ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc. THESIS

WASAN THAKER NADHIM NADHIM

DEPTH FROM BLUR

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2022

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPTH FROM BLUR

WASAN THAKER NADHIM NADHIM

MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved the award of degree of the Master of Science by the board of jury on 25/04/2022

……..	…….....	……..……....
Assoc.Prof.Dr. Mustafa ORAL SUPERVISOR	Assoc. Prof. Dr. Zekeriya TÜFEKÇİ MEMBER	Asst. Prof. Dr. Mümine KAYA KELEŞ MEMBER

This MSc thesis is written at the Computer Engineering Department of the Institute of Natural and Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Sadık DİNÇER
Director
Institute of Natural and Applied Science

ABSTRACT

MSc. THESIS

DEPTH FROM BLUR

WASAN THAKER NADHIM NADHIM

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc.Prof. Dr. Mustafa ORAL
Year: 2022, Pages:101
Jury : Assoc.Prof. Dr. Mustafa ORAL
: Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
: Asst. Prof. Dr. Mümine KAYA KELEŞ

The most accessible and appropriate approach to recording and storing the depth measurements collected from a scene is through a depth map; accurate depth maps are also essential in extended reality and movie production. This topic of study is both intriguing and beneficial, and lucrative. Several firms are developing depth estimates for a range of reasons. Some may use it to add effects (such as bokeh) to photos and selfies (portrait mode) based on distance from the camera; others, both aerial and terrestrial, may utilize it for replacing or supplementing existing sensors in autonomous vehicles. A depth map is a two-dimensional array with the x and y distance information corresponding to the array's rows and columns, as in a conventional picture. This study examines the single-image depth inference problem using focus and blur images. A comparative work that examines Carvalho's, Lee's and Laina's methods that produce a depth map from a single image is carried out in this thesis. Carvalho's method takes a single synthetic to defocus image as input, and the output is the depth map using D3-Net. Both Lee's and Laina's approaches build a depth map from a single image using encoder-decoder architecture and Residual Network (ResNet50), respectively. After that, the predicted depth maps were segmented into three classes (near, far, far away). This study aims to determine the best performing method for estimating the depth map using the New York University v2 dataset. Furthermore, we can use the segmentation results to navigate the cameras (drones, robots, autonomous vehicles, etc.). As the results show, Carvalho's method was the best in-depth map estimation because of the synthetic defocus image dataset. Nevertheless, Laina's method is the best segmentation in near and far areas. The experimental results demonstrate that the NYU v2 dataset used with these models achieved accuracy values of 99.8%, 99.0%, and 98.8% for Carvalho, Laina, and Lee, respectively, for the predicted depth map. For segmentation, the accuracy values were 77%, 55%, and 90% for Carvalho, Lee, and Laina, respectively.

Keywords: Depth map, Blur, Recurrent Neural Network, Convolutional Neural Network, Deep Learning.

ÖZ

YÜKSEK LİSANS TEZİ

BULANIKLIKTAN DERİNLİK ÇIKARIMI

WASAN THAKER NADHIM NADHIM

ÇUKUROVA UNIVERSITY
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç.Dr. Mustafa ORAL
Yıl: 2022, Sayfa: 101
Jüri : Doç.Dr. Mustafa ORAL
: Doç. Dr. Zekeriya TÜFEKÇİ
: Dr. Öğr. Üyesi Mümine KAYA KELEŞ

Bir sahneden toplanan derinlik ölçümlerini kaydetmeye ve saklamaya yönelik en erişilebilir ve uygun yaklaşım bir derinlik haritasıdır; doğru derinlik haritaları, genişletilmiş gerçeklik ve film prodüksiyonunda da önemlidir. Bu çalışma konusu hem ilgi çekici hem de faydalı ve kazançlı. Birkaç firma, çeşitli nedenlerle derinlik tahminleri geliştirmektedir. Bazıları bunu, kameradan uzaklığa bağlı olarak fotoğraflara ve özçekimlere (portre modu) efektler (bokeh gibi) eklemek için kullanabilir; hem havadan hem de karadan diğerleri, otonom araçlardaki mevcut sensörleri değiştirmek veya desteklemek için kullanabilir. Derinlik haritası, geleneksel bir resimde olduğu gibi, dizinin satırlarına ve sütunlarına karşılık gelen x ve y mesafe bilgilerine sahip iki boyutlu bir dizidir. Bu çalışma, odak ve bulanık görüntüleri kullanarak tek görüntülü derinlik çıkarsama problemini incelemektedir. Bu tezde, Carvalho'nun, Lee'nin ve Laina'nın tek bir görüntüden derinlik haritası üreten yöntemlerini inceleyen karşılaştırmalı bir çalışma yapılmıştır. Carvalho'nun yöntemi, girdi olarak tek bir sentetik odaktan bulanık görüntü alır ve çıktı, D3-Net kullanan derinlik haritası görüntüsüdür. Hem Lee'nin hem de Laina'nın yaklaşımları, sırasıyla kodlayıcı-kod çözücü mimarisini ve Kalıntı Ağı (ResNet50) kullanarak tek bir görüntüden bir derinlik haritası oluşturur. Daha sonra tahmin edilen derinlik haritaları üç sınıfa (yakın, uzak, uzak) ayrılmıştır. Bu çalışma, New York University v2 veri setini kullanarak derinlik haritasını tahmin etmek için en iyi performans gösteren yöntemi belirlemeyi amaçlamaktadır. Ayrıca, segmentasyon sonuçlarını kameralarda gezinmek için kullanabiliriz (drone'lar, robotlar, otonom araçlar vb.). Sonuçların gösterdiği gibi, Carvalho'nun yöntemi, sentetik defokus görüntü veri seti nedeniyle en iyi derinlemesine harita tahminiydi. Yine de, Laina'nın yöntemi yakın ve uzak alanlarda en iyi bölütlemedir. Deneysel sonuçlar, bu modellerle birlikte kullanılan NYU v2 veri setinin, tahmin edilen derinlik haritası için sırasıyla Carvalho, Laina ve Lee için %99,8, %99,0 ve %98,8 doğruluk değerlerine ulaştığını göstermektedir. Segmentasyon için doğruluk değerleri Carvalho, Lee ve Laina için sırasıyla %77, %55 ve %90 idi..

Anahtar Kelimeler: Derinlik haritası, Bulanıklık, Evrişimsel Sinir Ağı, Tekrarlayan Sinir Ağı, Derin Öğrenme

EXTENDED ABSTRACT

In classic image acquisition systems, depth information is still missing. Because of this, several methods have been presented to carry out this lost depth. The depth map estimation procedure restores the third dimension from a two-dimensional picture, i.e. the distance between each pixel in a scene.

Depth estimation marks figuring out the distance and structure of objects in a picture using either single or multiple views, where single-view or multi-view images are employed in computer vision. To model or rebuild 3D shapes, depth maps are used. 3D scanners can create depth maps, or they can be reconstructed from multiple images. To allow 2D image tools to process 3D images in machine vision and computer vision.

Accurate depth maps are essential in extended reality and movie production. For a variety of reasons, several companies are creating depth estimations. Others may utilize it for replacing or supplementing other detectors in independent automobiles, both aerial and terrestrial, to add effects (such as bokeh) to images and selfies (portrait mode) based on distance from the camera.

Using depth maps, simulating the effect of consistently thick semi-transparent material inside a picture - such as fog or vast quantities of water. Acting small depths of field, which appear to have certain sections of a picture out of focus.

The most challenging job in two-dimension to three-dimension (2D-to-3D) image transformation algorithms is to elicit depth maps from a single image. This is due to the ambiguity in the depth of images and noise, which may also contain blurring.

Many image transformation techniques are available; besides the remarkable development in the last few years, some holes in the 2D-to-3D transformation algorithms still exist. The most successful methods need human intervention and, thus, are expensive and time-consuming. Hence, many techniques have been developed which automatically estimate the depth using single or multiple images.

This thesis aims to examine three existing architectures for estimating the depth map and then segment the predicted depth map (by using threshold) to see which can be relied upon for estimating depth classes. Furthermore aims to use the results of the segmentation process in the cameras used in drones, robots, autonomous vehicles, etc.

In this study, three different techniques were compared; Carvalho's, Lee's, and Laina's methods used to estimate the depth map information; we also looked at how the blur affects depth estimation. The most current deep learning-based algorithms for depth estimation use geometrical structures of typical crisp pictures to predict related depth maps.

Cameras may create images with defocus blur based on the depth of the objects and the camera settings. As a result, these characteristics might be a crucial clue for learning to estimate depth.

Carvalho proposed a whole system for single-image depth prediction based on these indications, in which the impact of blur on depth prediction is investigated using a neural network technique. Carvalho investigated the effect of defocus blur on depth estimation using D3-Net, a dense deep defocus neural network. By putting it to the test on a synthetically defocused dataset derived from NYUv2, including optically realistic blur fluctuation.

These experiments reveal that out-of-focus blur significantly enhances the performance of depth-prediction networks, which is a significant suggestion for improving deep depth map estimate.

Furthermore, Carvalho's solution overcomes the present limits of single-image DFD by combining structural and blur information. These findings may be employed in a specialized device with genuine defocus blur to accurately forecast depth inside with good generalization.

Using relative depth maps, Lee proposed a new approach for estimating monocular depth. Lee first assessed relative depths between pairs of areas and conventional depths at various scales using a convolutional neural network. Second,

Lee recovered relative depth maps from selectively estimated data using the rank-1 property of pairwise comparison matrices. Third, ordinary and relative depth maps were disassembled and reassembled to form a final depth map.

Lee's method delivers acceptable and blurred performance, according to the results of the experiments. Lee used a network that contains an encoder-decoder.

The encoder part (Denese-Net BC) extracts deep features from an image to provide low-resolutions, high-level features. The decoder part contains up to 10 decoders that use the features from the encoder part to reconstruct ordinary and relative depth maps.

We can see the advantage of Lee's method when each depth in an ordinary depth map is compared with a larger region for a fixed number of comparisons.

Laina proposed a fully convolutional architecture with residual learning to explain the unclear mapping between monocular visuals and depth maps. Laina devised a new technique for learning feature map up-sampling within the network in order to improve result resolution.

Laina optimized using the reverse Huber loss, which is proper to the task and driven by the value distributions of standard, in-depth maps.

The model comprises a single architecture trained from start to finish without using post-processing approaches like CRFs or other improvement stages.

As a consequence, it can work with photos or videos in real-time. Laina used up-sampling blocks to replace the fully linked layer.

Laina picked ResNet because it allowed to build much deeper networks without deterioration or disappearing gradients, and the huge respective field of ResNet50 allows for high input inputs.

The initial half of the Laina network is based on ResNet 50, while the second part of the Laina network guides the network through a series of unpooling and convolutional layers to learn upsampling.

In comparison to the current state of the art, the model has rare parameters and needs less training data, and it performed well in trials.

We carried out experiments using defocused synthetic and focused images on examining the power of deep learning for deep depth prediction.

Experiments test deep convolutional neural networks(D3Net), recurrent neural networks(Encoder-Decoder), and residual networks(ResNet50) for depth map prediction on the NYU v2 dataset.

This study shows that out-of-focus blur significantly enhances the depth map prediction result.

After predicting the depth map from the three methods, we segmented the result depth map into three classes (near, far, faraway), to find out which methods are best in determining the faraway, near and far-distance pixels from the camera.

Depth map prediction and segmentation performance have to be made comparatively by using five and three evaluations metrics, respectively.

Furthermore, the predicted depth maps of each method have been visually examined and discussed. Five different metrics (MSE, RNS, Log RMS, Rel, Accuracy) measure each method's performance in the quantitative evaluation. According to these evaluations, it has been observed that Carvalho's method produces a successful depth map and outperforms the other techniques in depth map prediction. Laina's method gets good results with some noise. Lee's method gets a reliable result with apparent blur.

The testing results show that the NYU v2 dataset used with these models achieved accuracy values of 99.8%, 99.0%, and 98.8% for Carvalho, Laina, and Lee, respectively.

Nevertheless, measuring each method's performance depends on quantitative comparison using three metrics (Pixel accuracy, IoU, DC) in the segmentation process.

It has been proved that Laina's method produces the best depth map classes just in the near and far areas, whereas Carvalho's method was the best in the faraway area. The accuracy values were 77%, 55%, 90% for Carvalho, Lee, and Laina, respectively.

GENİŞLETİLMİŞ ÖZET

Klasik görüntü alma sistemlerinde derinlik bilgisi hala eksiktir. Bu nedenle, kaybedilen bu derinliği gerçekleştirmek için çeşitli yöntemler sunulmuştur. Derinlik haritası tahmin prosedürü, iki boyutlu bir resimden üçüncü boyutu, yani bir sahnedeki her piksel arasındaki mesafeyi geri yükler.

Derinlik tahmin işaretleri, tekli veya çoklu görünümler kullanarak bir resimdeki nesnelerin uzaklığını ve yapısını belirleyen, bilgisayarla görmede tekli veya çok yönlü görüntülerin kullanıldığı yerlerde. 3B şekilleri modellemek veya yeniden oluşturmak için derinlik haritaları kullanılır. 3D tarayıcılar derinlik haritaları oluşturabilir veya birden fazla görüntüden yeniden oluşturulabilir. 2B görüntü araçlarının 3B görüntüleri yapay görme ve bilgisayarla görmede işlemesine izin vermek.

Genişletilmiş gerçeklik ve film prodüksiyonunda doğru derinlik haritaları çok önemlidir. Çeşitli nedenlerle, birkaç şirket derinlik tahminleri oluşturuyor. Diğerleri, kameradan uzaklığa bağlı olarak görüntülere ve özçekimlere (portre modu) efektler (bokeh gibi) eklemek için bağımsız otomobillerdeki diğer dedektörleri değiştirmek veya desteklemek için kullanabilir.

Sis veya çok miktarda su gibi, bir resmin içindeki sürekli kalın yarı saydam malzemenin etkisini simüle eden derinlik haritalarını kullanmak. Bir resmin belirli bölümlerinin odak dışında olduğu görünen küçük alan derinlikleri. İki boyutludan üç boyutluya (2D'den 3D'ye) görüntü dönüştürme algoritmalarında en zorlu iş, tek bir görüntüden derinlik haritaları ortaya çıkarmaktır. Bunun nedeni, bulanıklığı da içerebilen görüntülerin ve gürültünün derinliğindeki belirsizliktir.

