	stride
AggConv Block : Input -> Filters : d , Stride : s				
convolution	3 x 3	d/6	1	s
	11 x 11	d/6	1	s
	5 x 5	d/6	2	s
	7 x 7	d/6	2	s
	9 x 9	d/6	3	s
	11 x 11	d/6	3	s
concatenation	-	-	-	-
batch Normalization	-	-	-	-
AggTrConv Block : Input -> Filters : d , Stride : s				
convolution	3 x 3	d/3	1	s
	7 x 7	d/3	1	s
	11 x 11	d/3	1	s
concatenation	-	-	-	-
batch Normalization	-	-	-	-

Table 4: Encoder - Decoder blocks constructed using the AggConv and AggTrConv blocks. This table describes the basic building blocks that is used in the architecture in table

propose an architecture that incorporates the different fields of view in each layer which is illustrated in Fig. 6 and tables 4,3. The architecture uses an encoder-decoder architecture like some of the other previously mentioned architecture. We define a base convolution block called Aggregated convolutions block (AggConv) which performs convolution with different sized kernels. To get bigger fields of view we use atrous convolution instead of just expanding the size of kernels which helps to save memory and time. The convolutions

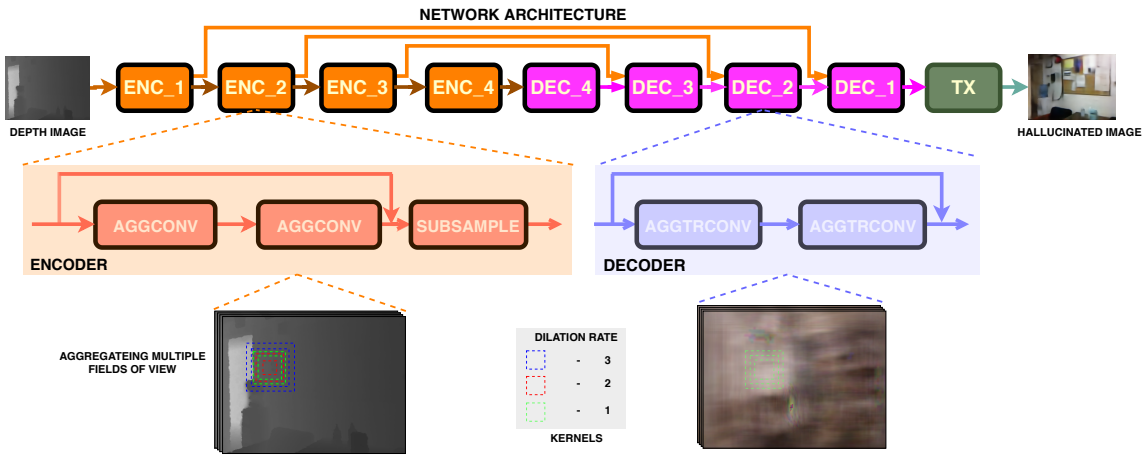


Figure 6: Illustration of the proposed architecture that aggregates multiple fields of view.

of the different kernels happen at different dilation rates as mentioned in table 3. The feature maps resulting from the different convolutions are concatenated to get a single set of feature maps after which batch normalization is applied followed by activation (ReLU). Finally, the block has an optional convolution operation with just 3x3 kernel to sub-sample if it is needed. All convolution operations have 'same' padding. Similar to the AggConv block, we also define an aggregated Transpose convolution (AggTrConv) that performs the upsampling by concatenating the transposed convolutions of the different kernels. Batch normalization is applied after, followed by ReLU activation. With these base AggConv blocks and AggTrConv blocks, we construct the encoder and decoder blocks of the architecture. The encoder has AggConv block followed by another one with sub-sampling by a factor of 2. Skip connections are added as well from the input of the encoder block to the output. The decoder block also follows the same structure with AggTrConv blocks instead of AggTrConv. The architecture is defined with 4 encoder blocks and 4 decoder blocks. The activation output from each encoder layer is added with the corresponding decoder layer as shown in Fig. 6.

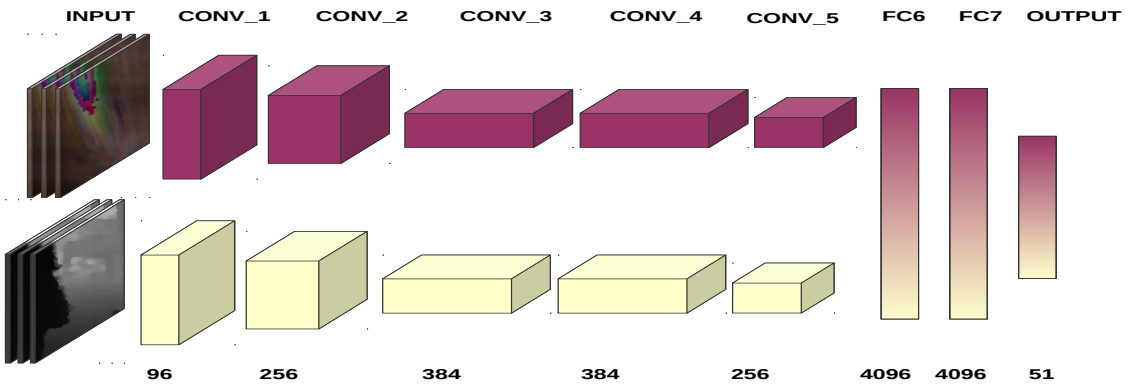


Figure 7: Two stream Alexnet configuration used for combining modalities and performing object classification.

3.6 Hallucinated Modality Helps :

The aim of the thesis is to show that the hallucinated modality can be utilized to help the system in the adverse scenario. While analyzing the visual output and the mean pixel error of the hallucinated modality with the ground truth RGB gives us a general idea of the performance of the hallucinator network but doesn't tell much if the hallucinated modality is, in fact, useful to mitigate the risk the system is facing. To validate the pragmatic use of the hallucinated modality we further devise experiments. We subject each of the modality as well as the combination of the modalities to an object classification task. We use the standard AlexNet configuration (Krizhevsky, Sutskever, and Hinton 2012) for its light network structure. To demonstrate the usefulness of the hallucinated modality we design a two-stream Alexnet that takes input from each of the modalities i.e. depth and the hallucinated modality to perform the classification task. An illustration of the two stream classifier can be seen in Fig. 7. We construct this by subjecting each modality to

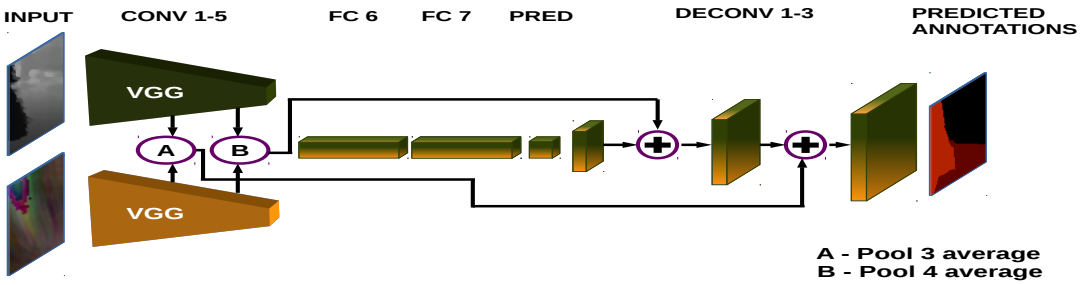


Figure 8: Two stream semantic segmentation network configuration used for combining modalities.

an Alexnet stream and then averaging the first fully connected layer which is the fc6 layer of each network and is fed into a single fully connected layer. We decided on fusing at the deeper end of the network as the deeper layers of the network capture more complex and meaningful abstraction relative to the earlier layers.

A similar setting is created for the task of semantic segmentation as well. We follow the fully convolutional architecture by Shelhamer, Long, and Darrell 2017 again for its light architecture. The segmentation network is also reconstructed with two streams to support multimodal segmentation that can take in the depth and the hallucinated images. A depiction of the two stream segmentation network is shown in 8. We use the VGG framework Simonyan and Zisserman 2014 to extract the features and we fuse at the first linear layer fc6. We also add the pooling responses from the fourth and third pooling layer of the VGG network in the deconvolution procedure to produce the predicted annotation. For two stream network, we average the pooling responses of both the streams at the 3^{rd} and the 4^{th} layer.

3.7 Hallucinated Modality Enhances:

Since the hallucinated network can directly be incorporated into the system, the decision system can use the additional information obtained from the hallucinated modality. We verified this by devising an experiment that takes in all the modalities along with the hallucinated modality to see if it performs better than the system without the hallucinated modality. We apply this on both the segmentation network as well the classification network. The networks are re-designed to take in three inputs namely, depth, RGB and the hallucinated modality, Similar to the two stream architectures mentioned above for the classification network the first fully connected layer of the Alexnet of the three modalities are averaged before giving it as input into the final connected layer. Similarly, the three stream segmentation network also takes input from the three different modalities and is fused into a single output layer in a similar manner as the two stream network. The performance of these networks is then compared to the two stream counterparts that take the RGB and depth as input.

3.8 Loss Formulation:

The loss for the hallucination experiments $\mathscr{L}_{hal}$ is designed with two components. The root mean square error $\mathscr{L}_{rmse}$ and the smoothness constraint $\mathscr{L}_{smooth}$. λ is used to adjust the relative importance between the two loss terms. We obtain a nice pixel-wise loss as shown in the equation. 3.4.

$$\mathscr{L}_{hal} = \mathscr{L}_{rmse} + \lambda \mathscr{L}_{smooth}. \tag{3.4}$$

The root mean squared error between the hallucinated images from the model and the ground truth RGB image helps the hallucinator to learn to hallucinate the structure and learn

the important abstraction between the two modalities. The main goal of this hallucination network is to capture the non-linear relationship between the data domains.

The Eq. 3.5 works well to capture the said abstraction.

$$\mathscr{L}_{rmse} = \sqrt{\mathscr{L}_{mse}} \quad = \sqrt{\frac{\sum_{i=1}^{N}(p_i - \bar{p}_i)^2}{N}}, \tag{3.5}$$

where N represents the number of pixels in target image I, and p_i, $\bar{p}_i$ the ground truth and reconstructed pixel respectively. To obtain a consistent and smooth mapping of the output we introduce an edge aware smoothness constraint. Smoothness constraints are commonly used in depth prediction like the work in Ranftl et al. 2016; Godard, Mac Aodha, and Brostow 2017. The smoothness constraint should enforce local smoothness and at the same time should preserve the edges. The Eq. 3.6 is used to do the same.

$$\mathscr{L}_{smooth} = \frac{1}{N}\sum_{N} H(\nabla I_{hal}) e^{-H(\nabla I)} \tag{3.6}$$

here I_{hal} represents the hallucinated tensor, I the ground truth, and H is the *Huber* function (Huber et al. 1964) that is formulated as:

$$H(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \leq \delta \\ \delta(|x| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}, \tag{3.7}$$

where $\delta = |x|$, and N the total number of pixels within one training batch.

The regularizer network mentioned for the linkNet architecture is trained with the mean squared error $\mathscr{L}_{mse}$ as shown in Eq. 3.5. The mean squared error helps to penalize heavily the difference in the hallucinated images thus helping in reconstructing the edges of the image. The classification and segmentation networks are subjected to the standard softmax cross entropy loss as shown in Eq. 3.8.

$$\mathscr{L}_{ce} = -\sum_{i=1}^{C} \bar{y}_i \log y_i, \tag{3.8}$$

where $\mathscr{L}_{ce}$ is the cross entropy loss for the task and C is the number of classes $\bar{y}_i$ represents the ground truth annotation while y_i is the predicted output. For classification the loss computed is between the prediction distribution and the class labels. For segmentation it is a pixel-wise classification between the annotated map and the predicted one.

The generator and the discriminator networks are trained together using a loss function that is formulated as a min-max game. The loss function has conditional adversarial loss along with a L1 loss as shown in equations 3.9 which is taken from Isola et al. 2017's work.

$$\begin{aligned} \mathscr{L}_{cGAN}(G,D) &= \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x,G(x,z)))], \\ \mathscr{L}_{L1}(G) &= \mathbb{E}_{x,y,z}[\| \, y - G(x,z) \, \|_1], \\ G* &= \arg\min_G \max_D \mathscr{L}_{cGAN}(G,D) + \lambda \mathscr{L}_{L1}(G). \end{aligned} \tag{3.9}$$

To get a general idea of the closeness of the hallucinated results to the RGB image we report the Mean absolute pixel difference as the error metric given by the equation 3.10. p_i and $\bar{p}_i$ are the ground truth and reconstructed pixels.

$$\mathscr{MAPD} = \sum_{i=1}^{N} \frac{1}{N} |p_i - \bar{p}_i| \tag{3.10}$$

Chapter 4

EXPERIMENTS AND RESULTS

We describe here the experimental details of the thesis and delve further into the results of the different networks. We explain the results of the different experiments and also talk about the other auxiliary experiments involved in the process.

4.1 Dataset

Here in we got into the nuances of the data used for the different experiments. To ensure fairness the data for all the experiments are the same. That is, the training dataset and the testing dataset are the same for the different experiments. Also, the training and test datasets are shuffled and after that, the train test split is not changed. So the training and test sets of the classification experiments, that is done on the hallucinated images from the depth images of the linkNet architecture is the same set of depth images from which the hallucinated images for classification of the aggregated architecture. They do not mix and are from the same distribution.

4.1.1 Hallucination:

We design our experiments on the following two datasets: NYUD dataset Silberman and Fergus 2011 and the University of Washington's RGBD dataset Lai et al. 2011. The datasets mentioned above have RGB images and their corresponding depth images. The UW-RGBD dataset has over 200,000 images belonging to 51 classes. Although the UW-RGBD dataset

has over 200K images the dataset is heavily skewed. For instance, some classes have less than 2000 images while others have over 10,000 images. To make sure there is no untoward bias, the dataset is split into 875 images per class for training and 100 images per class for testing. Hence, in total 44,625 training images and 5100 testing images are obtained for the hallucination experiment. The NYUD dataset, on the other hand, has only 2284 labeled images. So we used the raw dataset available in the NYUD-V1 Silberman and Fergus 2011. The raw dataset, unlike the labeled dataset, contains depth images which are not in-painted along with their corresponding RGB images and there are over 100000 such pairs. The NYUD V1 dataset has in total 135,314 RGB - depth image pairs. The raw depth images are in-painted to remove artifacts using a cross bilateral filterParis and Durand 2006 and then projected onto the RGB plane and linearly scaled to get the depth image representation. The images were split into train and test set with an 80:20 ratio.

4.1.2 Classification:

For classification, we used 500 images for training and 175 images for testing per class from the UW-RGBD dataset. Thus in total 25,500 training images and 8925 testing images are used for classification. None of these training and testing images overlap with the hallucination dataset. The dataset was subjected to a 51-way classification task. The entire image was not used as input due to data leakage (visual cues present in the image other than the object of interest that helps the network in the classification). We cropped the images with the given mask such that it contains only the object of interest. After subjecting it to a transformation we feed it into the classifier stream. Our choice of Alexnet is due to the fact that it is a relatively smaller model and can be easily transformed into two and three

stream models to fit within memory and, also, since the dataset for classification is relatively simple, it does not need a complex model.

4.1.3 Segmentation:

The segmentation experiment was carried out with NYUD-v2 dataset that has 2284 labels from 64 different indoor scenes. The depth images are in-painted to fill holes just like it was done for the Hallucination dataset. The dataset was split into 70:30 training-testing split. The segmentation task was 40 class segmentation procedure. The hallucinated network trained on NYUD-v1 raw images was used to obtain the hallucinated images here.

4.2 Implementation Details :

The hallucination experiments are done with the images maintained in their standard size preserving the same aspect ratio which is 640x480. The hallucination experiments were carried out in the YUV colorspace. The YUV colorspace which is mostly used for image and video transmission encodes the color information efficiently enabling lesser transmission error. To achieve this, the colorspace splits an image signal into one luma part(Y) that holds the structural information and two chrominance part (U and V) that hold the color information. It is easier for the network to learn information using data that is organized in a manner like the YUV colorspace than using data represented in an additive color model like the RGB space. Other than the color conversion we did not do any pre-processing steps.

4.2.1 Cross spectral Hallucination Architecture:

The cross-spectral Hallucination architecture with the adopted coarse to fine modeling was carried out with the same configuration of hyper-parameters for all the scales. We adopted 4 scales at 1/2, 1/4, 1/8, 1/16 and the combination of the output of the different models for each scale is done with weights 0.50, 0.28, 0.12, 0.1 respectively. The experiments are run in parallel on 3 NVIDIA GTX 1080 GPUs running Ubuntu 14.04. All the experiments for this architecture were run with a batch size of 5 images for a total of 30 epochs.

4.2.2 LinkNet Hallucination Architecture:

The linkNet architecture did not need the use of coarse to fine modeling. The hallucination procedure was carried on two different datasets. The hallucination procedure was carried on two different datasets. The hallucination procedure was carried on two different datasets. The linknet hallucination network was trained with a batch size of 24 for 15 epochs on a 16 GB Nvidia Tesla P100 GPU. ADAM optimizer Kingma and Ba 2014 with learning rate = 0.0005 , β_1 = 0.9 , β_2 = 0.999, ε=1e-08 was used to update the weights during backpropagation. The Huber delta δ was set to 0.001 and the smoothing weight λ was set to 50. The regularizer was trained with the same hyper-parameters as well.

4.2.3 GAN Hallucination Architecture:

The generative model based hallucination procedure explained in the chapter 3 is implemented using the author's source code. For both the NYUD as well as UWRGBD

dataset we use the following hyperparameters. The models are trained on a single NVIDIA Tesla P100 GPU with 16 GB RAM using pytorch (Paszke et al. 2017) framework. The hallucination models for both the datasets are trained for 150 epochs with 100 epochs at the base learning rate of 0.0002 and 50 epochs of linearly decaying the base learning rate to 0. The optimizer of choice like the other experiments is ADAM with $\beta_1 = 0.5$, β_2 and ε are maintained at 0.999 and 1e-08 respectively. The original image size is kept at 286x286. During training, the images are randomly cropped too 256x256. The experiments are carried out with a batch size of 32. No form of data augmentation is done to keep the experiments consistent. The number of generator filters and discriminator filters are maintained at 64 as mentioned in the original paper. The hallucination here is done in the RGB space.

4.2.4 Aggregated ConvBlock Hallucination Architecture:

The hallucinations using Aggregated convolution block architecture is implemented as a multi-GPU training pipeline. For both the NYUD dataset hallucination as well UW-RGBD hallucination the experiments were carried out with the same hyper-parameters. The architecture was implemented using data parallelism on 3 GPUs. A total batch size of 21 is used in training with each GPU taking in 7 images per batch. Like the linkNet architecture and the cross-spectral hallucination architecture, this is implemented using tensorflow(Martı´n Abadi et al. 2015). The architecture we used has 4 encoder layers and 4 decoder layers totaling 8 layers. We also experimented with 6 layers (3 encoders and 3 decoders) and 10 layers (5 encoders and 5 decoders) architecture. The test loss profile during the training procedure is shown in Fig. 9. The 8 layer architecture with 4 encoders and 4 decoders seemed to be optimal. The time and memory burden from the 10 layer architecture compared to the 8 layer architecture is significantly higher compared to the gains

in performance. This can be observed in the table. (Note that the different architectures in the figure 9 were trained with variable numbered GPUs. This explains the different training times of the architectures and as a result the different sample length summaries.) 9.

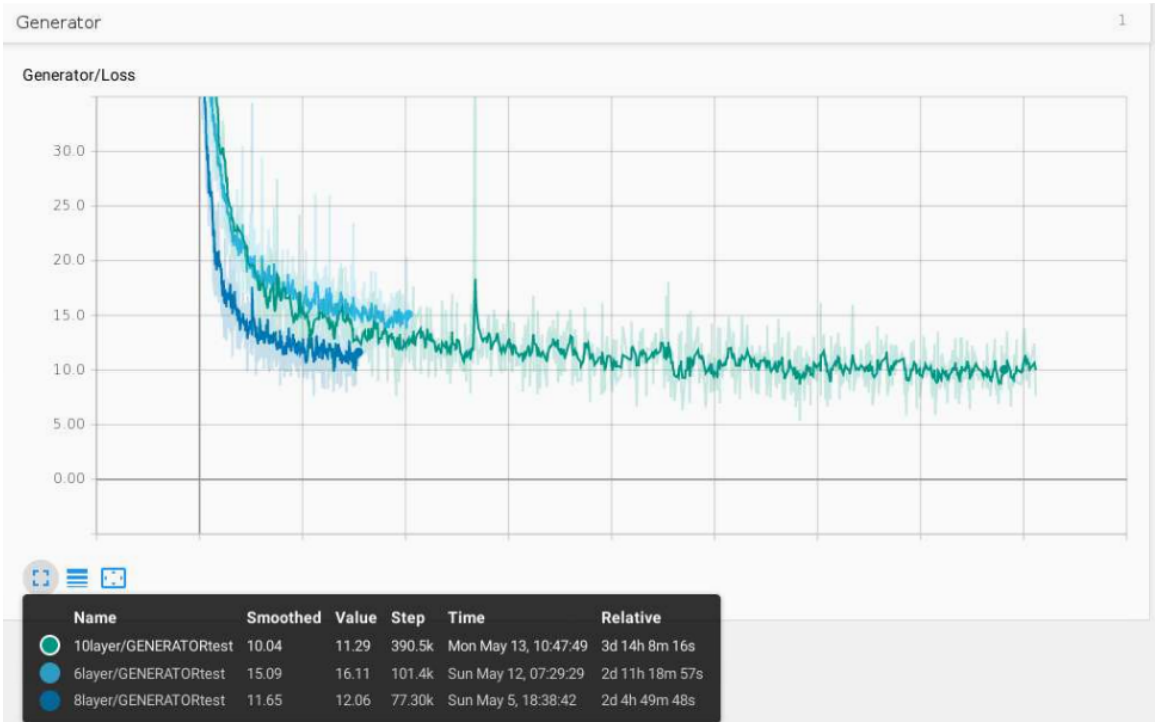


Figure 9: The test error profile while training of the differnt architectures.