Birçok görüntü dönüştürme tekniği mevcuttur; Son birkaç yıldaki kayda değer gelişmenin yanı sıra, 2B'den 3B'ye dönüştürme algoritmalarında bazı boşluklar hala mevcuttur. En başarılı yöntemler insan müdahalesine ihtiyaç duyar ve bu nedenle pahalı ve zaman alıcıdır. Bu nedenle, tek veya çoklu görüntüleri kullanarak derinliği otomatik olarak tahmin eden birçok teknik geliştirilmiştir.

Bu tez, derinlik haritasını tahmin etmek için mevcut üç mimariyi incelemeyi ve daha sonra derinlik sınıflarını tahmin etmek için hangisine güvenilebileceğini görmek için tahmin edilen derinlik haritasını (eşik kullanarak) bölümlere ayırmayı amaçlamaktadır. Ayrıca drone, robot, otonom araç vb. araçlarda kullanılan kameralarda segmentasyon işleminin sonuçlarını kullanmayı hedefliyor.

Bu çalışmada üç farklı teknik karşılaştırılmıştır; Derinlik haritası bilgilerini tahmin etmek için kullanılan Carvalho's, Lee's ve Laina's yöntemleri; bulanıklığın derinlik tahminini nasıl etkilediğine de baktık. Derinlik tahmini için en güncel derin öğrenme tabanlı algoritmalar, ilgili derinlik haritalarını tahmin etmek için tipik net resimlerin geometrik yapılarını kullanır.

Kameralar, nesnelerin derinliğine ve kamera ayarlarına bağlı olarak odak bulanıklığı olan görüntüler oluşturabilir. Sonuç olarak, bu özellikler derinliği tahmin etmeyi öğrenmek için çok önemli bir ipucu olabilir.

Carvalho, bulanıklığın derinlik tahmini üzerindeki etkisinin bir sinir ağı tekniği kullanılarak araştırıldığı bu göstergelere dayalı olarak tek görüntülü derinlik tahmini için bütün bir sistem önerdi. Carvalho, yoğun bir derin odaklanma sinir ağı olan D3-Net'i kullanarak odak bulanıklığının derinlik tahmini üzerindeki etkisini araştırdı. Optik olarak gerçekçi bulanıklık dalgalanması da dahil olmak üzere NYUv2'den türetilen sentetik olarak odaklanmamış bir veri kümesi üzerinde teste tabi tutularak. Bu deneyler, odak dışı bulanıklığın, derin derinlik haritası tahminini iyileştirmek için önemli bir öneri olan derinlik tahmini ağlarının performansını önemli ölçüde artırdığını ortaya koymaktadır.

Ayrıca, Carvalho'nun çözümü, yapısal ve bulanıklık bilgilerini birleştirerek tek görüntülü DFD'nin mevcut sınırlarının üstesinden gelir. Bu bulgular, iyi bir genelleme ile içerideki derinliği doğru bir şekilde tahmin etmek için gerçek odak bulanıklığına sahip özel bir cihazda kullanılabilir.

Göreceli derinlik haritalarını kullanan Lee, monoküler derinliği tahmin etmek için yeni bir yaklaşım önerdi. Lee, önce bir evrişimsel sinir ağı kullanarak çeşitli ölçeklerde alan çiftleri ve geleneksel derinlikler arasındaki bağıl derinlikleri

değerlendirdi. İkinci olarak, Lee, ikili karşılaştırma matrislerinin sıra-1 özelliğini kullanarak seçici olarak tahmin edilen verilerden göreceli derinlik haritalarını kurtardı. Üçüncüsü, sıradan ve göreceli derinlik haritaları demonte edildi ve nihai bir derinlik haritası oluşturmak için yeniden birleştirildi.

Lee'nin yöntemi, deneylerin sonuçlarına göre kabul edilebilir ve bulanık bir performans sunar. Lee, kodlayıcı-kod çözücü içeren bir ağ kullandı.

Kodlayıcı kısmı (Denese-Net BC), düşük çözünürlüklü, yüksek seviyeli özellikler sağlamak için bir görüntüden derin özellikleri çıkarır. Kod çözücü parçası, sıradan ve göreli derinlik haritalarını yeniden oluşturmak için kodlayıcı bölümündeki özellikleri kullanan en fazla 10 kod çözücü içerir.

Sıradan bir derinlik haritasındaki her derinlik, sabit sayıda karşılaştırma için daha büyük bir bölgeyle karşılaştırıldığında, Lee'nin yönteminin avantajını görebiliriz.

Laina, monoküler görseller ve derinlik haritaları arasındaki belirsiz haritalamayı açıklamak için artık öğrenme ile tamamen evrişimsel bir mimari önerdi. Laina, sonuç çözünürlüğünü iyileştirmek için ağ içinde özellik haritası yukarı örneklemeyi öğrenmek için yeni bir teknik tasarladı. Laina, göreve uygun olan ve standart, derinlemesine haritaların değer dağılımları tarafından yönlendirilen ters Huber kaybını kullanarak optimize etti.

Model, CRF'ler veya diğer iyileştirme aşamaları gibi işlem sonrası yaklaşımlar kullanılmadan baştan sona eğitilmiş tek bir mimariden oluşur.

Sonuç olarak, gerçek zamanlı olarak fotoğraf veya videolarla çalışabilir. Laina, tamamen bağlantılı katmanı değiştirmek için yukarı örnekleme blokları kullandı. Laina, ResNet'i seçti, çünkü bozulma veya gradyanlar kaybolmadan çok daha derin ağlar kurmaya izin verdi ve ResNet50'nin büyük ilgili alanı, yüksek girdi girdilerine izin verdi. Laina ağının ilk yarısı ResNet 50'ye dayanırken, Laina ağının ikinci kısmı, örneklemeyi öğrenmek için bir dizi havuzdan çıkarma ve evrişimsel katman aracılığıyla ağa rehberlik eder. Tekniğin mevcut durumuna kıyasla, model

nadir parametrelere sahiptir ve daha az eğitim verisine ihtiyaç duyar ve denemelerde iyi performans gösterir.

Derin derinlik tahmini için derin öğrenmenin gücünü incelemek için odaklanmış sentetik ve odaklanmış görüntüleri kullanarak deneyler gerçekleştirdik.

Deneyler, NYU v2 veri kümesinde derinlik haritası tahmini için derin evrişimli sinir ağlarını (D3Net), tekrarlayan sinir ağlarını (Kodlayıcı-Kod Çözücü) ve artık ağları (ResNet50) test eder.

Bu çalışma, odak dışı bulanıklığın derinlik haritası tahmin sonucunu önemli ölçüde geliştirdiğini göstermektedir. Üç yöntemden derinlik haritasını tahmin ettikten sonra, kameradan uzak, yakın ve uzak mesafe piksellerini belirlemede hangi yöntemlerin en iyi olduğunu bulmak için sonuç derinlik haritasını üç sınıfa (yakın, uzak, uzak) ayırdık.

Derinlik haritası tahmini ve segmentasyon performansı, sırasıyla beş ve üç değerlendirme metriği kullanılarak karşılaştırmalı olarak yapılmalıdır.

Ayrıca her bir yöntemin tahmin edilen derinlik haritaları görsel olarak incelenmiş ve tartışılmıştır. Beş farklı metrik (MSE, RNS, Log RMS, Rel, Accuracy), nicel değerlendirmede her yöntemin performansını ölçer. Bu değerlendirmelere göre Carvalho'nun yönteminin başarılı bir derinlik haritası ürettiği ve derinlik haritası tahmininde diğer teknikleri geride bıraktığı gözlemlenmiştir. Laina'nın yöntemi biraz gürültü ile iyi sonuçlar alır. Lee'nin yöntemi, belirgin bulanıklık ile güvenilir bir sonuç alır. Test sonuçları, bu modellerle kullanılan NYU v2 veri setinin Carvalho, Laina ve Lee için sırasıyla %99,8, %99,0 ve %98,8 doğruluk değerlerine ulaştığını göstermektedir.

Bununla birlikte, her yöntemin performansını ölçmek, segmentasyon sürecinde üç metrik (Piksel doğruluğu, IoU, DC) kullanan nicel karşılaştırmaya bağlıdır. Laina'nın yönteminin sadece yakın ve uzak alanlarda en iyi derinlik harita sınıflarını ürettiği, Carvalho'nun yönteminin ise uzak alanda en iyisi olduğu kanıtlanmıştır. Doğruluk değerleri Carvalho, Lee ve Laina için sırasıyla %77, %55, %90 idi.

ACKNOWLEDGEMENTS

First, I am grateful to God Almighty for providing me with the chance and ability to undertake this research and overcome the obstacles I encountered. Without his determination, this triumph would not have been feasible.

I want to express my deepest thanks to my supervisor, Associated Professor Dr. Mustafa Oral allowed me to work under his supervision and learn from his experience. He always encouraged and guided me throughout the research, from choosing the title to finding the results and organizing the thesis. It has been an excellent source of motivation and inspiration for me. Working with him has really improved my professionalism which will help me in my future endeavors.

Apart from my supervisor, I would like to thank the Master thesis jury members, Assoc. Prof. Dr. Zekeriya TÜFEKÇİ, Asst. Prof. Dr. Mümine KAYA KELEŞ, for their suggestions.

I also extend my thanks to all the professors and administrative staff I met in both the Department of Computer Engineering and Department of Electronics and Electrical Engineering, Çukurova University, for allowing me to realize my dream of completing my master's degree by allowing me to make use of the departmental facilities.

Finally, I must convey my heartfelt appreciation to all members of my family (my father, mother, and brother), you have always been the main supporter, and without you I would not have been able to reach this stage and finish this work. Thank you from the bottom of my heart.

Thank you

TABLE OF CONTENTS **PAGE**

LISTS OF TABLES **PAGE**

LISTS OF FIGURES **PAGE**

LIST OF SYMBOLS AND ABBREVIATIONS

AI	: Artificial Intelligence
DL	: Deep Learning
ML	: Machine Learning
ANN	: Artificial Neural Network
CNN	: Convolutional Neural Network
Caffe	: Convolutional Architecture for Fast Feature Embedding
CRF	: Conditional Random Field
DNN	: Deep Neural Network
RNN	: Recurrent Neural Network
MLP	:Multi-Layer Perceptron
RGB	: Red Green Blue
RGB-D	: Red Green Blue-Depth
ReLU	: Rectified Linear Unit
GAN	: Generative Adversarial Network
PSF	: Point Spread Function
NYU v2	: New York University version 2
3D	: Three-Dimensional
2D	: Two-Dimensional
MatConvNet	: Matlab Convolutional Neural Network
MRF	: Markov Random Field
DeMoN	: Depth and Motion Network
SIDFD	: Single Image Depth from Defocus
US	: United States
IDs	: Detection Systems
D3-Net	: Dense Deep Depth Estimation Network
DFD	: Depth from Defocus
MSE	: Mean Squared Error

RMS	: Root Mean Square
Rel	: Absolute Relative
log RMS	: Logarithmic Root Mean Square
TP	: True Positive
TN	: True Negative
FP	: False Positive
FN	: False Negative
IoU	: Intersection-Over-Union
DC	: Dice Coefficient
ResNet	: Residual Neural Network
GeoNet	: Geometric Neural Networks
DenseNet-BC	: Densely Connected Convolutional Networks
WSM	: Whole Stripe Masking
ALS	: Alternating Least Squares
SVD	: Singular Value Decomposition
BerHu	: inverse of Huber
DCNN	: Deep Convolutional Neural Network
CPU	: Central Processing Unit
GPUs	: Graphics Processing Units
ILSVRC	: ImageNet Large Scale Visual Recognition Challenge

1. INTRODUCTION

Depth estimation from a single image is a field as ancient as computer vision, and it encompasses a variety of techniques that have evolved through time. Depth estimation is a fundamental computer vision issue that involves estimating the depth information of a scene from one or more images. The estimated depth provides important geometric cues in image processing applications such as image synthesis (Cheng et al., 2008; Ndjiki-Nya et al., 2011), scene recognition (Ren et al., 2012; Shotton et al., 2013), pose estimation (Taylor et al. 2012; Ye et al., 2011), and robotics (Biswas and Veloso, 2012; Lenz et al., 2015). Furthermore, it is generally recognized that having access to somewhat accurate depth information improves several computer vision tasks as compared to their RGB-only counterparts, such as reconstruction (Silberman et al., 2012), identification (Ren et al., 2012), semantic segmentation (Eigen and Fergus, 2015), and human posture estimate (Taylor et al., 2012). Several techniques for inferring depth from multiple view images (Recht, 2009; Ren et al., 2012) or video sequences (Kundu et al., 2014; Wedel et al., 2006) supply encouraging results. However, when only a single image is available, it is difficult because it is poorly posed (Eigen et al., 2015).

Early monocular depth estimates algorithms incorporated assumptions about scenes, such as a room made up of box blocks (Gupta et al., 2010), an apartment scene (Saxena et al., 2009), a conventional interior room with a floor and walls (Delage et al., 2006; Lee et al., 2010), and the dark channel prior (He et al., 2011). When the assumptions are incorrect, however, these methods become unreliable. To understand the 3D geometry of an item or a scene, this reconstruction typically relies on precise depth computations. Traditional depth estimation methods use several physical features of perception to extract 3D information, such as stereoscopic vision, structured light, structure from motion, and other depth cues in 2D images (Saxena et al., 2009). Some of these technologies, however, have constraints that are reliant on the environment (e.g., texture, sun) or even need the use of numerous

devices (e.g., projector, camera), resulting in complicated systems. Multiple attempts have been made to create remote systems; arguably the most noteworthy are light-field cameras, which extract a depth map using a microlens array in front of the sensor (Ng et al., 2005).

Methods based on convolutional neural networks for monocular depth estimation (CNNs) (Chakrabarti et al., 2016; Eigen et al., 2015; Eigen and Fergus, 2015; Lee et al., 2018; Roy and Todorovic, 2016) in recent years, there has been much talk about it and because of advances in computer technology and the availability of an extensive range of training data, performance has increased significantly (Geiger et al., 2013; Silberman et al., 2012). The goal of these CNN-based techniques is to directly measure absolute depths. However, as highlighted in (Eigen et al., 2015), the monocular depth estimate is equivocal in terms of scale: at a closer distance, an object may appear to be identical to a smaller but identically formed object.

On the other hand, deep learning significantly improved conventional benchmark datasets' accuracy, placing these techniques first in state of the art. Starting with (Eigen et al., 2015), numerous deep learning-based techniques for depth estimate, known as deep depth map estimation, have been presented. These techniques employ a single point of view (a single picture), resulting in small, standardized systems. The bulk of them uses convolutional neural networks (CNNs) to estimate the 3D structure based on depth cues in the image and geometrical components of the scene (Eigen and Fergus, 2015; Li et al., 2015; Wang et al., 2015; Chakrabarti et al., 2016). To train the network, some people use supplementary depth cues like stereo information (Ummenhofer et al., 2017) and enhance predictions.

The defocus blur has long been an essential indication for depth assessment. Depth from Defocus (DFD) has been studied extensively before (Pentland, 1987; Levin et al., 2007; Trouvé et al., 2013; Sellent and Favaro, 2014; Zhuo and Sim, 2011). It resulted in a variety of depth prediction analytical approaches and optical systems. Because of the camera's depth of field, no blur can be seen, DFD with a

typical camera and a single picture suffers from the uncertain in-depth estimate about the focus plane and dead zone. It is tempting to combine the power of neural networks with defocus blur, which raises the question of defocus blur improves deep depth estimation performance or not.

Our work aims to examine three present estimating depth map algorithms and carry out comparative work to demonstrate the success rates of the methods and to show the influence of defocus blur images. It also aimed to show how successfully the algorithm allowed correct pixel depth classification (near, far, and faraway).

Because Carvalho and Laina's algorithms were pre-trained using a different dataset, and Lee's algorithm was trained using the NUY v2 dataset, the objective of this thesis is to observe the blind performance of the two algorithms that are not trained on the tested dataset.

Using NYU v2 images of simulated and real scenes, we computed their depth maps using the three methods and calculated the overall time it took.

Two steps separate this work; in the first step, we estimate the depth map using the three methods. We segmented the predicted depth map into three classes in the second step. The first method is Carvalho (Carvalho et al., 2019). To investigate the impact of defocus blur on depth estimation, researchers used a dense neural network called D3-Net (Carvalho et al., 2018). The depth estimation rendering is evaluated using a synthetically defocused NYUv2, including optically actual blur fluctuation, allowing comparison and analysis of various optical settings. This experiment demonstrates that neural networks employ defocused information, which is crucial for improving deep depth estimates. Furthermore, the suggested utilization of structural and blur information overcomes the present constraints of single-image DFD in terms of ambiguity and the dead zone in the focus plane. Eventually, we demonstrate how these results may be utilized in a specialized machine with genuine defocus blur to forecast depth inside accurately.

The second method is Lee (Lee and Kim, 2019); they present a unique relative depth map-based monocular depth estimation technique. This is due to the

fact that relative depth, or the ratio of the depths of two sites, is scale-invariant. Choosing the nearest of two points is far more accessible to a person than measuring the absolute depth of each point. As a result, relative depths are less difficult to compute than absolute depths. Lee designed the first CNN as part of an encoder-decoder system, comprising many decoder blocks for computing relative and ordinary depths at various scales. Second, Lee developed a pairwise comparison matrix that was sparsely populated with the projected relative depths. The entire matrix was recovered using the alternating least squares (ALS) approach (Koren et al., 2009), and a relative depth map was produced from the matrix's rank-1 property. Third, each depth map is disassembled into components, which are then reassembled using a limited optimization approach to create a final depth map.

The findings revealed that relative depth maps are more successful than standard depth maps at maintaining depth ordering. The effectiveness of segmentation, on the other hand, was lower than projected.

The third method, Laina (Laina et al., 2016), used a CNN; it was suggested to learn the mapping between a single RGB picture and the related depth. First, Laina developed a completely convolutional depth prediction architecture, complete with unique up-sampling blocks, that generates dense output maps with higher resolution while using fewer parameters.

Furthermore, a more efficient technique for upconvolutions was presented, which was used with the notion of residual learning (Chen et al., 2016) to create effective feature map upsampling up-projection blocks. Finally, to train the network, Laina optimized a loss based on the reverse Huber function (berHu) (Liu et al., 2016).

Many approaches have been proposed in the literature. However, there is still a lack of studies that have been done to evaluate these methods under realistic datasets like the NYU V2 dataset.

The main difference between this thesis and the studies in the literature is the proposition of a comparative study of deep learning approaches for depth

estimation from a single image, whether the image is in focus or blurred, using completely different artificial networks from each other. Furthermore, this thesis describes the limitations of these methods and briefly introduces future trends.

Also, we suggested a method to classify the predicted depth map into three classes to open the way for one of these algorithms to be used in the cameras used in drones, robots, self-driving cars, etc.

The thesis is categorised as follows: Chapter two studies the related work of depth maps and different architectures for estimating. Chapter Three explain the three algorithms used in the comparison, in addition to an accurate description of the database used. At the same time, the obtained results are mentioned in Chapter four. Chapter 5 contains a discussion about the three techniques. Finally, Chapter 6 concludes the thesis and suggestions for future studies.

2. RELATED WORKS

2.1. Estimation of Single Image Depth Maps

With the advancement of deep learning and high-end technology, computer vision tasks have witnessed new advancements. Depth detection is one of these jobs, which entails extracting three-dimensional information from two-dimensional items such as pictures and constructing a depth map. There are several ways for estimating depth from a single picture (Koch et al., 2018). An artificial neural network trained on pairs of pictures and their depth maps is frequently used in these approaches. These approaches are simple to comprehend and use, giving enough accuracy.

2.1.1. Single Image Blur Estimation

There are two types of single-image blur estimating methods: edge-based and region-based techniques. The main idea behind edge-based approaches is to estimate a distributed blur map along picture edges. Some edge-based algorithms additionally investigate interpolation/extrapolation schemes to transmit the blur information retrieved at edge locations, resulting in a dense blur map (Karaali and Jung, 2018). Regional-based algorithms typically examine picture patches to determine the local blur level and employ a regularization strategy (Zhu et al., 2013). Some of these techniques operate in the frequency domain, utilizing the convolution theorem to connect blur Point Spread Function (PSF) convolutions in the spatial domain with frequency domain multiplications (Zhu et al., 2013).

2.2. Relative Depth Map Estimation

Before adopting Convolutional Neural Networks (CNNs), some methods used a relative depth map, the proportion of two areas' depths in an image.

For monocular depth estimation, Zoran et al. (Zoran et al., 2015; Chen et al., 2016) employed pairwise depth comparison findings among pixels.

To reconstruct a comprehensive depth map, Zoran et al. (Zoran et al., 2015) estimated relative depths among sampled points and propagated them to super pixels. Estimating the relationships between pixels has fixed a more specific problem than metric regression. Zoran's approach works fully on two major mid-level vision tasks: essential image decomposition and depth from an RGB image.

2.2.1. Learning with Ordinal Relations

Chen et al. (Chen et al., 2016) introduced a unique method that learns to predict metric depth from relative depth annotations by categorizing relative depths between pixels into three classes: "near," "far," and "equal." By training the network with various loss functions based on paired labels, Chen et al. (Chen et al., 2016) were able to generate a pixel-level prediction. Chen et al. (Chen et al., 2016) also provided a new RGB-D dataset that dramatically enhances single-image depth estimation in the real world.

A Markov Random Field (MRF) model estimate depth map was suggested by Saxena et al. (Saxena et al., 2009). They solved the challenge of predicting depth maps from single monocular images of uncontrolled settings, such as woods, trees, buildings, people, buses, shrubs, and so on, using supervised learning. To forecast depth, the MRF was discriminatively trained. As a result, rather than modeling the joint distribution of image features and depths, they just simulated the posterior distribution of depths given the image features. Even in uncontrolled situations, the algorithm produced reasonably accurate depth maps.

Liu et al. (Liu et al., 2010) addressed the problem of depth perception from a single monocular image by incorporating predicted semantic information. Predicted labels are utilized as extra constraints to help speed up the optimization process. The first phase of Liu's method did a semantic segmentation of the scene, and in the second phase, the semantic labels led the 3D reconstruction.

Liu's technique provides a number of benefits: The semantic class of a pixel or area can be used to apply depth and geometry restrictions. In addition, depth can

be more smoothly predicted by measuring the difference in appearance concerning a provided semantic class. Finally, combining semantic features achieves state-of-the-art outcomes with a significantly simpler model. Liu's technique has the disadvantage of not being able to learn from richer data sources, such as synthetic data and images with poor labels.

To create finer and edge-conforming depth maps, Conditional Random Field (CRF) models are frequently fused with CNNs. Liu et al. (Liu et al., 2016) developed a deep structured learning project that learns the unary and pairwise possibilities of continuous CRF in a single deep CNN frame, with the goal of concurrently searching the deep CNN and continuous CRF capabilities. Then, to speed up the patch-wise convolutions in the deep model, Liu proposed a realistic model based on fully convolutional networks and a revolutionary super pixel pooling approach that is 10 times faster. Liu's technique had a major flaw in that it didn't use any geometric hints.

Heo et al. (Heo et al., 2018) introduced a monocular depth prediction approach that included reliability-based refinement and was based on Whole Strip Masking (WSM). Heo first developed a Convolutional Neural Network (CNN) that was specifically designed for depth estimation. Second, Heo verified the accuracy of a depth estimate by adding additional layers to the primary CNN. Conditional Random Field (CRF) optimization was used by Heo et al. (Heo et al., 2018) to enhance the estimated depth map utilizing reliability information. Heo's approach gives state-of-the-art depth estimation performance while only requiring a small number of parameters, according to test findings.

By merging attention processes, Xu et al. (Xu et al., 2018) created a novel attention model that learns robust multi-scale characteristics automatically. The suggested attention model regulates the amount of information that should flow between related features at various sizes automatically. In addition, attention models have been widely utilized in computer vision, with particular success in increasing the performance of Convolutional Neural Networks (CNNs) in pixel-level prediction tasks.

The Xu (Xu et al., 2018) technique merges multi-scale information acquired from distinct front-end Convolutional Neural Network (CNN) layers using a continuous Conditional Random Field (CRF). The suggested attention model is particularly noteworthy since it is completely integrated into the CRF, allowing for end-to-end architectural training. Xu's method is thought to be much better than prior depth estimate approaches based on CRF-CNN models because it combines multi-scale information at the feature level and uses a structured attention mechanism.

Qi et al. (Qi et al., 2018) proposed that depth and surface normal be deduced in one unified system utilizing Geometric Neural Networks (GeoNet). GeoNet's design comprises a two-stream CNN that predicts the depth and surface normal of each image separately. The two networks form the depth-to-normal and normal-to-depth mapping by handling the two streams. The results show that the GeoNet can predict the geometrically consistent depth and normal maps. Furthermore, it outperforms state-of-the-art depth estimate approaches when it comes to surface normal estimation. Furthermore, because these two networks have few learning parameters, they are computationally efficient.

For simultaneously estimating monocular depth map, optical flow, and camera motion from video, Yin and Shi (Yin and Shi, 2018) proposed an unsupervised learning system, GeoNet. The approach's foundation is built by the nature of 3D scene geometry. In addition, Yin and Shi proposed an adaptive geometric consistency loss to improve resilience towards outliers and non-Lambertian areas, resolving occlusions and texture ambiguities effectively.

The ability to learn these low-level vision tasks without expensive gathered ground truth data is demonstrated by the crucial findings connected to baselines, even the supervised ones—the Yin and Shi techniques' limitation is the gradient locality problem of warping-based losing.

Konrad et al. (Konrad et al., 2012) proposed an obvious algorithm that learns the scene depth from an extensive repository of image and depth pairs and is more effective computationally; this is accomplished by finding a median over the

recovered depth maps, then smoothing with cross-bilateral filtering. The produced images create a comfortable 3D perception but are not entirely without flaws.

2.3. Depth Map Prediction using Convolutional Residual Networks

Standard monocular depth estimation approaches have relied mostly on hand-crafted features and probabilistic graphical models to solve the challenge, solid assumptions about scene geometry. Unlike traditional Convolutional Neural Network (CNN) techniques, which need a multi-step procedure to modify their initially coarse depth predictions, these methods use a robust, single-scale CNN architecture that is guided by residual learning:

He et al. (He et al., 2015) proposed using a residual learning frame to make training of significantly deeper networks easier. Instead of learning unreferenced functions, He et al. (He et al., 2015) reconstructed the layers as learning residual functions regarding the layer inputs. For many visual identification tasks, the depth of representations is critical, and it was only because of He's method's intense representations that an apparent relative improvement was realized.

Mun and Ho (Mun and Ho, 2019) suggested the Residual Network (ResNet) based light field depth estimation network. Mun used the residual network with the neighbor sub-aperture images to handle discontinuous depth regions. The depth discontinuity issue is managed by selecting an optimal depth cost value from a set of sub-aperture depth maps. Mun got images that transformed a viewpoint by applying the phase shift process. To estimate the depth map of inhomogeneous areas, Mun used the phase-shifted images

Ladický et al. (Ladický et al., 2014) introduced a pixel-wise classifier that can estimate a semantic class and a depth label from a single image. Ladický demonstrated that using perspective geometry; a pixel-wise depth classifier could be reduced to a much simpler classifier that just predicted the possibility of a pixel being at a predetermined canonical depth.

Conditioning depth on the semantic class supports the classifier in distinguishing between ambiguities of the differently ill-posed problem. The main weaknesses of Ladický's method are the failure to deal with low-resolution images, many requirements in terms of hardware and the visible inability to determine the objects more specifically for semantic classes with high variance.

Non-parametric techniques for depth transfer are being developed soon:

Karsch et al. (Karsch et al., 2012) presented an approach that automatically uses non-parametric depth sampling to construct adequate depth maps from images. Karsch's method requires similar data (i.e., training with outdoor images for an indoor query is likely to fail). On the other hand, the method can handle large volumes of depth data with just minor degradation in output quality.

Using a combination of CNNs and graphical models to estimate depth maps is a new approach.