Another hyperparameter in our architecture is the dilation. We designed the architecture to take the information of the local neighborhood, therefore, keeping the dilation rates between the different kernel layers small. We empirically tested the effect of having bigger dilation rates as well. Hence, the same architecture with the same kernels and hyper-parameters is subjected to the bigger dilation rates. The new dilation rates are 1,5 and 10 instead of 1,2 and 3. We observed that this didn't cause much difference and the performance was the same. This can be inferred from Fig. 10 and table 9. The "small dilation " refers to dilation rates of 1,2 and 3 while the "big dilation" refers to dilation rates of 1,5 and 10.

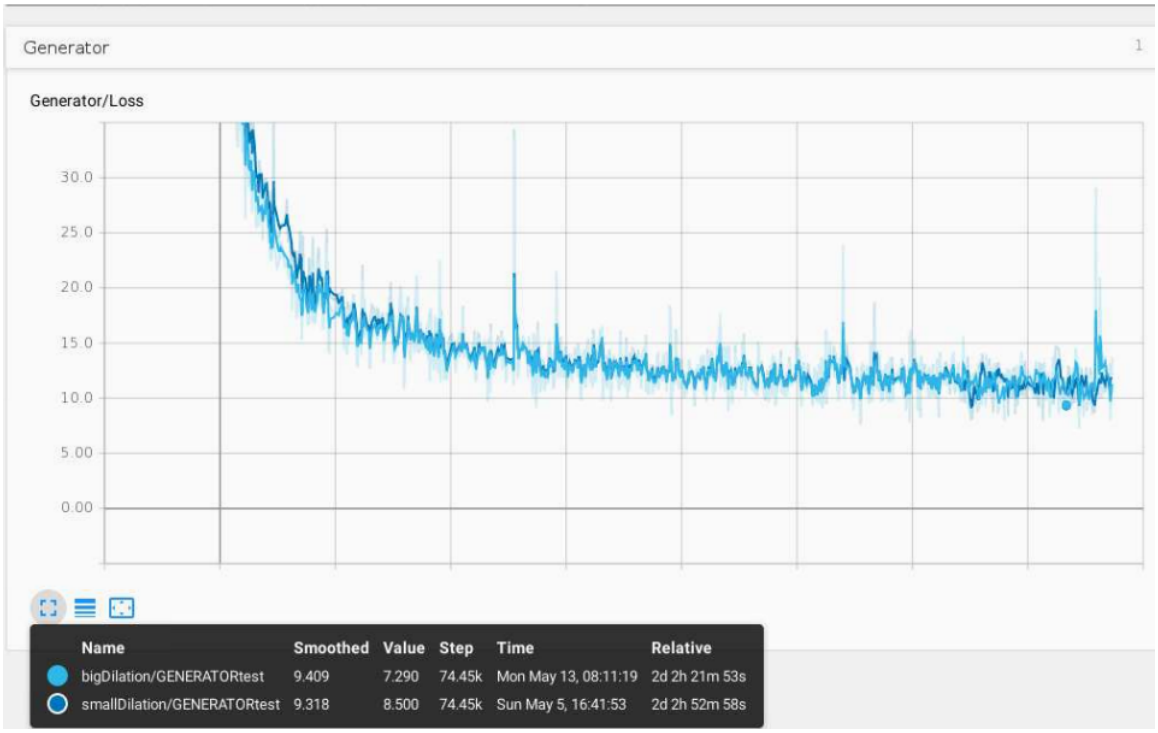


Figure 10: Test error profiles of the Aggregated Conv blocks architecture with different dilation rates.

4.2.5 Classification and Segmentation Architecture:

The training for classification was carried out with a batch size of 200 and a learning rate of 0.00001 for 5 epochs. The semantic segmentation was completed with a learning rate of 0.00001 and batch size of 25 for 100,000 iterations.0 The VGG part of the segmentation model use pre-trained weights from a VGG network trained on imagenetDeng et al. 2009. The hyper-parameters are the same for single, double and triple stream networks for the respective tasks.

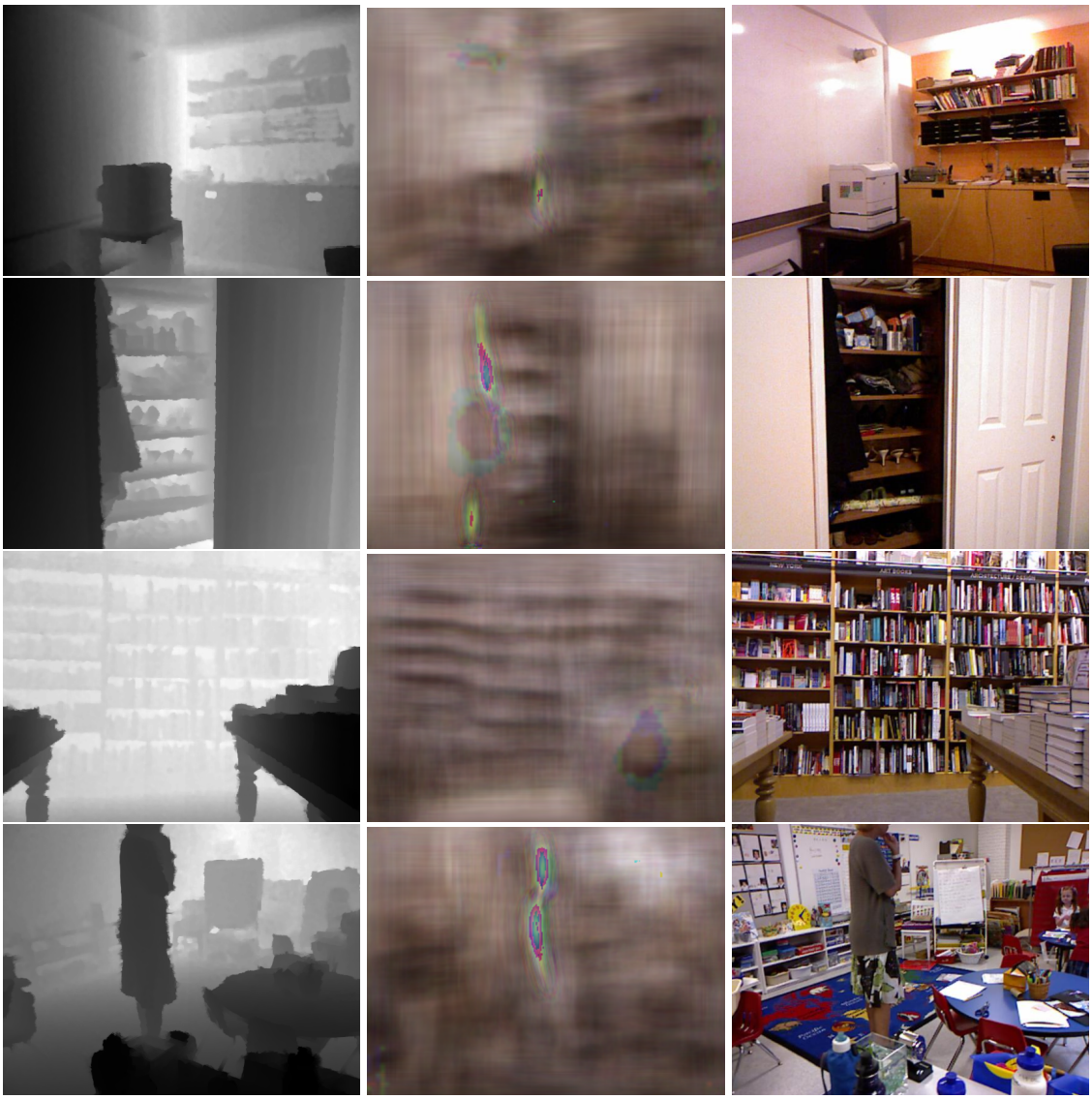
4.3 Result

In this section, we delve further into the experimental results obtained from the hallucination experiments from the different architectures. We also report the results from the different experiments done on various validation experiments mentioned in the above sections. Depth to image transformation is a highly under-constrained process and thus we do not expect the mapping to produce any visually pleasing output. Yet, the networks were able to produce surprisingly good results.

4.3.1 CrossSpectral Hallucination Architecture Results:

Using this fully convolutional architecture from Lezama, Qiu, and Sapiro 2017 produced visually appealing results in the UWRGBD dataset. It did not perform as well with the NYUD dataset. The samples produced using the former dataset are much more pronounced in terms of image content compared to the samples produced by the latter. We ascribe this observation to the fact that the UWRGBD dataset has much lesser inter-class and intra-class variance. The dataset is obtained by placing the object on a turntable and a video sequence is obtained. Results from the UWRGBD dataset can be seen in Fig. **??**

, On the other hand, the NYUD database varies wildly on both inter-class and intra-class level. The NYUD database is a collection of a bunch of scenes such as kitchen, office, bedrooms, etc. An image of the office class will vary wildly with another image of the same class and it will be very different from an image of the kitchen class. This difference in the variance causes the less pronounced NYUD hallucinated images. Despite this difference in the variance we are able to observe the subtle definition of prominent objects. Results of the NYUD hallucination is seen in Fig 11



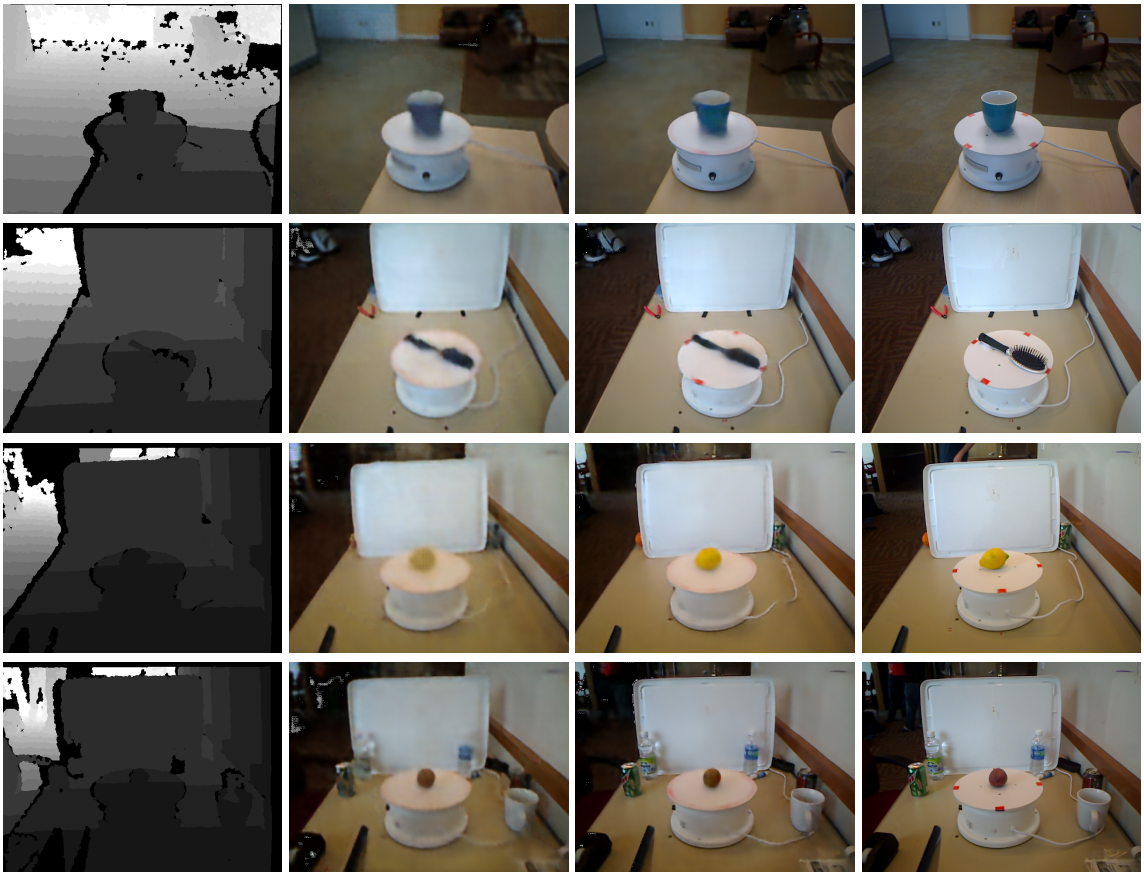
(a) Depth (b) Hallucinated (c) RGB

Figure 11: NYUD dataset hallucination results using the cross spectral hallucination architecture. (a) is the depth input image, (b) is the result of hallucination ,(c) is groundTruth