Li et al. (Li et al., 2015) proposed a new and widely applicable methodology for estimating depth and surface normals from single monocular images. It entails deep CNN regression and refinement using a hierarchical Conditional Random Field (CRF).

The framework has two levels of operation: super-pixel and pixel. To learn the mapping from multi-scale image patches to depth or surface typical values at the super-pixel level, Li et al. (Li et al., 2015) created a Deep Convolutional Neural Network (DCNN) form. Second, employing multiple potentials on the depth or surface normal map, the projected super-pixel depth or surface typical is enhanced to the pixel level.

Wang et al. (Wang et al., 2015) suggested a consolidated method for simultaneously estimating a single image's depth and semantic labels. Wang created a hierarchical Conditional Random Field (CRF) that incorporates the potential of a global CNN, a local CNN, and a regional CNN. Wang used a trained Convolutional Neural Network (CNN) to anticipate a global layout of pixel-by-pixel depth values and semantic labels. A state-of-the-art depth prediction is more accurate than the

combined network. By allowing for interactions between depth and semantic information, CNN was exclusively trained for depth prediction.

2.4. Estimation of Deep Monocular Depth

Many works have been created to estimate monocular depth using machine learning approaches. To solve this challenge, Saxena et al. (Saxena et al., 2005) employed a supervised learning strategy, which started with a training set of monocular pictures (of unstructured outside landscapes, such as woods, trees, and buildings) and their accompanying ground-truth depth maps. The depth map is then predicted as an image function using supervised learning. The Markov Random Field (MRF) in Saxena's model integrates multiscale local and global image characteristics. Saxena's approach demonstrates that an algorithm can routinely retrieve accurate depth maps even in unstructured settings.

Here are some more new deep convolutional network-based solutions:

Eigen et al. (Eigen et al., 2015; Eigen and Fergus, 2015) presented a multi-scale architecture for eliciting the global and local notification from the scene to estimate the corresponding depth map. This solution solved the problem by combining two deep network stacks: one that makes a poor global projection based on the entire image and the other that does not. The other refines this projection locally while efficiently utilizing the whole raw data distributions.

Cao et al. (2018) proposed a formulated depth estimation as a pixel-wise classification task. To increase the accuracy of the predictions, Cao employed a Conditional Random Field (CRF) to post-process the output of a deep Residual Network (ResNet) (He et al., 2015). This strategy quickly increases the confidence of a prediction by using information gain matrices during training and fully linked CRFs during post-processing.

Xu et al. (Xu et al., 2019) presented a deep model to combine complementary information collected from several CNN side outputs. Moreover, Xu's method adopted an intensely supervised approach combining the expected results of a

residual network (ResNet). A Conditional Random Field (CRF) module integrates the depth prediction with different scales and achieves high execution for the task. This approach outperforms by a significant margin on the other methods that used the cascade and the multi-scale models.

More recently Jung et al. (Jung et al., 2017) presented a Generative Adversarial Network (Goodfellow et al., 2020) (GANs) for the deep depth estimation field. To perfect the depth map predictions, this network adapter is an adversarial loss. As a generator, a two-stage convolutional network was created to anticipate the global and local structures of the depth image in order. Notably, this model can generate accurate and discontinuity-preserving depth prediction without any low-level segmentation of super pixels. The results display a substantial improvement, particularly on depth boundaries. In different strategies, (Godard et al., 2016)(Garg et al., 2016)(Ummenhofer et al., 2017) proposed investigating the epipolar geometry by using deep networks. Experimental results explain that these methods outperform if the dataset is a sizeable RGB-D dataset.

To estimate depth and camera motion from successive, unconstrained image pairings, Ummenhofer et al. (Ummenhofer et al., 2017) employed an end-to-end convolutional network termed Depth and Motion Network (DeMoN). Multiple stacked encoder-decoder networks make up the network architecture. The core of the system is an iterative network that may improve predictions. Ummenhofer's network calculates depth, velocity, surface normal, optical flow between images and matching confidence. The first deep network to estimate depth and camera motion from two unconstrained images was the Depth and Motion Network (DeMoN). DeMoN uses motion parallax, unlike other networks that estimate the depth of a single image. Unsupervised learning was utilized by Godard et al. (Godard et al., 2016) and Garg et al. (Garg et al., 2016) to rebuild stereo information and prophesy depth.

Kendall and Gal (2017) and Carvalho et al. (2018) investigated the reuse of feature maps during learning, based on an encoder-decoder with dense skip

connections. Kendall and Gal devised a regression function that takes into account the uncertainty in the data. This strategy enhances baseline performance by allowing the model to predict uncertainty more easily and lowering the loss. Furthermore, Carvalho et al. conducted a study on the impact of regression losses and experience circumstances on deep learning depth estimate. To enhance prediction, the network uses an end-to-end adversarial loss function. Small datasets create lower losses in this network.

2.4.1. Depth Estimation using Depth from Defocus (DFD)

Geometrical optics can be used to connect an object's blur to its depth. Pentland (Pentland, 1987) was the first work that used defocus blur to inferring depth. This approach looks at a source of depth information: focal gradients, which are caused by the restricted depth of field seen in most optical systems. The main limitation of this method represents in measuring focal gradient depth information. This method ensures enough high-frequency information to measure the variation between images.

Most modern works utilize of DFD with a Single Image Depth from Defocus (SIDFD). Zhuo and Sim (Zhuo and Sim, 2011) used analytical samples for the scene, like sharp edge examples. Trouvé et al. (Trouvé et al., 2013) and Levin et al. (Levin et al., 2010) used statistical scene Gaussian priors. Levin et al. (Levin et al., 2007), Sellent and Favaro. (Sellent and Favaro, 2014), Veeraraghavan et al. (Veeraraghavan et al., 2007), and Chakrabarti and Zickler (Chakrabarti and Zickler, 2012) proposed the coded apertures that improving depth estimation accuracy concerning standard optics. With model-based systems, determining depth from the amount of blur involves a tedious explicit calibration, which is often done using point sources or a known high-frequency pattern (Levin et al., 2007; Delbracio et al., 2011) at each probable depth.

Anwar et al. (Anwar et al., 2017) described a technique for estimating depth that uses deep representations learnt from cascaded convolutional and entirely

related neural networks. Anwar's network working on a patch-pooled of feature maps reconstruct an all-focus image. This approach is quite quick, and it greatly increases depth accuracy.

3. MATERIALS AND METHODS

3.1. Dataset and Image Pre-processing

Computer vision has exhibited a renewed interest in RGBD (Red Green Blue Depth) based vision algorithms and 3D reasoning techniques with the Microsoft Kinect introduction. Where laser range finders are expensive and often difficult to calibrate, the Kinect is cheap and easy to use (Han et al., 2013). Kinect choice and similar cameras have improved results in various jobs such as posture estimation, pedestrian detection, segmentation, tracking, and depth map estimation.

These efforts have been made viable by releasing 3D datasets that have allowed researchers to train models requiring a significant amount of data and that reliable permit comparison of algorithmic performance. Previous datasets, however, have been limited in several ways: Firstly, most previous images do not exhibit diversity in lighting conditions, object types, or backgrounds. Many are taken outdoors or are of simple indoor office scenes. Secondly, although it was set that a few of these datasets have been annotated with semantic class labels (Silberman et al., 2012), these semantic classes are exclusively limited to a small set of classes and rarely contain pixel-wise labels. Thirdly, previous datasets failed to capture real-world indoor scenes relating to everyday life (bedrooms, bathrooms, living rooms, etc.).

In this study, the New York University version 2 (NYU v2) (Silberman et al., 2012) dataset was used. The data for the second iteration of the dataset was gathered from a variety of business and residential buildings in three US locations (US).

To have a different scene content and not look like other frames, each room scanned from several locations, including its corners and center. Where manually selected at most three frames, carefully chosen from each room filmed to ensure each image provided a distinct viewpoint of the room and displays large occlusion and lighting variation.

The NYU-Depth v2 data collection comprises video sequences from diverse interior situations captured by the Microsoft Kinect's RGB and Depth cameras (Silberman et al., 2012). The NYU v2 data set is commonly used for benchmarking various tasks including semantic segmentation (Rani et al., 2016; Xie et al., 2016; Brownlee, 2018; Garg et al., 2016; Pykes, 2020), contour segmentation (Witkin, 1981; Malik and Rosenholtz, 1997; Simonyan and Zisserman, 2014; Karaali and Jung, 2018), 3D detection (Ummenhofer et al., 2017), depth from RGB (Tang et al., 2013; Laina et al., 2016), normal from RGB (Pentland, 1987; D'Andres et al., 2016; Xu et al., 2018) and physics-based reasoning (Tulsiani et al., 2017). Has features (Silberman et al., 2012):

- 1449 aligned RGB and depth images with heavily labeled pairs (2.8GB).
- 464 new scenes from three cities
- 407,024 unlabeled new frames
- A class and an instance number are assigned to each object.

The NYU v2 dataset includes the following components (Silberman et al., 2012): A portion of the video data that has dense multi-class labels is labeled. Where depth labels are missing, the data has been preprocessed to incorporate them. The raw RGB, depth, and accelerometer data produced by the Kinect is called raw. The Toolbox is used to manipulate the data and labels.

- ✓ Labeled: A subset of the Raw Dataset is a subset of the video data followed by dense multi-class labelling. It contains synchronized pairs of RGB, and Depth frames labeled with dense annotations for each picture. The labeled dataset is stored in MATLAB. Mat file, including the following variables (Silberman et al., 2012):

- **accelData -** An Nx4 matrix of accelerometer measurements was utilized to show when each frame was used. The roll, yaw, pitch, and tilt angles of the gadget are all listed in the columns.
- **images** -The image was presented with an HxWx3xN matrix of RGB images; W and H stand for width and height, respectively; N stands for the number of images.
- **instances** – Instance mappings in a WxHxD matrix. To reclaim masks for each object instance in a scene, use the Toolbox's get instance masks.m command.
- **depths** –WxHxN depth map matrix, where W and H are the width and height, respectively, and N is the number of images. The depths are the values in metres of the depth elements.
- **labels** – Object label masks in a WxHxN matrix, where W and H are the width and height, respectively, and N is the number of images. The labels are numbered 1 through C, with C denoting the total number of classes. When the label value of a pixel is 0, that pixel is considered 'unlabeled.'
- **scenes** – The title of the scene from which each image was shot is stored in a Nx1 cell array.
- **sceneTypes** – The scene type from which each image was obtained is represented as a Nx1 cell array.
- **names** - A Cx1 cell array with all class titles in English.
- **namesTolds** – Intrusion Detection Systems (IDs) are mapped from English label titles (with C key-value pairs).
- **rawDepths** – • WxHxN matrix derived from raw depth maps; W and H stand for width and height, respectively, while N stands for the number of images. Those depth maps are the Kinect's raw output.

- **rawDepthFilenames:** The filenames (in the Raw dataset) used for each of the depth images in the labeled dataset are stored in a Nx1 cell array.
- **rawRgbFilenames:** The filenames (in the Raw dataset) used for each of the RGB images in the labeled dataset are stored in a Nx1 cell array.

✓ Raw: The raw RGB contains the raw image as well as the accelerometer provided by the Kinect. The average frame rate for RGB and Depth cameras is between 20 and 30 frames per second (variable over time). The timestamps for every RGB, depth, and accelerometer file are included in every filename, even if the frames are not synchronized.

Figure 3.1. RGB camera output (left), preprocessed depth (middle), and image labels (right) (Silberman et al., 2012).

Even though the NYU-Depth dataset (Silberman et al., 2012) comprises over 230k pairs of images, we chose a smaller selection of 1449 RGB and depth images. We divided the dataset into two groups, with 55% (795 pairs) used for training and 45% (654 pairs) used for testing. As a result, experimentations in this study were done using a reduced dataset to execute faster trials. The resolution of the original frames from Microsoft Kinect output is 640x480.

Figure 3.2.Sample images of NYU v2 dataset.

3.2. Hardware and Frameworks

3.2.1. Hardware

The selected deep learning applications were implemented using the MATLAB framework and Python on the virtual machine and Intel Corei9-12900K with 12GB DDR4/DDR5 of memory, Windows 10 Operating System.

3.2.2. MATLAB

MATLAB creates a controlled desktop environment for repetitive analysis and design activities with a programming language that conveys matrix and matrix mathematics directly. The Live Editor allows you to create scripts that mix code, output, and formatted text into an executable notebook. Professionally produced, well tested, and fully documented MATLAB Toolboxes. You may explore how different algorithms function with your data using MATLAB apps. Furthermore, the flexibility to scale your analytics to run on clusters, GPUs, and clouds with modest code changes. There is no need to modify your code or learn advanced data programming or out-of-memory operations. MATLAB can analyze data, design algorithms, and create models and applications (Matlab, 1994).

3.2.3. Kinect

Microsoft's Kinect is a collection of motion sensor input devices initially introduced in 2010. RGB cameras, infrared projectors, and detectors that assess depth using structured light or time-of-flight calculations are typical components of the devices, which may be utilized for real-time sign recognition and body skeleton identification, among other things. It also has microphones for voice control and speech recognition (Wikipedia, 2020).

3.2.4. Caffe Model

Caffe (Convolutional Architecture for Fast Feature Embedding) is a deep learning framework that allows users to create image classification and segmentation

models. Users create and store their models as essential text PROTOTXT files at first. The user then uses Caffe to train and enhance their model, which stores the user's trained model as a CAFFEMODEL file. The CAFFEMODEL file format is a binary protocol protector. As with PROTOTXT files, they cannot be read, viewed, or edited with a source code editor. CAFFEMODEL files are designed to be used in applications that use Caffe-based trained image classification and segmentation models (Caffemodel, 2021).

3.2.5. MatConvNet Model

MatConvNet is a MATLAB implementation of a Convolutional Neural Network (CNN). The toolkit was created with a focus on simplicity and adaptability. It provides processes for computing linear convolutions utilizing filter banks, feature pooling, and much more and offers the building elements of CNNs as easy-to-use MATLAB functions. MatConvNet enables quick development of new CNN architectures, as well as efficient computing on the CPU and GPU, allowing intricate models to be trained on large datasets like the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (MatConvNet, 2014).

3.2.6. Python

Python is a high-level programming language with operational semantics. Python's simple syntax emphasizes readability while lowering software maintenance costs. Python's support for modules and packages increases program flexibility and code reuse.

Python is a famous programming language. Guido van Rossum built it in 1991 (Anonymous python, 1999).

3.2.6.1. PyTorch

An open-source machine learning system quickens the way from inquiring about prototyping to generation deployment.

PyTorch is an optimized tensor library (Jwalapuram, 2020). They are utilized for Profound Learning applications utilizing GPUs and CPUs. It is an open-source library for Python, planned unequivocally by the Facebook AI Investigate group. It is one of the standard Machine learning libraries.

3.3. Deep Learning

3.3.1. Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) are a critical component of the Deep Learning approach. ANNs are "complex computer code built with a lot of simple, extremely linked processing components that are inspired by human biological brain structure for replicating human brain functioning & processing data (Information) models," according to AILabPage (Anonymous neural network, 2020).

Artificial Neural Networks (ANNs) are complicated forms of artificial neurons that can process many inputs and output a single result. The primary purpose of an Artificial Neural Network is to turn data into meaningful output.

An Artificial Neural Network is made up of an input and output layer and several hidden layers. All of the neurons in an Artificial Neural Network impact one another; hence they are all linked, as illustrated in Figure 3.3.

The transfer of information in an Artificial Neural Network happens in two ways:

Feedforward Networks: The signals in this model go in a specific direction toward the output layer. Feedforward networks have a single input layer and a single output layer, with zero or several hidden layers. They are commonly used in pattern recognition.

Feedback Networks: In this approach, the repeating or interactive networks treat the sequence of inputs by using their internal state. Signals can pass through the network's (hidden layer/s) loops in both ways. They are employed in jobs that need time-series and sequential data.

3.3.1.1. Artificial Neural Network elements

1- Neurons, Input Layers, and Weights: A neuron is the basic unit of a neural network. The data comes from a third-party source or other nodes. Every node in the following tier is connected to another node in that layer, and each connection has a different weight. A neuron's weights are chosen depending on its relative significance to other inputs. When all node values from the input layer (along with their weight) are compounded, a value for the first hidden layer is generated. The first hidden layer contains a predetermined "activation" function that decides whether or not this node will be "activated" and how "active" it will be based on the data from the input layer. As seen in Figure 3.3.

2- Hidden Layers and Output Layers: The hidden layer is the layer that sits between the input and output layers. Because it is constantly concealed from view, it is called the hidden layer. The hidden layers of an Artificial Neural Network are where the essential calculations occur. As a result, the hidden layer receives all of the data from the input layer and does all of the necessary computations to provide a result. The result is then passed to the output layer, which displays the computation result.

Financial operations (Shachmurove, 2002), business planning (Alrammahi and Radif, 2019), trading (Chen, 2020), business analytics (Bihl et al., 2018), and product maintenance are all areas where neural networks are used (Chen, 2020). Business applications for neural networks include forecasting and marketing research solutions (Abbas, 2015), cheating detection (Dunham, 2020), and risk assessment (Dunham, 2020). (Kenton, 2020). A neural network calculates and identifies opportunities for making trading judgments based on data analysis. Other methods of technical analysis are unable to detect exact nonlinear interdependencies and patterns that the networks can.

The Vanishing and Exploding Gradient is a well-known issue in neural networks (Pykes, 2020). As demonstrated in Figure 3.5, this issue is related to the

backpropagation algorithm. The gradients are used to refresh the neural network weights using the backpropagation process. Furthermore, ANN cannot capture consecutive data in the input data, which is required for trading with sequence data.

3.3.1.2. Artificial Neural Network Types

The most basic version of ANN is the Multi-Layer Perceptron (MLP), which has three layers (Input, Hidden, and Output), as depicted in Figure 3.3. The data is received by the input layer, processed by the hidden layer, and generated by the output layer. Each layer is essentially attempting to learn certain weights. Multi-Layer Perceptron can solve problems correlated to tabular data, image data and text data. MLP has the advantage of being able to learn any nonlinear function. The issues with MLP arise from transforming a 2-dimensional image to a 1-dimensional vector before training the model in an image classification problem.

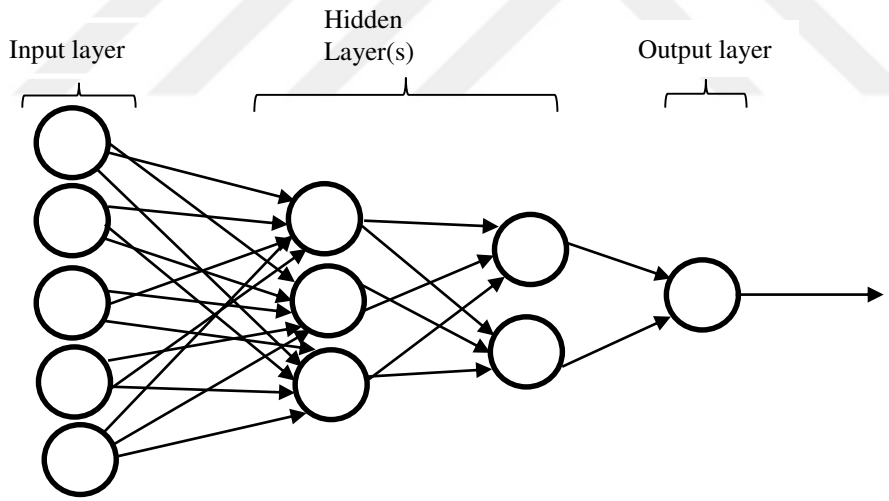

Figure 3.3. Example of an artificial neural network

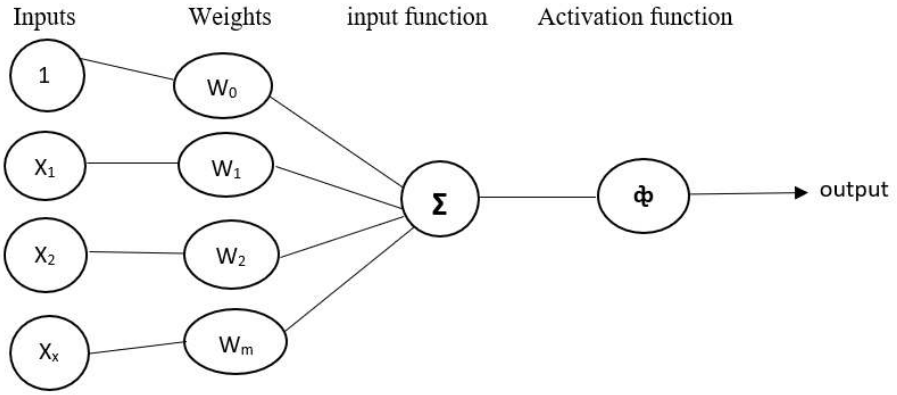

Figure 3.4. A diagram of one node.

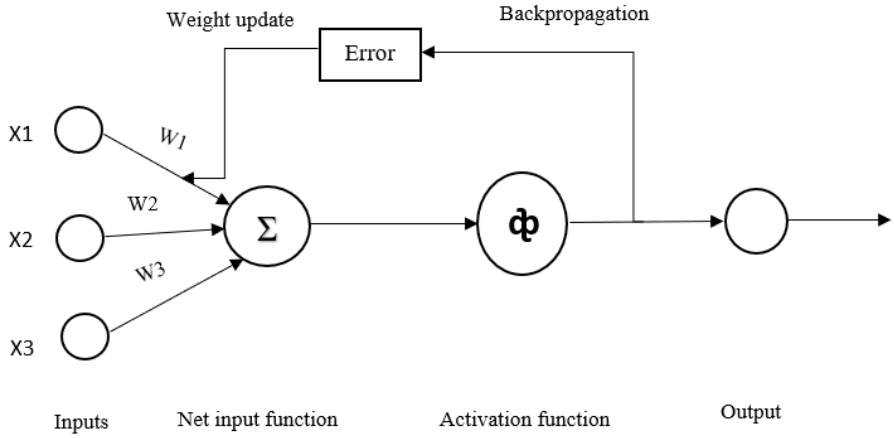

Figure 3.5. Backward Propagation neural network

A Recurrent Neural Network (RNN), a form of artificial neural network used in this study, is a variation of ANN. As demonstrated in Figure 3.6, RNN may be used to handle issues involving time series data, text data, and audio data.

Recurrent neural networks, like feedforward networks, learn from training data. Their "memory" distinguishes them since they use data from prior inputs to determine the present input and output. The sequence's fundamental constituents

determine the output of recurrent neural networks. While future occurrences can help determine a sequence's output, unidirectional recurrent neural networks can't forecast them.

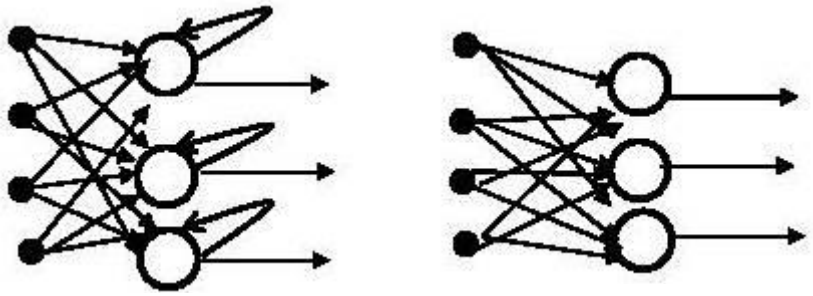

Figure 3.6. Recurrent Neural Network structure

Encoder-Decoder RNNs are one of the types of RNNs: the encoder-decoder design for recurrent neural networks is the standard neural machine-translation approach, as shown in Figure 3.7. This architecture is very new, investigated in 2014; however, it has been accepted as the primary technology within Google's translation service (Brownlee, 2018).

Encoding means converting data into a wanted format. When a series of Spanish words are converted into a two-dimensional vector, the two-dimensional vector is referred to as a hidden state in machine learning. A recurrent neural network is stacked to create the encoder (RNN). The encoder's output is a two-dimensional vector. The vector's length is determined by the number of cells in the RNN.

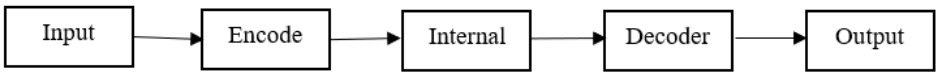

Figure 3.7. The encoder-decoder architecture

Decoding means converting a coded message into an understandable language. Using a decoder in a machine learning model means converting a two-

dimensional vector into a series of outputs. When a series of Spanish words are converted into a two-dimensional vector, the two-dimensional vector is referred to as a hidden state in machine learning. A recurrent neural network is stacked to create the encoder (RNN). The encoder's output is a two-dimensional vector. The vector's length is determined by the number of cells in the RNN.

For long input series, the fundamental constraint of this encoder-decoder paradigm is that all of the information must be summed in a one-dimensional vector, which might be challenging to perform.

Some of the best-performing approaches employed the encoder-decoder deep neural network design. Not just for in-depth perception but also for image segmentation, optical flow estimates, and image restoration, among other computer vision problems (Rocha, 2019).

In-depth map estimation, the encoder-decoder has been used for the supervised (Zhou et al., 2018) and unsupervised (Godard et al., 2016) version to solve the problem. In most cases, the encoder part of the network reduces the input and generates a feature vector from the picture. At the same time, the decoder creates a depth map based on the outputs of the hidden layer, which varies depending on the approach.

The other type of neural network used in this study is Convolutional Neural Networks (CNN): The Latin convolver, "to convolve," means to twist together, a convolution is a method of merging two functions by multiplying them. As for the mathematical meaning, convolution is the integral calculating how much two functions overlap as one moves over the other. The building stones of CNNs are filters, also known as kernels. Using the convolution technique, kernels choose similar characteristics from the input. These filters aid in the selection of the most relevant and appropriate characteristics from the supplied data.

Although convolutional neural networks were developed to handle problems with picture data, they also function well with many inputs. CNN models are widely

employed in various applications and domains, but they are especially common in image and video processing projects.

Convolution Neural Network (CNN) has the following advantages:

- CNN learns the filters automatically without telling them.
- CNN captures the spatial properties of an image.
- CNN replicates the parameter sharing theory.

CNN is the most excellent solution for computer vision tasks since it treats every input as an image (but it does not have to be an image). In addition, as illustrated in Figure 3.8, CNN processes full pictures and traditional neural networks.

For example, the first neuron of the first hidden layer would have 3072 weights for a picture of 32323 (32 pixels wide, 32 pixels high, and three color channels for red, green, and blue). Under challenging issues, a 32-pixel square cannot be used as the input picture. Assume a more reasonable picture size of 6404803, with 921600 weights in the first hidden layer's first neuron. In a neural network with the first hidden layer, about a million weights in just one neuron describe an unsuccessful neural network.

Furthermore, activation volumes are three-dimensional brain structures seen in CNNs. Instead of being wholly linked to hidden layers, convolutional layers are responsible for identifying elements from the image, such as edges, patterns, forms, and textures. This is done using filters or kernels, almost tiny matrices of values that incorporate the weights. These matrices convolve the image, which results in distinct activation volumes. The pooling layers (average or maximal pooling) help in picture compression.

Although several new solutions for depth estimation tasks use Convolutional Neural Networks, they vary widely from one to another. It can be divided into four subgroups of learning-based approaches: monocular, multi-view, encoder-de-coder, and transfer learning. Methods that use just one RGB image as input are named

monocular depth estimation algorithms. Some algorithms utilize two or more images as inputs and are labeled as multi-view methods. Transfer learning is a method that has been used to tackle the computer vision problem. We can summarize some of the variations among different types of neural networks in Table 3.1.

Although Convolutional Neural Networks are used in various innovative solutions for depth estimation challenges, they differ significantly from one another. Monocular, multi-view, encoder-decoder, and transfer learning are the four subgroups of learning-based techniques. Monocular depth estimation techniques employ just one RGB image as input. Multi-view techniques refer to algorithms that take two or more pictures as inputs. Transfer learning is a technique for dealing with computer vision problems. Table 3.1 summarizes some of the differences between different types of neural networks.