4.3.2 LinkNet Hallucination results :

The linkNet architecture due to its encoder-decoder architecture that compresses the image with only the important information reconstructs from the smaller dimension space. While with cross spectral architecture the huge number of parameters and the constant size

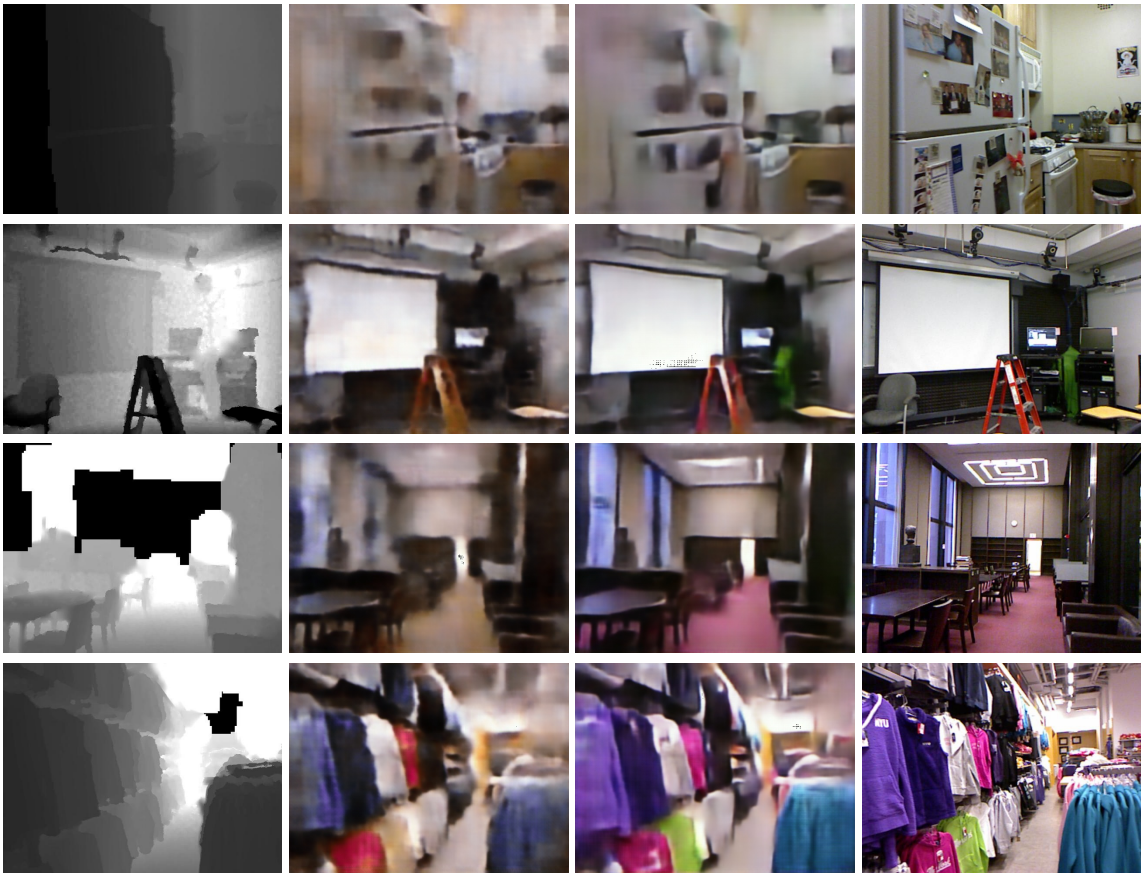
of the feature maps might lead to memorizing or learn on a trivial one-to-one mapping the linkNet architecture does much better job of learning the correct family of functions for the mapping to take place. 12.



(a) Depth (b) Hallucinated (c) Hallucinated* (d) RGB

Figure 12: UWRGBD dataset hallucination results using the linkNet architecture. (a) is the depth input image, (b) is the result of hallucination after the first stage, (c) is the hallucination result after subjecting it to regularizing stage ,(d) is groundTruth

As the UWRGBD is an easy dataset the hallucination does a pretty good job in the first stage itself but we still can find minor improvements after the regularization stage. The NYUD dataset, on the other hand, provides a lot of evidence for the effectiveness of the regularization stage. The results from the NYUD dataset can be seen in Fig.13.



(a) Depth (b) Hallucinated (c) Hallucinated* (d) RGB

Figure 13: NYUD dataset hallucination results using the linkNet architecture. (a) is the depth input image, (b) is the result of hallucination after the first stage, (c) is the hallucination result after subjecting it to regularizing stage ,(d) is groundTruth

4.3.2.1 Regularizer Network Significance

Although the hallucinator network produces convincing RGB renditions from the depth images, the results still seem to display some visible discrepancies between the original RGB and hallucinated data. The hallucinator network seems to be concerned with reproducing the overall structure of the image and doesn't give much importance to color information. Moreover, as the hallucinator network is trained with a weighted smoothness constraint to ensure local smoothness the hallucinator network ignores smaller objects in the RGB

image. The regularizing autoencoder helps to overcome these shortcomings. The results displayed in Fig. 14 and Fig 15 illustrates this ability of two different and datasets of varying complexity. The orange annotations in the image are displayed to understand the improvements the regularizer brings to the results. The regularizer helps to maintain the color information, re-introduce smaller components of the images missed by hallucinator, de-blurs the image and also removes irregularities. As seen in the Fig 14 and Fig 15 the color is better reconstructed by the regularizer. We can see the regularizer is able to reproduce smaller details such as electrical socket on the wall or the Apple symbol on the computer the person has and preserve color information like the color of the wall. This ability of the regularizer makes it a non-trivial part of our experiment. Although the hallucinator with the regularizer does well, it still does poorly when it comes to images that are not from the distribution. If it were to predict the color of a round-ish object which in the training set closely resembles an apple it would predict it as red, but in reality, it could be an orange. The regularizer also has associated drawback if we adopt it into the training procedure. It makes it difficult to be adopted as an online training procedure as the regularizer is trained only after the generator is trained to convergence.

4.3.3 GAN Hallucination Results :

The generative model based hallucination procedure explained in the above sections with the pix2pix architecture also does a relatively good job. It learns the structure of the general overal structure of the objects that occur a lot. For example, in the UWRGBD dataset the GAN is able to produce the structure and the color of the sofa that appears in the background

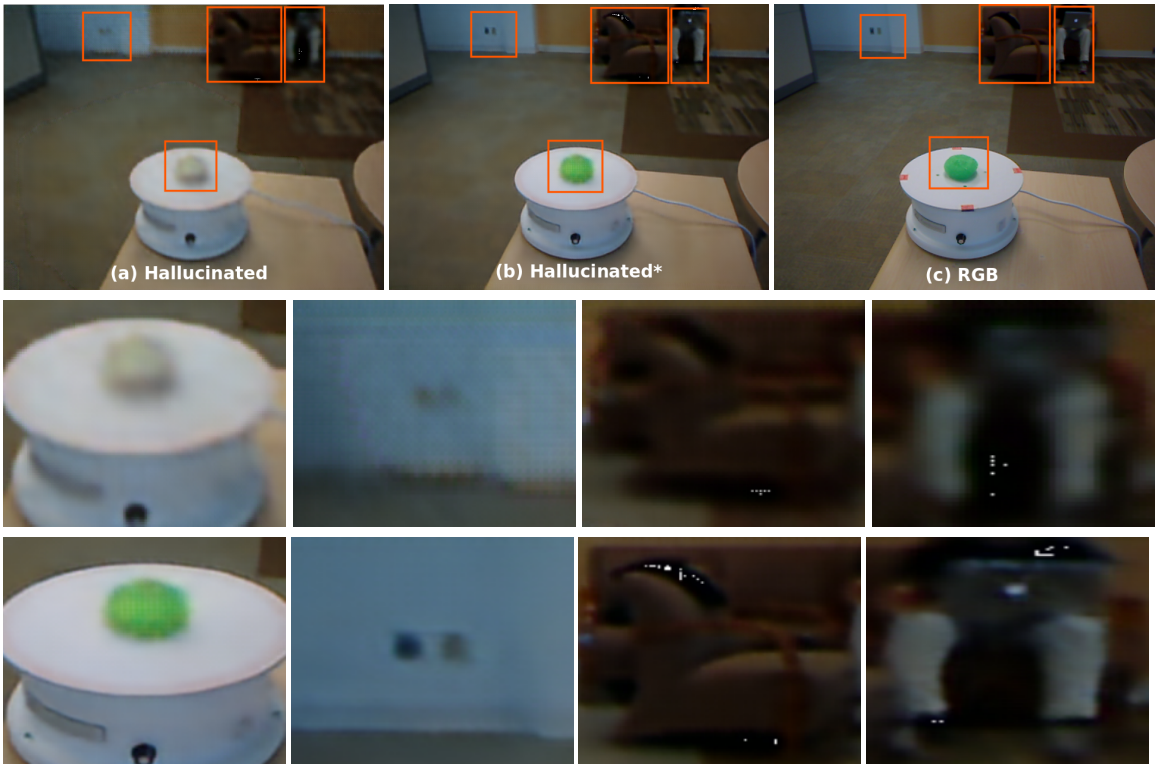


Figure 14: Examples of how the regularizer helps in improving the results in UWRGBD dataset. The first row depicts (from left to right) (a) the hallucinated image without regularization, (b) hallucinated image with regularization, and (c) the ground truth RGB image. The second row are zoomed version of the highlighted areas in the hallucinator (a) output and the third row are zoomed versions of the annotations in the regularizer (hallucinated*) (b) output.

pretty well. It also produces the turn table pretty well as well as the experimental set up of the dataset with the objects like pliers, soda cans etc.