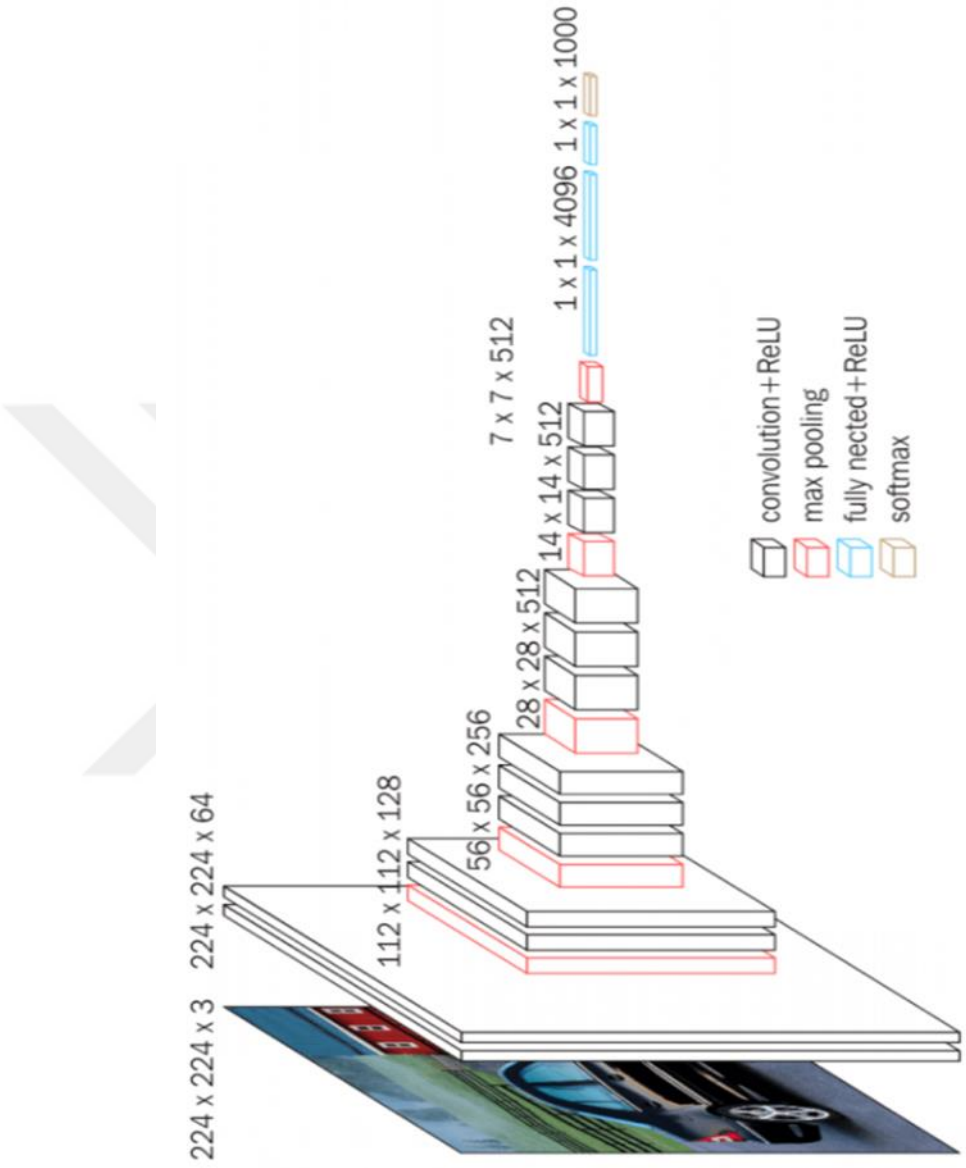

Figure 3.8. Model of a Convolutional Neural Network for image recognition (Hassan, 2018)

Table 3.1. Various types of Neural Networks (MLP(ANN) vs. RNN vs. CNN) are compared.

Data	MLP	RNN	CNN
	Tabular data	Data from a sequence (Time Series, Text, Audio)	Image information
Recurrent connections	No	Yes	No
Sharing of parameters	No	Yes	Yes
Spatial relationship	No	No	Yes
Vanishing & Exploding Gradient	Yes	Yes	Yes

One of this study's used Convolutional Neural Network types is Residual Neural Network (ResNet). It is based on the cerebral cortex, a known structure opyramidal cells. Residual Neural Networks go over certain levels by overstepping connections or shortcuts. As illustrated in Figure 3.9, typical ResNet models are implemented utilizing double or triple-layer skipping with non-linear components (ReLU) and batch normalization in between. Forms that contain many parallel skips are indicated as DenseNets, as shown in Figure 3.10. In the case of residual neural networks, a non-residual network can be defined as a normal network. The addition of skip connections is justified for two reasons: to bypass the gradient fading problem or to mitigate the saturation (precision saturation) problem, adding extra layers to a deep model that is adequate leads to a bigger training error (residual neural network - Wikipedia, 2021).

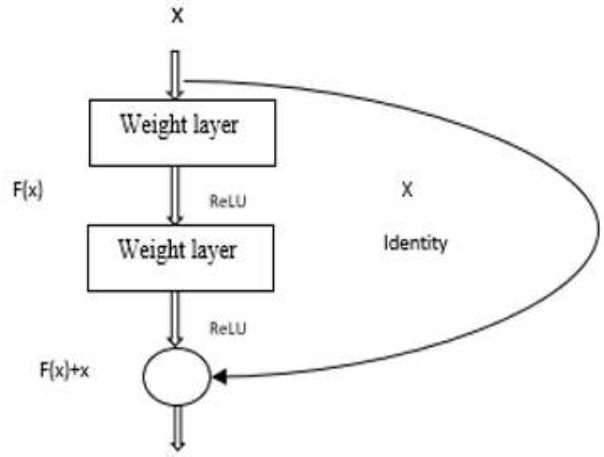

Figure 3.9. This figure shows the " shortcut connection " The curved arc "this is the so-called "shortcut connection". This figure also explains the true meaning of ResNet.

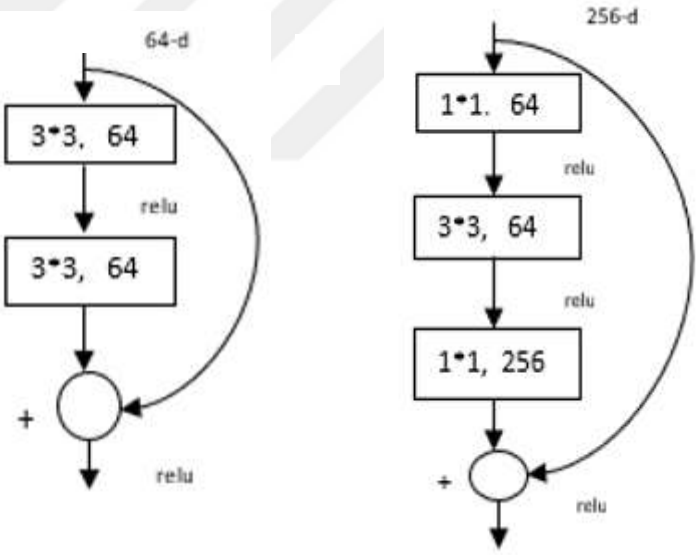

Figure 3.10. Two ResNet designs. These two structures are for ResNet34 (right picture) and RsNet 50/101/152 (left picture).

3.3.2. Concepts of Deep Neural Networks

The depth of deep-learning networks distinguishes them from ordinary single-hidden-layer neural networks. The perceptron was shallow in prior versions

of neural networks, consisting of one input, one output, and one hidden layer. There should be more than three layers in deep learning, including input and output.

Deep is more than a word. It is a well-defined phrase that denotes the presence of several concealed layers. Every layer of nodes in a deep-learning network trains on a unique set of features depending on the output of the preceding layer. Because the nodes collect and recombine characteristics from the preceding layer, the nodes can identify the more complicated as the neural network proceeds. Unlike most machine-learning algorithms, deep-learning networks extract features automatically without human intervention.

3.4. Depth Map

A depth map is a form of an image made up of gray pixels with values ranging from 0 to 255. As illustrated in Figure 3.11, the "0" value of gray pixels represents "3D" pixels that are placed at the farthest point in the 3D scene, while the "255" value of gray pixels represents "3D" pixels that are positioned at the closest point. A depth map is an image in 3D computer graphics and computer vision that contains information on the distance between the surfaces of scene objects from a perspective, as illustrated in Figure 3.12.

To create a depth map, one needs to thoroughly understand the perspective perspectives of objects inside a 2D picture. In addition, consider a 3D environment in which the items within the 2D image are determined from far to close. A "Grey Scale" marked 000 255 values is also required to establish the physical positions of the relevant things in mind.

Original 2D Image Depth Map Gray Sc

Figure 3.11. RGB Image and 8-bit depth map (Olivier, 2019)

3.4.1. Depth Maps Uses

1- Simulating small depths of field - when elements of a scene appear out of focus. Depth maps may be used to selectively blur an image to different degrees (Depth Map - Wikipedia, 2021).

2- To provide distance information and develop and generate auto stereograms and other similar applications to create the appearance of 3D seeing using stereoscopy (Depth Map - Wikipedia, 2021).

3- Making depth image datasets (Depth Map - Wikipedia, 2021).

4- Single-view, multi-view, and other pictures are used in computer vision to model and rebuild 3D forms (Depth Map - Wikipedia, 2021).

5- To facilitate the preparation of 3D pictures using 2D image tools in machine vision and computer vision (Depth Map - Wikipedia, 2021).

6- Shadow mapping is a part of a procedure used in 3D computer graphics to produce shadows cast by light (Depth Map - Wikipedia, 2021).

7- To improve the rendering of 3D sceneries, z-buffering and z-culling techniques can be applied (Depth Map - Wikipedia, 2021).

8- Simulating the effect of thick semi-transparent substances in a scene, such as fog, smoke, or significant amounts of water (Depth Map - Wikipedia, 2021).

9- Subsurface scattering adds realism by imitating translucent features like human skin's semi-transparent characteristics (Depth Map - Wikipedia, 2021).

Cubic Structure Depth Map(a) Depth Map(b)

Figure 3.12. Depth map representation, depth map(a) represents closely is darker, depth map(b) represents nearer the focal level is dusky (Depth Map - Wikipedia, 2021).

3.5. Deep Depth from Defocus

Deep learning-based depth estimation algorithms use geometrical characteristics of typical crisp images to anticipate identical depth maps. Defocus blur images, for example, are produced by cameras based on the depth of the objects and the camera settings.

Most of these algorithms employ convolutional neural networks (CNNs), relying on the image's geometrical characteristics to estimate the 3D structure based on depth cues (Li et al., 2015; Eigen and Fergus, 2015; Wang et al., 2015; Chakrabarti et al., 2016). Additionally, other depth cues such as stereo information may be used to train the network and enhance predictions (Ummenhofer et al., 2017); another important signal for depth estimation is the defocus blur.

To estimate 3D information of an unknown scene, Depth from Defocus (DFD) requires a scene model and a precise calibration between blur and depth value. As a result, combining defocus blur with the capability of neural networks is appealing.

To investigate the influence of defocus blur on depth estimation, Carvalho et al. (Carvalho et al., 2019) proposed a complete system for single-image depth map

prediction using depth-from-defocus and Dense Deep Depth neural network D3-Net (Carvalho et al., 2019). Carvalho et al. (Carvalho et al., 2019) conducted a series of experiments using synthetic defocus data to test deep learning's ability to forecast depth maps.

They also employed blur as a cue and did not use image processing for data increment capable of affecting out-of-focus data. Carvalho (Carvalho et al., 2019) utilized the D3-Net architecture from (Carvalho et al., 2019), shown in Figure 3.14. They initialized the D3-Net encoder, matching Deep Convolutional Neural Network (DenseNet- 121) with pre-trained weights and the random weighted D3-Net decoder. Comprehensive experiments were conducted to test deep convolutional neural networks on synthetic defocused images. To improve generalization, Carvalho added dropout regularization with a possibility of 0.5 to the first four convolutional layers of the decoder, as illustrated in Figure 3.13.

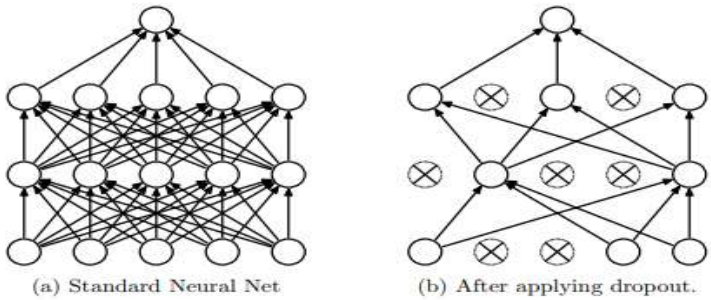

Figure 3.13. (a) Two hidden layers of a conventional neural network. (b): A thinner network created by applying dropout to the network on the left.

3.5.1. Synthetic NYUv2 with defocus blur

Carvalho (Carvalho et al., 2019) chose the parameters coordinating to a synthetic camera with a f-number 2.8, the focal length of 15mm, to make out-of-focus images that are physically realistic. Apart from the objects in the camera depth of field, which remain crisp, objects in the depth range of 1 to 10m will display minor

defocus blur levels. Closer objects will have to defocus blur, with the blur being in the 0-3m depth range.

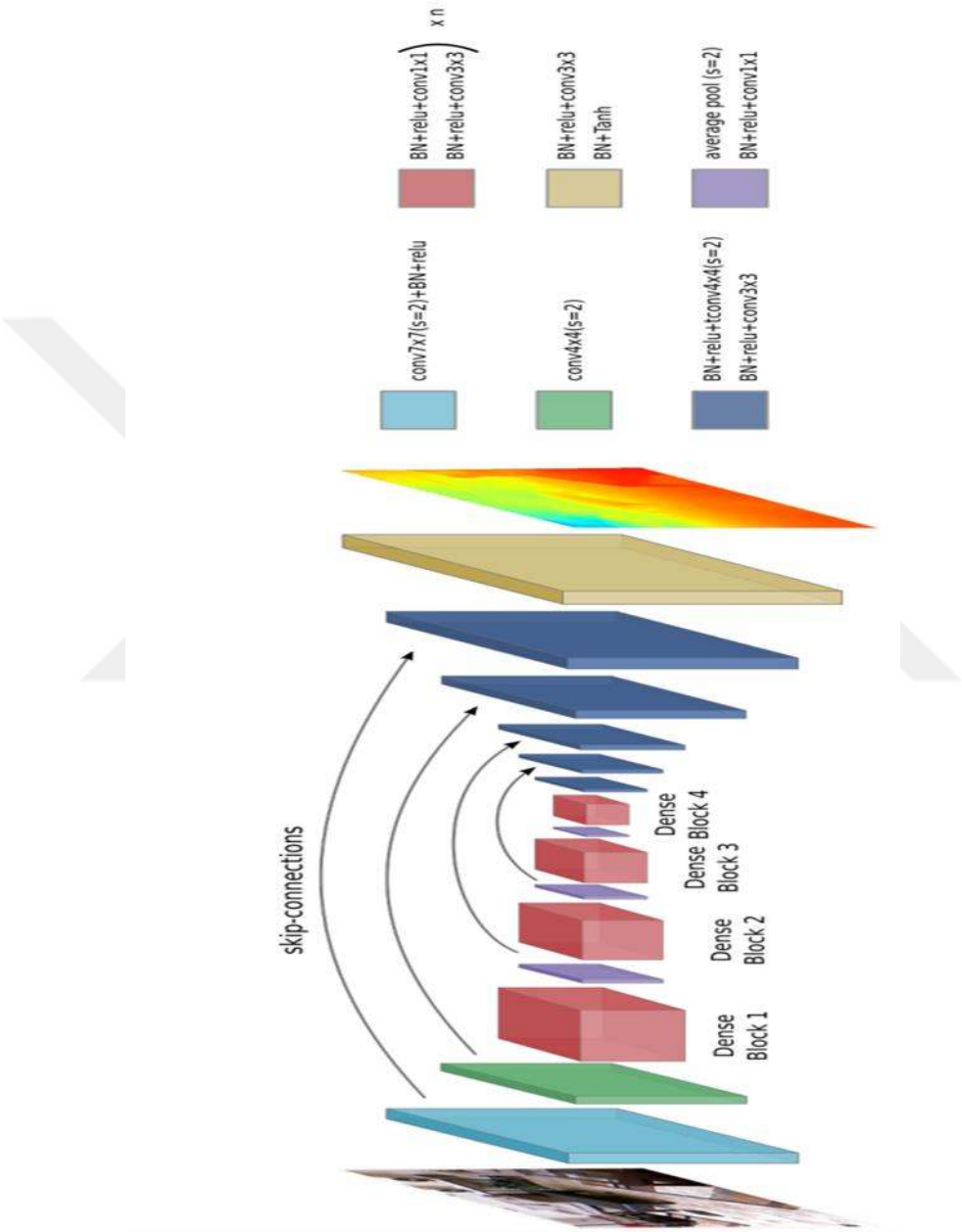

Figure 3.14. D3-Net is a network architecture that was developed by D3-Net (Carvalho et al., 2019). The encoder component coincides with a Dense Net-121 (Huang et

al., 2016). U-Net is used to build the encoder-decoder framework (Carvalho et al., 2019).

Figure 3.15. Sample images of blurred NYU v2 dataset using Carvalho method

3.6. Relative Depth Maps for Monocular Depth Estimation

Depth estimation is a genuine computer vision issue that involves estimating the depth information of a scene from a single image. In vision applications such as image synthesis (Cheng et al., 2008; Ndjiki-Nya et al., 2011), robotics (Biswas and Veloso, 2012; Lenz et al., 2015), pose estimation (Taylor et al., 2012; Ye et al., 2011), and scene recognition (Taylor et al., 2012), estimated depths provide important geometric clues (Ren et al., 2012; Shotton et al., 2013). With the advancement in computing hardware and the availability of a large amount of training data, monocular depth estimation methods based on convolutional neural networks (CNNs) have been suggested in recent years (Laina et al., 2016; Chakrabarti et al., 2016; Lee et al., 2018; Roy and Todorovic, 2016). (Silberman et al., 2012; Geiger et al., 2013). These CNN-based approaches attempt to predict depths directly. On the other hand, relative depth is the ratio between the depths of two places. Even for a human, determining the closest one between two points is easier than estimating the absolute depth of each point. In other words, relative depths are more easily approximated than absolute depths.

Lee et al. (Lee and Kim, 2019) developed a monocular depth estimation technique based on relative depth maps. The suggested method is depicted in Figure 3.16.

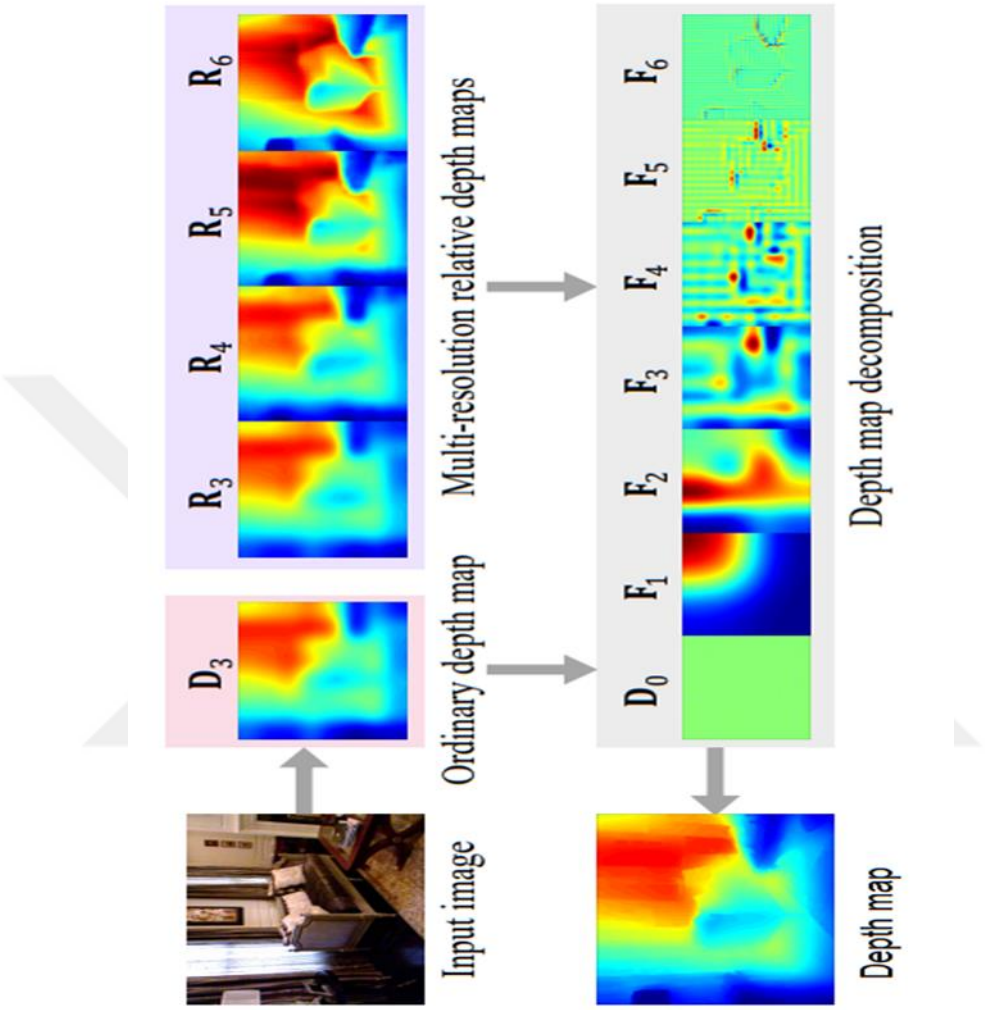

Figure 3.16. A description of Lee's algorithm. An image is first used to create one conventional depth map and four relative depth maps. They are then divided into depth components, which are then combined to create an ideal depth map (Lee and Kim, 2019).

3.6.1. Decomposition of a Depth Map

Let $I \in R^{r*c}$ be an image with the size r*c. The purpose is to calculate $D \in R^{r*c}$, the associated depth map. In the first, Lee et al. (Lee and Kim, 2019) defined and calculated relative depth, the ratio between the depths of two areas in an image, as an unchanging quantity. If the relative depths of all pairs of pixels in a image are known, rebuilding a depth map at a standard scale is feasible. Lee offers four suggestions to establish his point:

Suggestion (1): A scaled depth map D/g(D) can be rebuilt if the relative depths of all pixel pairs in (I) are known.

Lee verified it by assuming that all relative depths in D/D (i, j) are known for any pixel (i, j). g(D)/D (i, j) and its counterpart D (i, j)/g(D) are derived by geometrically averaging these depths (D). As a result, we will obtain D/g (D).

We can determine the geometric mean of a depth map D by

$$g(D) = \prod_{i=1}^{r} \prod_{j=1}^{c} D(i,j)^{1/rc} \quad (3.1)$$

Where D (i, j) is the (i, j) th depth in D.

Suggestion (2): D/g(D) has a geometric mean of 1.

According to suggestions 1 and 2, Lee established that g(D/g(D)) = g(D)/g(D) = 1; if all relative depths between pixel pairs are known, the relative depth map may be reconstructed.

According to suggestions 1 and 2, the relative depth map can be rebuilt by:

$$R = D/g(D) \quad (3.2)$$

D = g(D)R may thus be represented as the relationship between the original depth map D and the relative depth map R.

Lee then scaled down the depth map D to various proportions. Let D_n denote a depth map with a size of ($2^n * 2^n$). D_{n-1} is a lower resolution depth map derived from D_n by:

$$D_{n-1}(i,j) = \prod_{k=0}^{1} \prod_{l=0}^{1} D_n(2i-k, 2j-l)^{\frac{1}{4}} \quad (3.3)$$

The geometric mean of the four related depths in D_n is a depth in D_{n-1}. Lee devised the upsampling operation (U) to duplicate the size of a depth map horizontally and vertically by repeating each input depth four times to fill in the corresponding four pixels in the output depth map. Then comes F_n, which is given by:

$$F_n = D_n \oslash \cup (D_{n-1}) \quad (3.4)$$

Where $\oslash$ represents the Hadamard division.

Equivalently to,

$$D_n = \cup (D_{n-1}) \otimes F_n \quad (3.5)$$

Where $\otimes$ represents the Hadamard product.

Suggestion (3):

$$\prod_{k=0}^{1} \prod_{l=0}^{1} F_n(2i-k, 2j-1)^{\frac{1}{4}} = 1 \quad (3.6)$$

for each (i,j) and $g(F_n) = 1$

D_n can be decomposed in a logarithmic scale by applying recursive of :

$$\log D_n = log \cup^n (D_0) + \sum_{i=1}^{n} \log \cup^{n-i} (F_i) \quad (3.7)$$

The element-wise logarithmic function is called log. This indicates that $\log D_n$ is decomposed into the mean depth map $\log \cup^n(D_0)$ and the residual depth maps $\log \cup^{n-i}(F_i)$ for $1 \leq i \leq n$.

The arithmetic mean of each residual map $\log \cup^{n-i}(F_i)$ is zero, as suggested by suggestion 3. The relative depth map R_n can be deconstructed similarly as:

$$\log R_n = \sum_{i=1}^{n} log \cup^{n-1} (F_i) \tag{3.8}$$

In Lee's method, given an image I, D_n and Rn are estimated for $3 \leq n \leq 7$. Then, each D_n or R_n is decomposed by ($\log D_n$) or ($\log R_n$) equations, respectively.

Table 3.2. Results of decomposition of depths maps D_n and R_n for $3 \leq n \leq 7$ (Lee and Kim, 2019).

	D_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
D_3	√	√	√	√	-	-	-	-
D_4	√	√	√	√	√	-	-	-
D_5	√	√	√	√	√	√	-	-
D_6	√	√	√	√	√	√	√	-
D_7	√	√	√	√	√	√	√	√
R_3	-	√	√	√	-	-	-	-
R_4	-	√	√	√	√	-	-	-
R_5	-	√	√	√	√	√	-	-
R_6	-	√	√	√	√	√	√	-
R_7	-	√	√	√	√	√	√	√

Suggestion (4): P_3 is a rank1 matrix if there is no estimation mistake.

$$P_3 = [d_1, d_2, \dots, d_{64}]^T [\frac{1}{d_1}, \frac{1}{d_2}, \dots, \frac{1}{d_{64}}] \tag{3.9}$$

Lee created the pairwise comparison matrix P_3, including the relative depths between all pixel pairs in D_n, in order to provide consistent and dependable results from the estimated relative depth. In an ideal case, the comparison matrix would be:

$$P_{n,n-1} = \left[d_1^n, d_2^n, \dots, d_{2^{2n}}^n\right]^T \left[\frac{1}{d_1^{n-1}}, \frac{1}{d_2^{n-1}}, \dots, \frac{1}{d_{2^{2n-2}}^{n-1}}\right] \quad (3.10)$$

Where d_i^n indicates the *i*th depth in the reshaped vector of D_n. Without estimation errors, the rank of $P_{n,n-1}$ is equal 1.

3.6.2. Network for Depth Estimation

Lee et al. (Lee and Kim, 2019) utilized the encoder-decoder architecture (Badrinarayanan et al., 2017; Yang et al., 2016) to estimate depth maps, as shown in Figure 3.17.

Encoder component: Deep features are taken from an image in the encoder part. The encoder converts an image into high-level, low-resolution characteristics. Except for the last dense block, the encoder was DenseNet-BC (Huang et al., 2016), which consisted of one convolution layer, one max-pooling layer, and three pairs of dense block and transition layer, as shown in Figure 3.17.

Decoder component: Ordinary depth maps Dn and relative depth maps Rn were reconstructed using the features retrieved in the decoder section by the 10 decoders. The 10 decoders added higher-resolution depth maps Rn and Dn to the low-resolution features. Each decoder has one dense block and many Whole Stripe Masking (WSM) blocks (ranging from 0 to 4) (Heo et al., 2018). WSM is a fundamental architecture upsampling block (Szegedy et al., 2014). WSM uses kernels with horizontal or vertical diameters equal to the whole input signal to improve the receptive field greatly. The number of WSM blocks is determined by the resolution of a targeted depth map.

Each decoder uses ordinal regression (Li and Lin, 2006) to rebuild the depths. Many binary classifiers can be used in an ordinal regression assignment to determine if a value is larger than other criteria. Lee quantized their reconstruction levels and ordinal loss function for ordinary depth maps Dn decoders. However, different reconstruction levels are required in decoders for relative depth maps R_n.