However, when it comes to reconstructing the pixels of the object of interest, that is, object at the center of the turn-table like the apple,banana, it doesn't work as well. The pix2pix network retains a little bit of the structure of the object of interest but it doesn't produce a qualitatively well defined structure as the linkNet architecture. In many cases, it completely misses the color and structure of the object even in a relatively easy dataset like

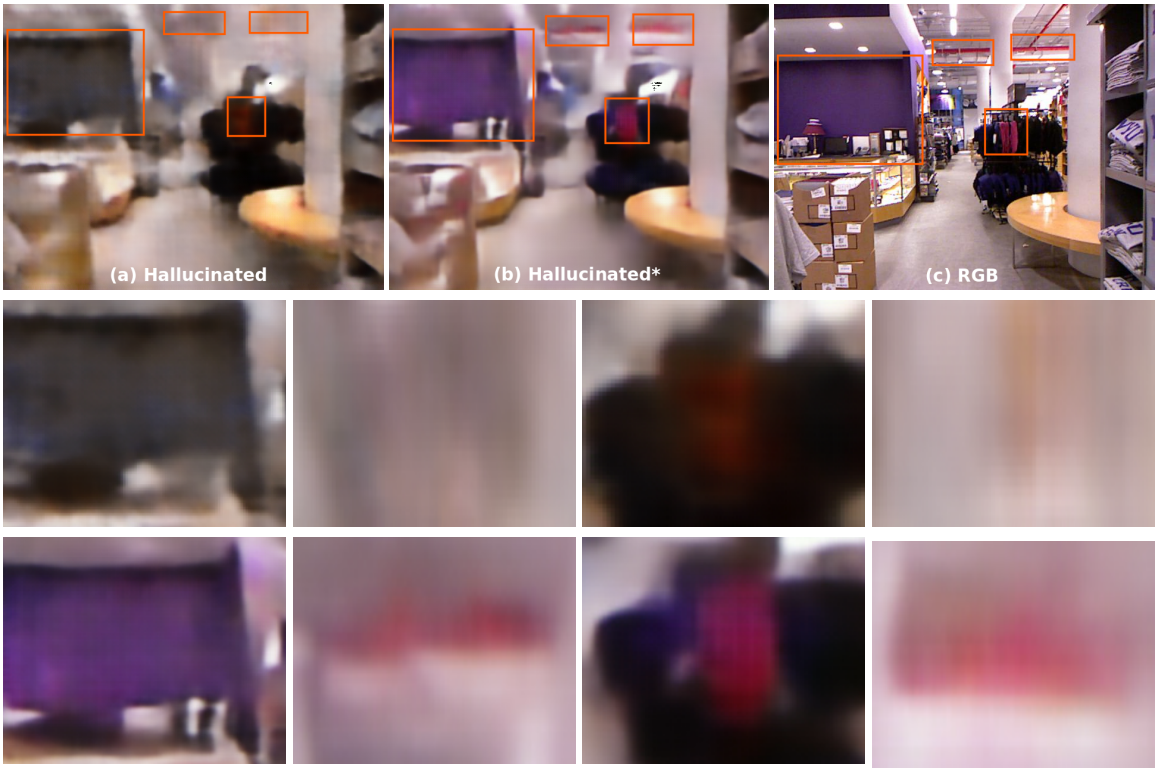
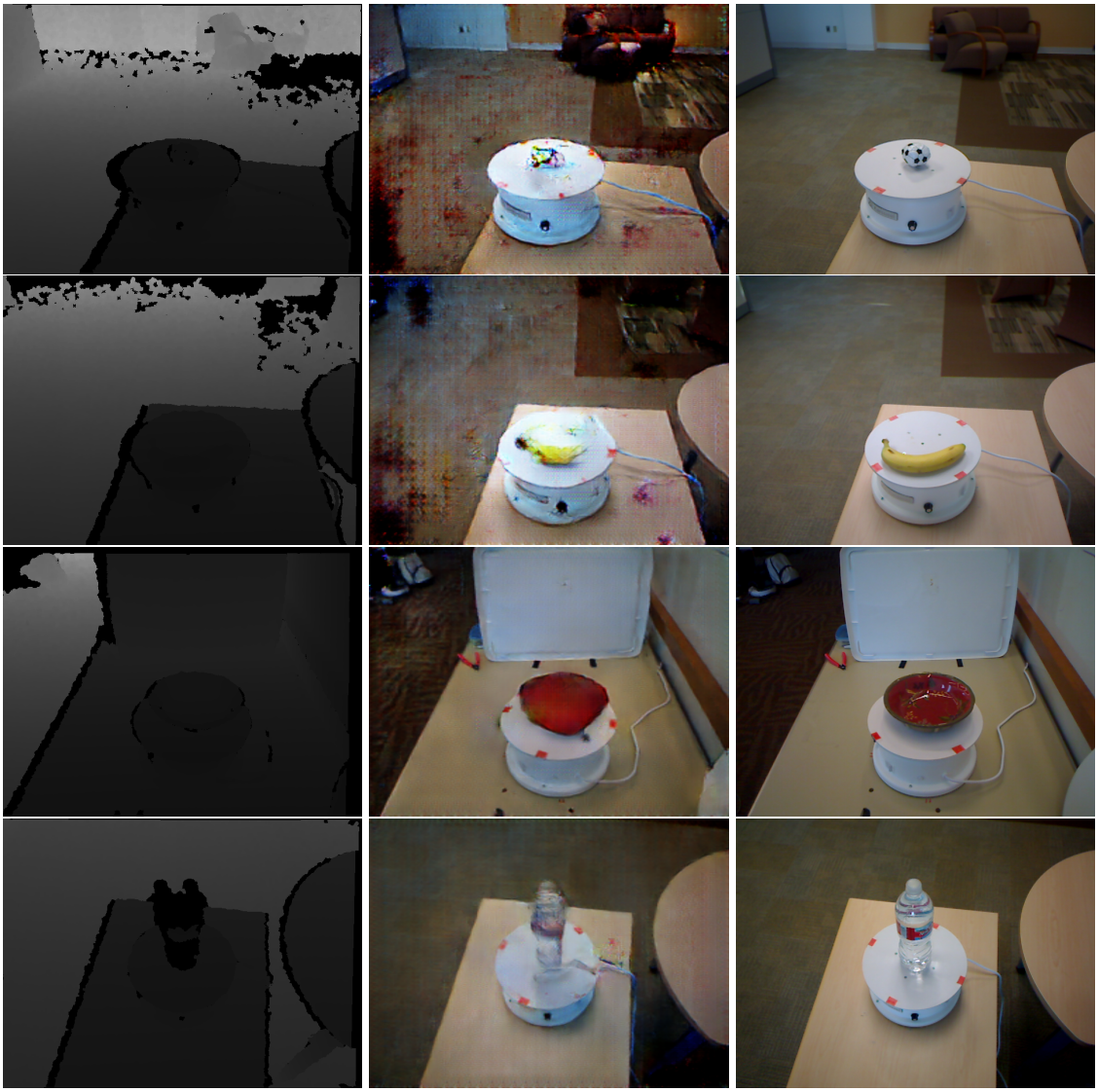


Figure 15: Illustration of regularizer's importance in the NYUD dataset. The first row depicts (from left to right) (a) the hallucinated image without regularization, (b) hallucinated image with regularization, and (c) the ground truth RGB image. The second row are zoomed version of the highlighted areas in the hallucinator (a) output and the third row are zoomed versions of the annotations in the regularizer (hallucinated*) (b) output.

the UWRGBD dataset. Some results from the UWRGBD dataset hallucinated using GANs can be seen in 16.

The same pattern can be observed in the NYUD dataset as well. The generator of the pix2pix produces commonly occurring structures such as the wall, bookshelves pretty well but miss a lot of the finer details and it especially misses some of the colors as well. Moreover, the texture of the image compared to those hallucinated by the Aggregated and linkNet architectures is less accurate. Some examples from the test set of the GAN based hallucination are shown in Fig. 17.



(a) Depth (b) Hallucinated (c) RGB

Figure 16: UWRGBD dataset hallucination results using the GANs (pix2pix architecture). (a) is the depth input image, (b) is the result of hallucination ,(c) is groundTruth

The pix2pix architecture does a pretty bad job when there are multiple objects present in the dataset. The NYUD images are chaotic and have a lot going on in the images which leads to a confused result and visually poor results. In additions to this, GANs have their own disadvantages which include mode collapse, their difficulty in getting the object count right and a major difficulty which is optimizing the GAN loss formulation to convergence.

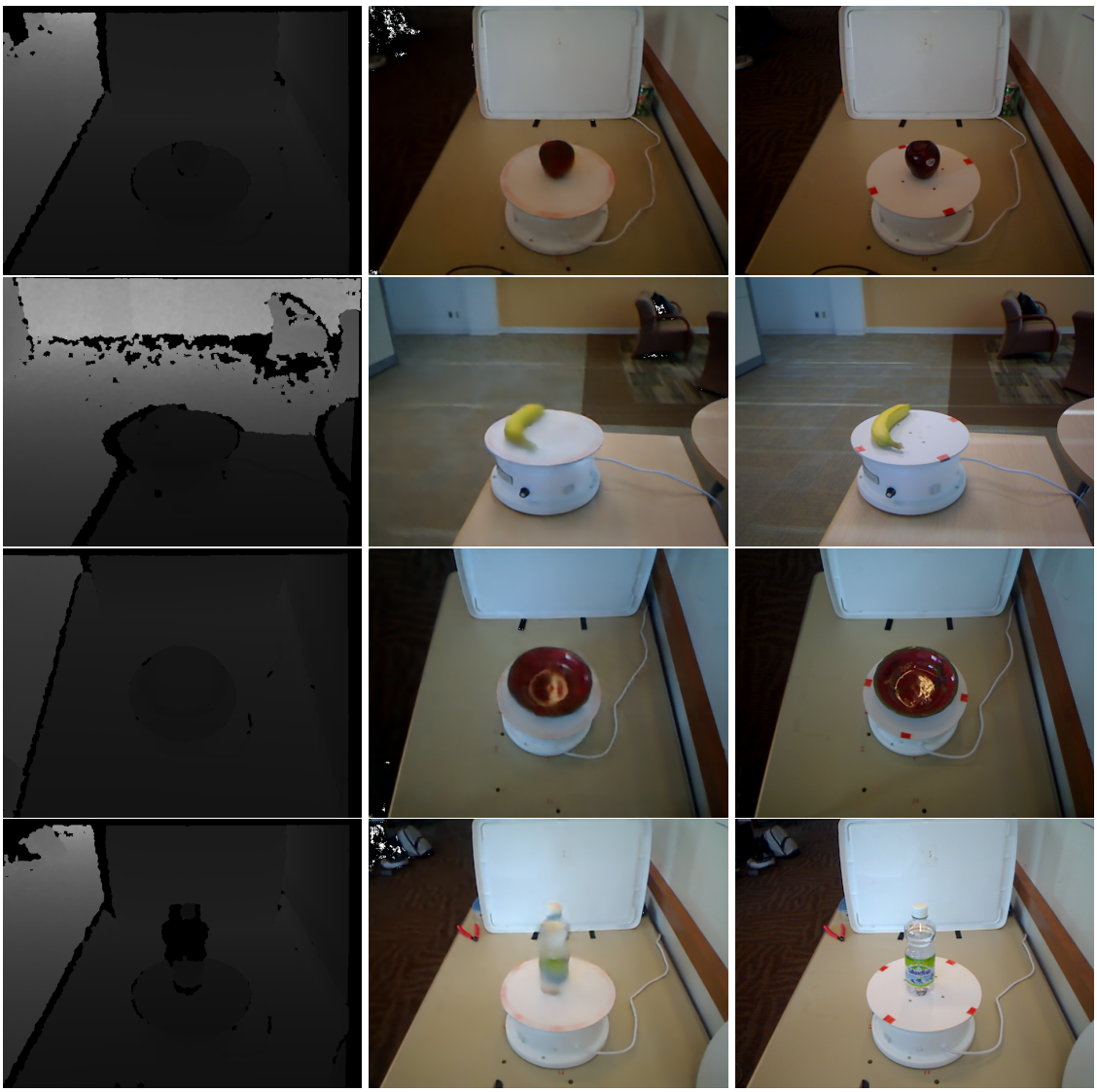
(a) Depth (b) Hallucinated (c) RGB

Figure 17: NYUD dataset hallucination results using the GANs. (a) is the depth input image, (b) is the result of hallucination ,(c) is groundTruth

4.3.4 AggConv Hallucination Results :

Our architecture leverages the knowledge of the local neighborhood by accumulating information of different receptive fields with the help of the aggregating convolutional block. This helps the architecture to reproduce the structure and the color information of the object

using the correlation that exists between the object and its neighbors. The architecture does a pretty good job in both the datasets. The results of hallucination using our architecture on the UWRGBD dataset can be seen in 18.



(a) Depth (b) Hallucinated (c) RGB

Figure 18: UWRGBD dataset hallucination results using our proposed architecture with Aggregated convolutional blocks. (a) is the depth input image, (b) is the result of hallucination ,(c) is groundTruth

The architecture does a solid job on the more difficult NYUD dataset as well. Unlike

GANs it does not get confused with the number of objects and unlike the linkNet architecture, it preserves the color information of the different objects pretty well all the while keeping the artifacts minimum in the generated images. Some, hallucination results of this dataset can be seen in Fig. 19.

(a) Depth (b) Hallucinated (c) RGB

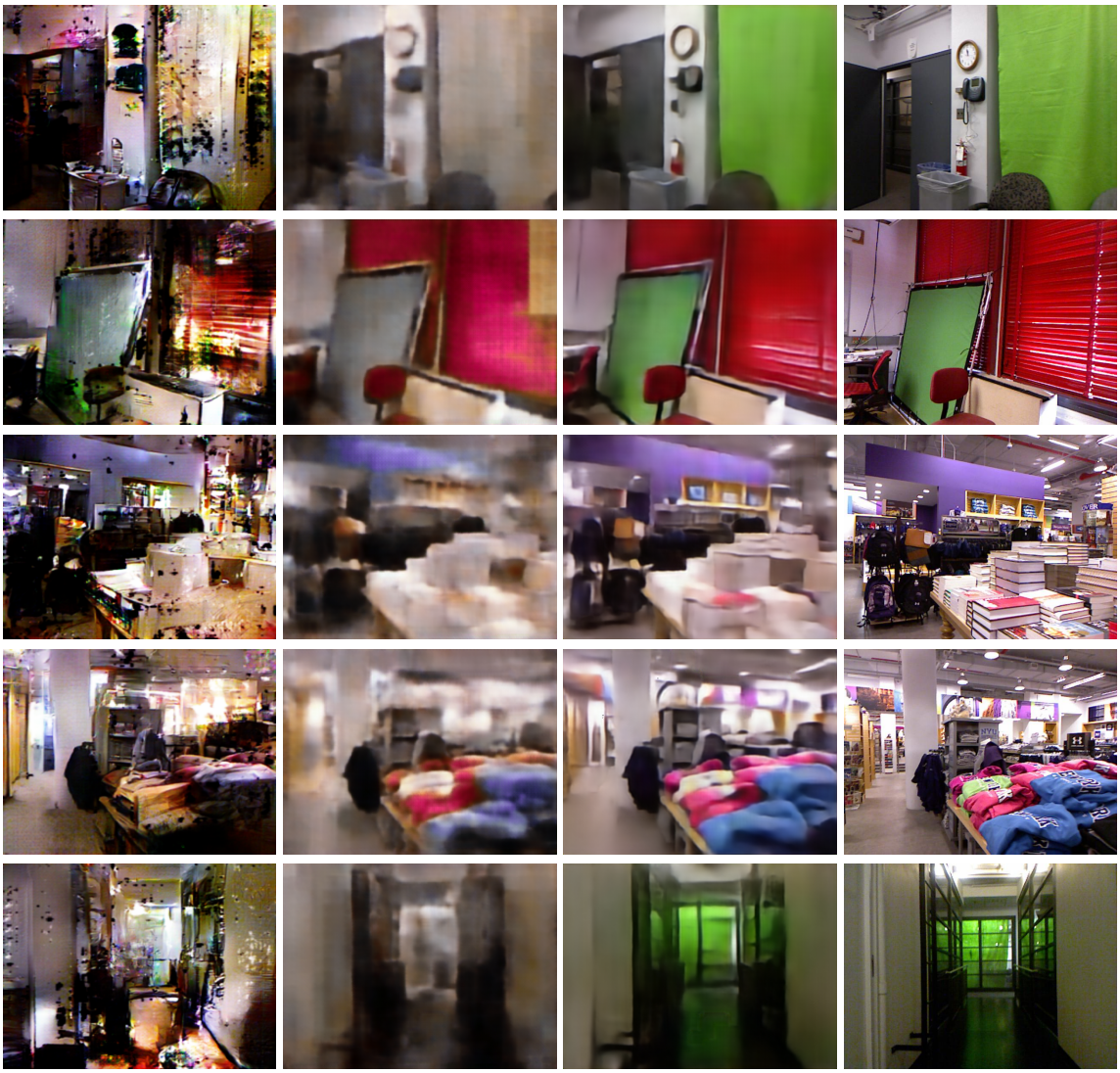
Figure 19: NYUD dataset hallucination results using AggConv blocks (our proposed architecture). (a) is the depth input image, (b) is the result of hallucination ,(c) is groundTruth

The proposed architecture does a nice job on both the datasets. However, the architecture understandably does not produce perfect renditions of the RGB image. It still does miss out on minor details in the image. For instance, in the hallucinated images shown in 18 the hallucination completely misses out on the red markers on the turntable. Information that is very specific to the objects are replaced by the average of what they are, like the text on the water bottle is replaced by the average of the information that is usually present. Hence, such information will appear blurry. This can be seen in the NYUD dataset as well. Sparsely occurring objects in the dataset with specific information of their own are blurry. The architecture again misses the whiteboard on the wall in the first image of Fig. 19 as well as the specular reflections of the light on the board.

4.3.5 A Visual Comparison of the Results with Other Networks.

To have a better perspective of the hallucination results of the different architectures used for this purpose, we present in Fig. 20 a side by side view of the results from the test set depth images. The results shown are some samples from the test set of the NYUD dataset which is the more varied and difficult of the two datasets that were experimented with in this thesis. The ability of our architecture in preserving color as well as maintaining structural integrity is evident from the results displayed below. The GAN does not do an effective job in maintaining the color of the image and it introduces a lot of textural artifacts compared to the other networks. The linkNet architecture, while it rebuilds the structure reasonably well it still misses out the color information and other minor details of the scene and is generally more blurry. The more blurry it is, the more the network is confused with that particular area and replaces it with the average of the likely outcomes. The proposed architecture can be

seen doing pretty well with color as well as structure information preservation. (Over here, we use the 1st stage results of the linkNet to ensure fairness between the different networks)



(a) GAN (b) linkNet (c) AggConv (d) RGB

Figure 20: A visual comparison of the different architectures used as baselines along side our architecture results. (a) results of GAN (pix2pix Hallucination), (b) results of LinkNet architecture, (c) is the hallucination result of our proposed network ,(d) is groundTruth RGB image

The table 5 gives a quantitative perspective about the performance of the different

hallucination networks. The absolute difference between the ground truth the hallucinated results of the test set is reported in the table below. The lower the MAPD the better the performance.

Mean Absolute Pixel Difference		
Dataset	Architecture	*MAPD*
NYUD	GAN	137.57
NYUD	linkNet	10.76
NYUD	AggConv (ours)	**5.96**
UWRGBD	GAN	134.155
UWRGBD	linkNet	3.39
UWRGBD	AggConv (ours)	**2.36**

Table 5: Mean absolute pixel difference indicates how much a pixel in an image deviates on an average from it's true value.

4.4 Using the Hallucinated Modality to mitigate risks:

Substituting the lost modality with hallucination helps: The Table 6 provides evidence to the fact that the hallucinated images indeed captures some of the necessary RGB space information. We conduct the experiments in two settings to show how well the hallucinated data capture RGB space information. In **Setting A** the classification and segmentation networks are trained with RGB data. Then it is tested with all data modalities namely RGB, depth and hallucinated. Over here we to compare the performances of the different networks we include the GAN based hallucinated modality, linkNet based hallucinated modality and our proposed architecture which is AggConv network based hallucinated modality. Since we are trying to capture RGB space information a network trained on RGB data should be able to extract and use features from the hallucinated data and that can indeed be seen in the first section of Table 6. In **Setting B**, networks are trained and tested for each of the

modalities. This is done to show the hallucinated data can be used as a stand-alone modality. While in the classification task it significantly outperforms the depth modality it produces a comparable performance in segmentation task. (In table 6 the RGB column should not be considered while comparing performances as we consider RGB data to be lost. It serves only for reference.)

Hallucinated Modality's Effectiveness					
Task	Object Classification	Semantic Segmentation			
Metric	Total Accuracy	Pixel Accuracy	Mean Accuracy	Mean IoU	Freq. IoU
Setting A					
RGB	96.67%	53.64 %	40.10%	30.13%	44.82 %
Depth	2.12%	18.25 %	12.95 %	4.58 %	10.23 %
GAN	27.48 %	25.84 %	18.20 %	9.69 %	18.04 %
LinkNet	29.19 %	31.87%	19.87 %	11.17 %	22.07 %
AggConv (ours)	**51.14 %**	**35.78%**	**22.22 %**	**13.42%**	**25.33%**
Setting B					
RGB	96.67%	53.64 %	40.10%	30.13%	44.82 %
Depth	51.77 %	**50.53 %**	**35.73 %**	**26.05 %**	**40.67 %**
GAN	80.09 %	36.54 %	22.78 %	13.66 %	26.32 %
LinkNet	82.11 %	47.63%	32.09 %	22.71 %	23.85 %
AggConv (ours)	**91.16 %**	49.84%	34.31 %	24.95%	40.19%

Table 6: The table provides evidence for the effectiveness of using the hallucinated modality in two different settings.

Combining the working modality with the hallucinated modality maintains the overall system's performance: The loss of the primary modality could be anticipated and as a countermeasure, the same task could be trained on other modalities, but that does not ensure good performance. For instance, a pipeline in a self-driving car could be trained for lane detection using the RGB camera data and as a back-up, a network for depth-based detection could be trained in the same way as well. The depth-based system would not perform as well, as that modality is not information-rich like the RGB modality for this task. We believe, in this case, the hallucinated data in combination with the depth data could be better than just having a depth data based back up. This can be seen well depicted in

Table 7. The original system is trained with RGB and depth data. Both classification and segmentation tasks perform much better with hallucinated and depth modalities together than just having depth. There is an increase of approximately 40% classification accuracy and 2.5% mean IoU score for segmentation task which is a significant increase for semantic segmentation. This result validates our claim that data can be hallucinated and be used along with the lower dimensional data to reduce the risk. The performance is comparable to the performance of the original system. Thus in the case of a lost modality, hallucinated data can be helpful. (In Table 7, like in Table 6, RGB + Depth column has been given for reference. It is the original system performance. Since we are considering lost RGB modality the performance comparison is between Depth and hallucinated modality combined with depth modality.)

Hallucinated Modality reduces risk					
Task	Object Classification	Semantic Segmentation			
Metric	Total Accuracy	Pixel Accuracy	Mean Accuracy	Mean IoU	Freq. IoU
RGB + Depth	97.78 %	55.52%	42.30 %	32.08 %	46.60 %
Depth	53.15%	50.53 %	35.73 %	26.05 %	40.67 %
GAN + Depth	86.15 %	52.45 %	37.19 %	27.46 %	42.57 %
LinkNet + Depth	88.01 %	52.03 %	38.15 %	28.02 %	42.33 %
AggConv (ours) + Depth	**92.37 %**	**52.95%**	**38.51 %**	**28.61%**	**43.32%**

Table 7: This table shows the benefits of incorporating the data from the hallucinated modality with the depth modality when the RGB modality is lost. It can be seen that the risk to the system is reduced.

Incorporating the hallucinated modality while all others are working enhances the overall system's performance: An added advantage that we observed from the hallucinated data is that it can be incorporated into the original system to improve the system performance. The hallucinated data captures the space between the depth modality and the RGB modality which could be further leveraged for each task. It could be considered an ensemble of different spaces to make additional gains. Table 8 provides evidence for the same thus proving hallucinated data aides in enhancing the performance existing system.

Both classification and segmentation task benefit from the added modality with segmentation gaining as much as 2.5% on mean IoU score.

Hallucinated Modality Enhances.					
Task	Object Classification	Semantic Segmentation			
Metric	Total Accuracy	Pixel Accuracy	Mean Accuracy	Mean IoU	Freq. IoU
RGB + Depth	97.78 %	55.52%	42.30 %	32.08 %	46.60 %
RGB + Depth + GAN	98.83%	57.45 %	44.41 %	34.35 %	48.83 %
RGB + Depth + LinkNet	97.44 %	57.12 %	44.06 %	33.92 %	48.54 %
RGB + Depth + AggConv (ours)	**98.12 %**	**57.53%**	**44.75 %**	**34.41%**	**48.85%**

Table 8: The hallucinated modality can be incorporated with the fully functioning system to get the ensemble effect and enhance performance further. This table provides evidence for the same.

Chapter 5

LIMITATIONS:

The proposed method is intended for systems that are assumed to be working well and for a reasonable time before the adverse event happening. This is an essential assumption as the training data for the hallucination scheme is generated during the normal working condition on the robot or autonomous system. The more the data from different scenarios, the more the hallucination procedure can generalize. As mentioned before hallucination from low to high dimensional modality is fundamentally ill-posed. To overcome some of the difficulties in this prediction process we utilize the neighborhood information. An object's information that cannot be obtained from the lower dimensional modality such as color, in the depth modality is obtained using the correlation between that information and the neighborhood of the object of interest. This correlation is learned from the explicit relationship that exists in the training data. When this relationship no longer holds, during inference the model still predicts from the relationship that is in memory.

This is better explained with the help of an example. Consider the structure of a sofa represented in the depth image. When hallucinating only from the sofa, the best guess would be to assign the color of the sofa as the average of all the colors it has seen in the training dataset. Thus two similar structured sofas can be incorrectly predicted with the wrong color. The hallucination scheme described by us would look at the sofa as well as the neighborhood. To keep this example intuitive, consider the local neighborhood of the sofa in one instance to have writing desks, chairs, markers, board, etc and another instance of a similar structure sofa to have TV, soda cans, plates, etc. The first instance sofa can be considered as a sofa in a study room while the second instance would be a sofa in the living

room. Now, in the training data, if the study room sofa is predominantly blue and the living room sofa is predominantly red in color, the network can learn to make such associations of its neighborhood to approximate better the color of the sofa when it is predicting. Thus, during inference, if these associations are in accordance with the training data it will predict correctly. But, if that's not the case and during inference, the sofas are switched, it will still predict the colors it is trained on and would wrongly predict the colors which will be a failure case. This we consider as the most important limitation of our method.

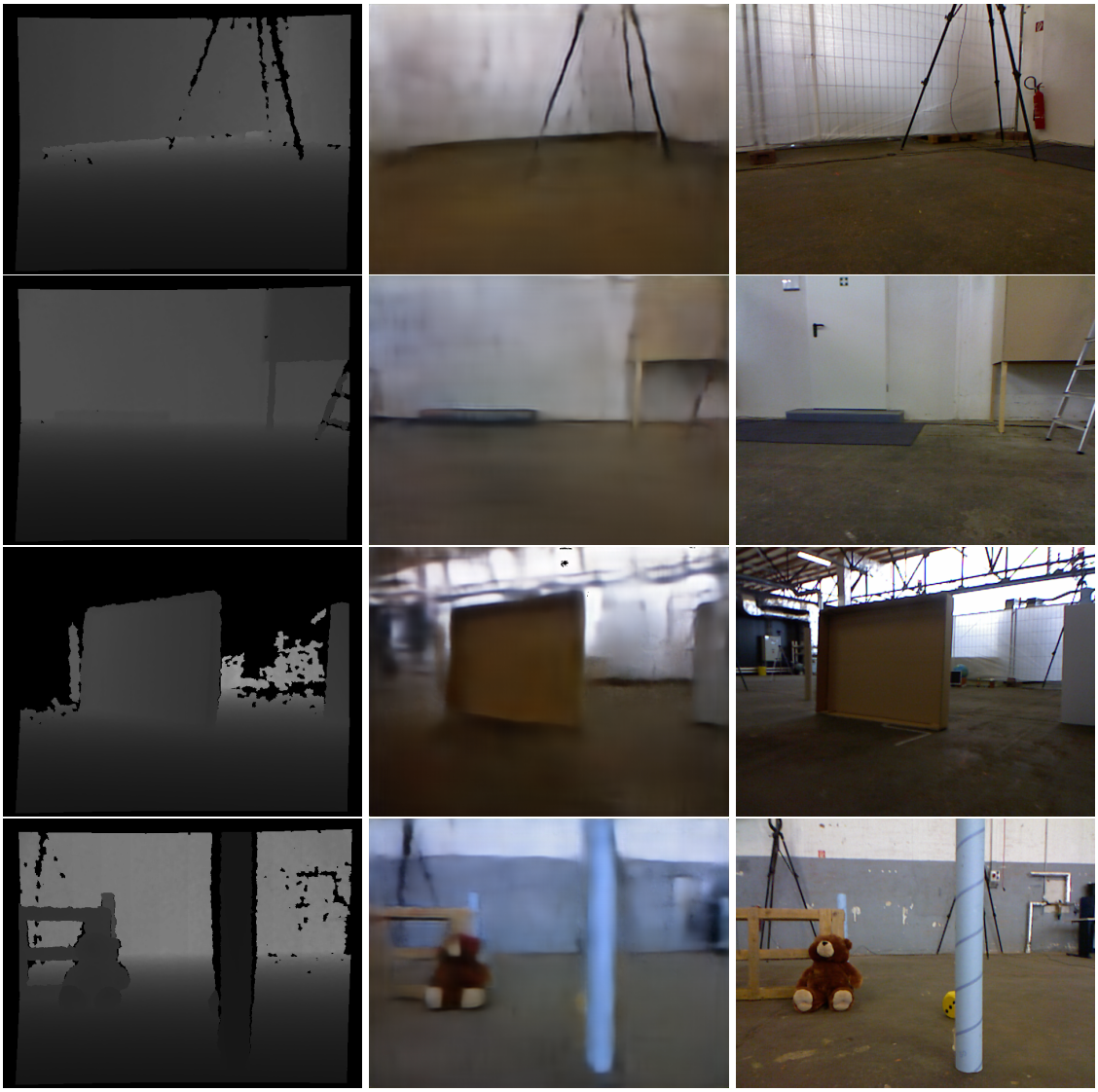
To further put it in an empirical perspective we used the hallucination model that was trained on the NYUD dataset and applied it on the RGBD SLAM dataset from Technical University of Munich Sturm et al. 2012. The results for the same can be seen in Fig. 21. The results are not as good as expected and this comes as a consequence of the limitation explained above. The model that was trained on the NYUD dataset picked on certain relationships between a given pixel and it's neighbors. In the TUM dataset however, this relationship does not hold true and as a result it perform badly. If the hallucinated results are observed a little more closely we can see that the model trained on the NYUD dataset could capture some of the prominent abstractions yet it loses most of the structural integrity as well as color information.

This doesn't mean the hallucination procedure becomes completely useless. In real world an abrupt change in the data distribution is rare and our assumption that the relationship of a pixel with it's neighbors will hold true. Also, this is an easily fixable situation as well. If the data from the other distribution is available as well during normal working conditions they can be incorporated into the training procedure as well and the hallucination model will work good enough to mitigate the risks. This can be seen in Fig. 22. We fine tuned the hallucination model that was trained with the NYUD dataset with data from the TUM

(a) Depth (b) Hallucinated (c) RGB

Figure 21: TUM dataset results using the hallucinator model trained on the NYUD dataset. (a) is the depth input image, (b) is the result of hallucination ,(c) is groundTruth

datasetSturm et al. 2012. In particular we use the dataset under “robot slam” to do this. The sequences “fr2/pioneer360”, “fr2/pioneer_slam” and “fr2/pioneer_slame3” are used as training dataset while “fr2/pioneer_slam2” is used as the testing dataset. The results shown in Fig. 22 are from test set. There are in all 6000 and odd images in the training set and 2000 and odd images in the test set, hence we decided to finetune the NYUD dataset trained



(a) Depth (b) Hallucinated (c) RGB

Figure 22: TUM dataset results after fine tuning the NYUD trained model with the TUM data. (a) is the depth input image, (b) is the result of hallucination ,(c) is groundTruth

model and not train it from scratch. We trained for 8 epochs in a single GPU implementation with a batch size of 7. The rest of the hyper parameters are the same. The results are pretty good. The finetune model preserves the structural integrity as well as most of the color information even with a small dataset thus proving that chnage in data distribution can be farily easily accommodated.

Chapter 6

CONCLUSION

We bring to light the importance of hallucination in multi-modal systems and the challenges in hallucinating from low to high dimension modality. We describe a common adverse scenario in autonomous systems, which is the loss of a data modality and present a method to hallucinate data from the existing modality by capturing a non-linear mapping between the data spaces. We experimented with different state of the art methods for predicting the higher dimensional pixel from the lower dimensional pixels. A novel architecture is also presented that incorporates the information from its neighborhood to make the prediction. A qualitative, as well as quantitative comparison of the experimented methods, were presented and analyzed

We further present evidence that hallucinated data captures the said abstraction, and that it can be used alongside the lower dimensional modality to reduce the adverse effects of the system because of the loss of modality. Moreover, an added advantage of the recovered modality that is observed is also presented with evidence which is, the hallucinated modality can help improve systems gains due to the ensemble effect of including it in the system pipeline. The evidence was provided on two fundamental vision tasks: classification and semantic segmentation on two publicly available and widely adopted benchmarking datasets. We show that our work could be potentially applicable in the fields of assured autonomy, and in improving the reliability of the autonomous and robotics systems with data from multiple modalities. This work finds significance in robotics and autonomous system that require multiple layers of redundancy to ensure reliability in critical situations due to the consequences directed at the system and the environment surrounding it. Although methods

can be devised to use the depth modality to perform better in such adverse scenarios they are limited by the modality-specific constraints. In this work, done by (Luan et al. 2016) the authors deal with the depth modality and its effects on the decision system as a function of the distance of the object in focus. The authors claim that a decision system operating on the near field object does much better and with the same objects in the far field does much worse. That is a classifier is able to easily classify an object near field but struggles when it is at a distance. This introduces further uncertainty into the system that is already trying to overcome that uncertainty. Thus, adopting hallucination schemes can beat those constraints as well, ensuring a safer system and consequently safer environment.

REFERENCES

Atrey, Pradeep K, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. 2010. "Multimodal fusion for multimedia analysis: a survey." *Multimedia systems* 16 (6): 345–379.

Bach-y-Rita, Paul, Mitchell E. Tyler, and Kurt A. Kaczmarek. 2003. "Seeing with the Brain." *International Journal of Human–Computer Interaction* 15 (2): 285–295. doi:10.1207/S15327590IJHC1502_6. eprint: https://doi.org/10.1207/S15327590IJHC1502_6.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. 2017. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." *IEEE transactions on pattern analysis and machine intelligence* 39 (12): 2481–2495.

Castrejon, Lluis, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. "Learning Aligned Cross-Modal Representations from Weakly Aligned Data." In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on.* IEEE.

Chaurasia, Abhishek, and Eugenio Culurciello. 2017. "LinkNet: Exploiting encoder representations for efficient semantic segmentation." In *2017 IEEE Visual Communications and Image Processing, VCIP 2017, St. Petersburg, FL, USA, December 10-13, 2017,* 1–4. doi:10.1109/VCIP.2017.8305148.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE transactions on pattern analysis and machine intelligence* 40 (4): 834–848.

Christoudias, C. Mario, Raquel Urtasun, Mathieu Salzmann, and Trevor Darrell. 2010. "Learning to Recognize Objects from Unseen Modalities." In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I,* 677–691. doi:10.1007/978-3-642-15549-9_49.

Dayan, Peter, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. 1995. "The helmholtz machine." *Neural computation* 7 (5): 889–904.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. "ImageNet: A Large-Scale Hierarchical Image Database." In *CVPR09.*

Eigen, David, Christian Puhrsch, and Rob Fergus. 2014. "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network." In *Advances in Neural Infor-*

mation Processing Systems 27, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, 2366–2374. Curran Associates, Inc. http://papers.nips.cc/paper/5539-depth-map-prediction-from-a-single-image-using-a-multi-scale-deep-network.pdf.

Godard, Clément, Oisin Mac Aodha, and Gabriel J Brostow. 2017. "Unsupervised monocular depth estimation with left-right consistency." In *CVPR,* 2:7. 6.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative adversarial nets." In *Advances in neural information processing systems,* 2672–2680.

Gupta, Saurabh, Judy Hoffman, and Jitendra Malik. 2016. "Cross Modal Distillation for Supervision Transfer." In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016,* 2827–2836. doi:10.1109/CVPR.2016.309.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition." In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 770–778. June. doi:10.1109/CVPR.2016.90.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. "Mask R-CNN." In *Proceedings of the International Conference on Computer Vision (ICCV).*

Hoffman, J., S. Gupta, and T. Darrell. 2016. "Learning with Side Information through Modality Hallucination." In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 826–834. June. doi:10.1109/CVPR.2016.96.

Huang, Xin, Yuxin Peng, and Mingkuan Yuan. 2017. "Cross-modal Common Representation Learning by Hybrid Transfer Network." In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017,* 1893–1900. doi:10.24963/ijcai.2017/263.

Huber, Peter J, et al. 1964. "Robust estimation of a location parameter." *The annals of mathematical statistics* 35 (1): 73–101.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. "Image-to-Image Translation with Conditional Adversarial Networks." In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on.*

Karpathy, Andrej, Armand Joulin, and Fei-Fei Li. 2014. "Deep Fragment Embeddings for Bidirectional Image Sentence Mapping." In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014,*

December 8-13 2014, Montreal, Quebec, Canada, 1889–1897. http://papers.nips.cc/paper/5281-deep-fragment-embeddings-for-bidirectional-image-sentence-mapping.

Karras, Tero, Samuli Laine, and Timo Aila. 2019. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 4401–4410.

Khaleghi, Bahador, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi. 2013. "Multisensor data fusion: A review of the state-of-the-art." *Information fusion* 14 (1): 28–44.

Kingma, Diederik P, and Jimmy Ba. 2014. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980.*

Kingma, Diederik P, and Max Welling. 2013. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114.*

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems 25,* edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, 1097–1105. Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Lahat, Dana, Tülay Adali, and Christian Jutten. 2015. "Multimodal data fusion: an overview of methods, challenges, and prospects." *Proceedings of the IEEE* 103 (9): 1449–1477.

Lai, K., L. Bo, X. Ren, and D. Fox. 2011. "A large-scale hierarchical multi-view RGB-D object dataset." In *2011 IEEE International Conference on Robotics and Automation,* 1817–1824. May. doi:10.1109/ICRA.2011.5980382.

Laina, I., C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. 2016. "Deeper Depth Prediction with Fully Convolutional Residual Networks." In *2016 Fourth International Conference on 3D Vision (3DV),* 239–248. October. doi:10.1109/3DV.2016.32.

Lezama, J., Q. Qiu, and G. Sapiro. 2017. "Not Afraid of the Dark: NIR-VIS Face Recognition via Cross-Spectral Hallucination and Low-Rank Embedding." In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 6807–6816. July. doi:10.1109/CVPR.2017.720.

Li, Dongge, Nevenka Dimitrova, Mingkun Li, and Ishwar K. Sethi. 2003. "Multimedia Content Processing Through Cross-modal Association." In *Proceedings of the Eleventh ACM International Conference on Multimedia,* 604–611. MULTIMEDIA '03. Berkeley, CA, USA: ACM. doi:10.1145/957013.957143.

Luan, Wentao, Yezhou Yang, Cornelia Fermüller, and John S. Baras. 2016. "Reliable Attribute-Based Object Recognition Using High Predictive Value Classifiers." In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III,* 801–815. doi:10.1007/978-3-319-46487-9_49.

Mansimov, Elman, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. 2015. "Generating images from captions with attention." *arXiv preprint arXiv:1511.02793.*

Martı´n Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. https://www.tensorflow.org/.

Paris, Sylvain, and Frédo Durand. 2006. "A fast approximation of the bilateral filter using a signal processing approach." In *European conference on computer vision,* 568–580. Springer.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. "Automatic differentiation in PyTorch."

Ranftl, Rene, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. 2016. "Dense monocular depth estimation in complex dynamic scenes." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 4058–4066.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection." In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 779–788. June. doi:10.1109/CVPR.2016.91.

Reed, Scott E, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. 2016. "Learning what and where to draw." In *Advances in Neural Information Processing Systems,* 217–225.

Reed, Scott, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. "Generative adversarial text to image synthesis." *arXiv preprint arXiv:1605.05396.*

Roe, AW, SL Pallas, YH Kwon, and M Sur. 1992. "Visual projections routed to the auditory pathway in ferrets: receptive fields of visual neurons in primary auditory cortex." *Journal of Neuroscience* 12 (9): 3651–3664. doi:10.1523/JNEUROSCI.12-09-03651.1992. eprint: http://www.jneurosci.org/content/12/9/3651.full.pdf.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. “U-net: Convolutional networks for biomedical image segmentation.” In *International Conference on Medical image computing and computer-assisted intervention,* 234–241. Springer.

Sacks, Ethan. 2018. “Self-driving Uber car involved in fatal accident in Arizona.” Https://www.nbcnews.com/tech/innovation/self-driving-uber-car-involved-fatal-accident-arizona-n857941.

Salvador, Amaia, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. “Learning Cross-modal Embeddings for Cooking Recipes and Food Images.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Shelhamer, Evan, Jonathan Long, and Trevor Darrell. 2017. “Fully Convolutional Networks for Semantic Segmentation.” *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (4): 640–651.

Silberman, N., and R. Fergus. 2011. “Indoor Scene Segmentation using a Structured Light Sensor.” In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition.*

Simonyan, K., and A. Zisserman. 2014. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *CoRR* abs/1409.1556.

Socher, Richard, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. “Zero-Shot Learning Through Cross-Modal Transfer.” In *Advances in Neural Information Processing Systems 26,* edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, 935–943. Curran Associates, Inc. http://papers.nips.cc/paper/5027-zero-shot-learning-through-cross-modal-transfer.pdf.

Srivastava, Nitish, and Ruslan R Salakhutdinov. 2012. “Multimodal Learning with Deep Boltzmann Machines.” In *Advances in Neural Information Processing Systems 25,* edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, 2222–2230. Curran Associates, Inc. http://papers.nips.cc/paper/4683-multimodal-learning-with-deep-boltzmann-machines.pdf.

Sturm, J., N. Engelhard, F. Endres, W. Burgard, and D. Cremers. 2012. “A Benchmark for the Evaluation of RGB-D SLAM Systems.” In *Proc. of the International Conference on Intelligent Robot Systems (IROS).* October.

Sutskever, Ilya, Geoffrey E Hinton, and Graham W Taylor. 2009. “The recurrent temporal restricted boltzmann machine.” In *Advances in neural information processing systems,* 1601–1608.

Szegedy, C., Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. "Going deeper with convolutions." In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1–9. June. doi:10.1109/CVPR.2015.7298594.

Vrečko, A., D. Skočaj, N. Hawes, and A. Leonardis. 2009. "A computer vision integration model for a multi-modal cognitive system." In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 3140–3147. October. doi:10.1109/IROS.2009.5354358.

White, Benjamin W., Frank A. Saunders, Lawrence Scadden, Paul Bach-Y-Rita, and Carter C. Collins. 1970. "Seeing with the skin." *Perception & Psychophysics* 7, no. 1 (January). doi:10.3758/BF03210126.

Xu, Dan, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. 2017. "Learning Cross-Modal Deep Representations for Robust Pedestrian Detection." In *CVPR,* 4236–4244. IEEE Computer Society.

Xu, Dan, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. 2015. "Learning Deep Representations of Appearance and Motion for Anomalous Event Detection." In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015,* 8.1–8.12. doi:10.5244/C.29.8.

Zhang, Han, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks." In *Proceedings of the IEEE International Conference on Computer Vision,* 5907–5915.

Zhao, Hengshuang, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. "Pyramid scene parsing network." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2881–2890.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss." In *Computer Vision (ICCV), 2017 IEEE International Conference on.*