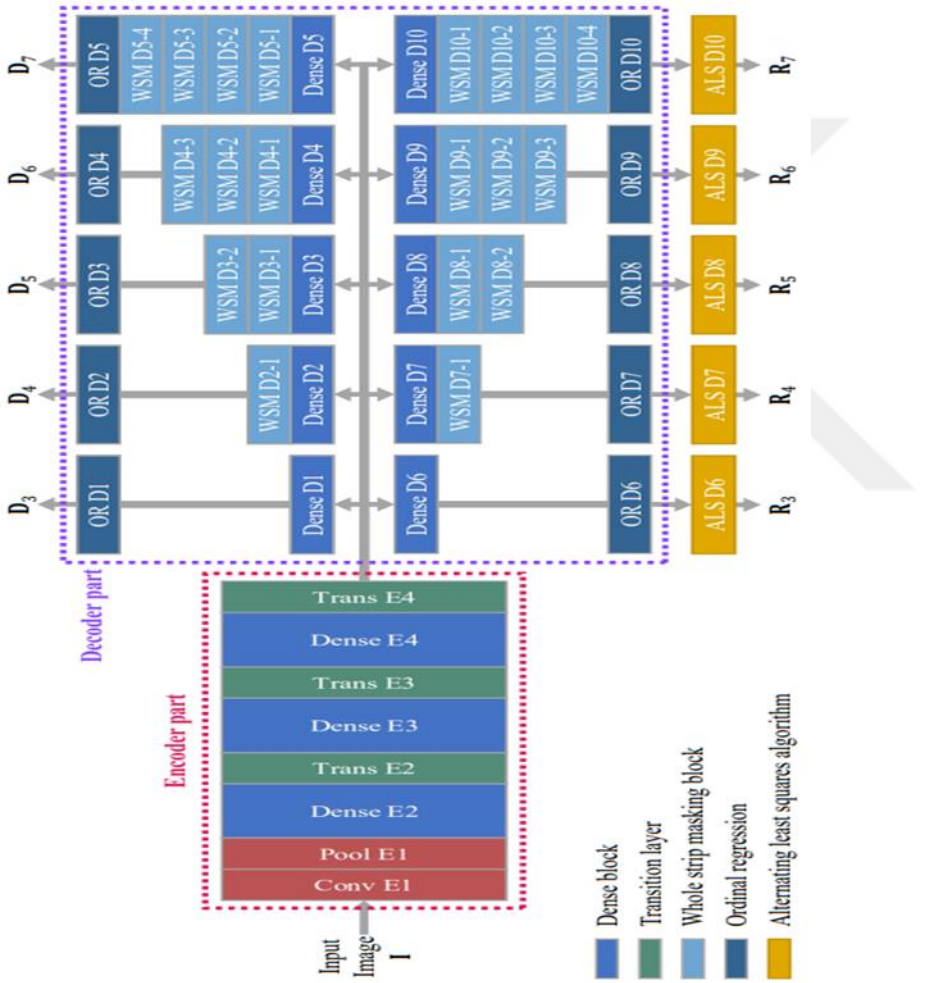

Figure 3.17. The suggested depth estimation network's structure. Up to 10 decoders can be utilized, as indicated above. In the default setup, the five decoders (D3, R3, R4, R5, R6) are used. WSM is for Whole Strip Masking (Heo et al., 2018), OR stands for Ordinal Regression, and ALS stands for Alternating Least Squares (Lee and Kim, 2019).

3.6.3. Reconstruction of a Relative Depth Map

The ratio of two depths is called relative depth. The relative depths of all pixel pairs can be estimated; however, this process is complex since $(2^n \times 2^n) \times (2^n \times 2^n) = 24^n$ pairs should be taken into account in Dn. To decrease the complexity, for each pixel in Dn, Lee estimated the depth ratios for the neighboring 3×3 pixels only, reducing the number of pairs to $3^2 \times 2^{2n}$. Also, these neighboring 3×3 pixels are chosen from D_{n-1}, rather than D_n, as shown in Figure 3.18. This is advantageous since each depth in Dn is compared to a bigger region for a certain number of comparisons. The Alternating Last Squares (ALS) technique is used to recreate unexplained relative depths.

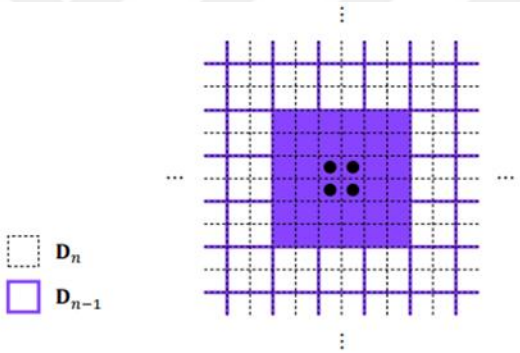

Figure 3.18. Each depth in D_n, shown by a dot, is compared to the depths of the 3×3 nearest pixels in D_{n-1}, represented by purple squares, to determine relative depths (Lee and Kim, 2019).

Lee used Singular Value Decomposition (SVD) when there were estimation errors. It is known that:

$$\acute{P}_{n,n-1} = \sigma_1 u_1 v_1^T \tag{3.11}$$

The greatest singular value is σ_1, and the related singular vectors are u_1 and v_1.

Lee employed the ALS algorithm (Koren et al., 2009) as follows: considered S represent the set of positions (r, c) in $P_{n,n-1}$, where the decoder estimates the relative depths. Let p and q be vectors with sizes of 2^{2n} and 2^{2n-2}, respectively. Then alternatively repeat the next two stages.:

$$q \leftarrow argmin_q \sum_{(r,c)\in s} \left(p(r)q(c) - P_{n,n-1}(r,c)\right)^2 \quad (3.12)$$

$$p \leftarrow argmin_p \sum_{(r,c)\in s} \left(p(r)q(c) - P_{n,n-1}(r,c)\right)^2 \quad (3.13)$$

Each step ensures that the convex condition is met, and the closed-form solution for q or p is quickly found. As a result, the algorithm generates concurrent solutions p˜ and q˜, yielding the approximation.

$$\tilde{p}_{n,n-1} = \tilde{p}\tilde{q}^T \quad (3.14)$$

Lee reconstructed R_n's relative depth map by normalizing and reshaping the left vector p˜. Figure 3.19. shows the process of refilling a sparse $P_{n,n-1}$ and restoring a relative depth map R^n.

An ordinary depth map robustly reconstructs the overall depth distribution, while relative depth maps are more usable for estimating fine detail. Combining these maps with multiple resolutions permits an accurate depth map that carries the advantage of these component maps. Finally, Lee optimized the network utilizing the Nestrov method (Gaviraghi et al., 1983).

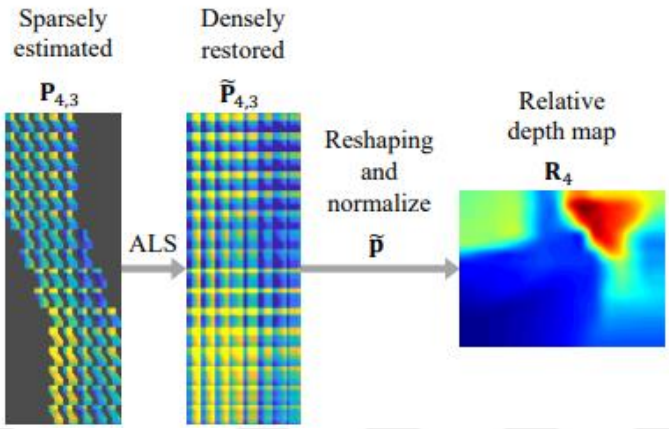

Figure 3.19. A sparse comparison matrix $P_{4,3}$ was converted to a dense matrix $\tilde{P}_{4,3}$ using the ALS method. $\tilde{P}_{4,3}$ is then reshaped and normalized to provide R_4, a relative depth map (Lee and Kim, 2019).

3.7. Fully Convolutional Residual Networks for Deeper Depth Prediction

By introducing an fully convolutional architecture with residual learning to resolve the confusing mapping between monocular pictures and depth maps, Laina et al. (Laina et al., 2016) proposed a novel technique for estimating the depth map of a single image, which allows dense output maps with higher resolutions while using fewer parameters and training with a single order of data. Further, Laina proposed a better efficient schematic of evolutions and combined them with the residual learning concept (He et al., 2015) to make up-projection blocks for efficient upsampling feature maps.

3.7.1. CNN Architecture

The receptive field is an essential feature of architectural structure because there are no complete explicit connections in the fully convolutional neural network for depth map prediction. As illustrated in Figure 3.20, Residual Network (ResNet) (He et al., 2015) batch normalization, as well as skip layers that skip two or more convolutions and are summed to their output (Ioffe and Szegedy, 2015) after each convolution.

It is feasible to design even deeper networks without deterioration or disappearing gradients by following the ResNet structure. ResNet-50 catches input sizes of 483 × 483, which is large enough to capture the complete input image even at higher resolutions.

Laina selected the input of 304 * 228 pixels (as in (Eigen et al., 2015)) and indicated that the output map was at about half the resolution of the input. When the last pooling layer is removed, the final convolutional layers provide 2048 feature maps with a spatial resolution of 10 8) pixels, according to the input size and architecture.

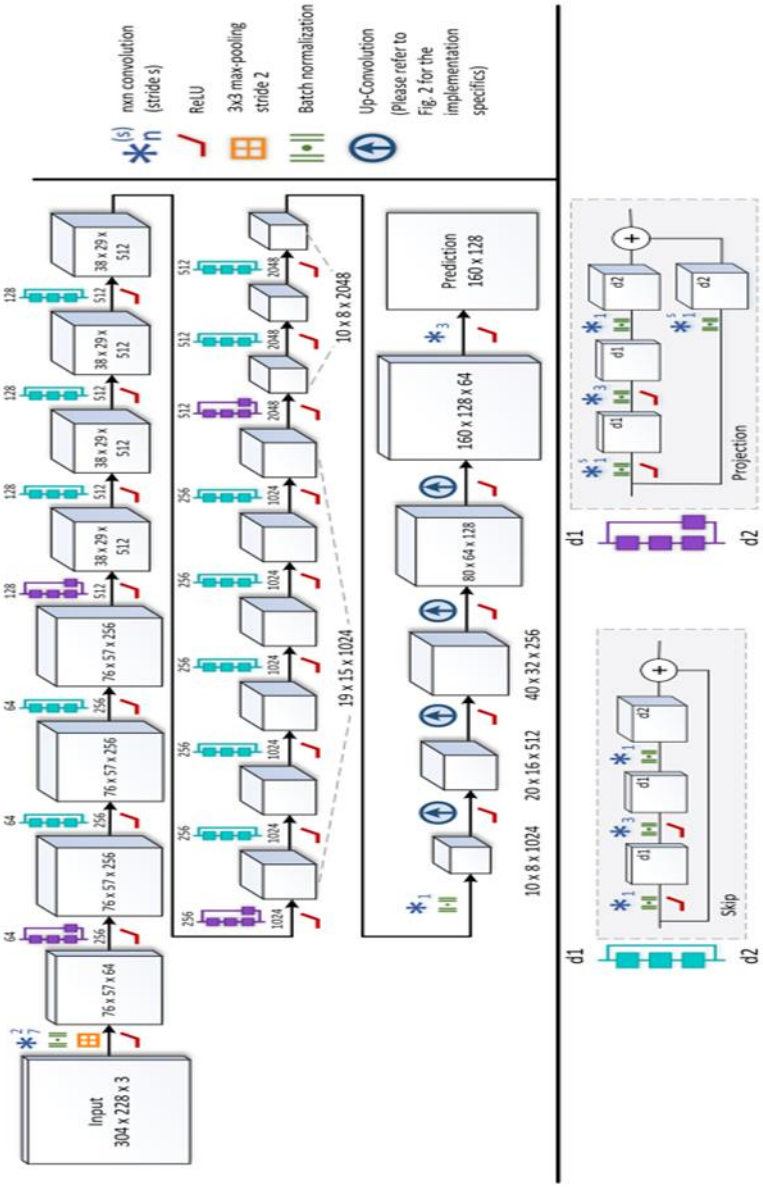

Figure 3.20. It demonstrates that the suggested design is based on ResNet-50. Laina replaced the completely connected layer, which was part of the original architecture, with new upsampling blocks, resulting in an output resolution that is about half that of the input (Laina et al., 2016).

The Laina network's initial section is based on ResNet-50 and is set up with pre-trained weights. Through a sequence of unpooling layers and convolutional layers, the second component of the network instructs the network to learn to scale up. The dropout is performed after a series of upsampling blocks, followed by a final convolutional layer that leads to the prediction.

3.7.2. Blocks for Up-Projection

It is the reverse action of pooling, improving the spatial resolution of feature maps (Shelhamer et al., 2017; Dosovitskiy et al., 2014; Zeiler et al., 2011). Laina tweaked (Dosovitskiy et al., 2014)'s strategy for implementing unpooling layers to duplicate the size by mapping each input into the top-left corner of a 2 × 2 (zero) kernel.

Following every one of these layers is a 5 x 5 convolution applied to multiple non-zero elements at each position and followed by ReLU activation. This block was dubbed "up-convolution" by Laina. Moreover, layered four of these up-convolutional blocks, resulting in a 16-fold increase in the size of the smallest feature map, resulting in the best memory-resolution trade-off. Additionally, upsampling res-blocks were created by extending basic convolutions utilizing an inverse idea (He et al., 2015).

Laina's concept was to add a projection connection from the lower resolution feature map after the up convolution and then a primary (3 × 3) convolution. But due to various sizes, the small map requires to be up sampled with another convolution on the projection unit, whereas since unpooling only requires to be used once for both units, therefore applied 5x5 convolutions individually to two units. Laina called the new upsampling block up-projection upward because it develops the idea of the connection of the projection (He et al., 2015) to the up-convolutions.

The up-projection block sequence allows high-level information to be transferred across the network more effectively while gradually expanding the size of the feature maps. This allows the whole convolutional coherent network for depth

prediction to be built. The differences between the up-convolutional and up projection blocks are shown in Figure 3.21.

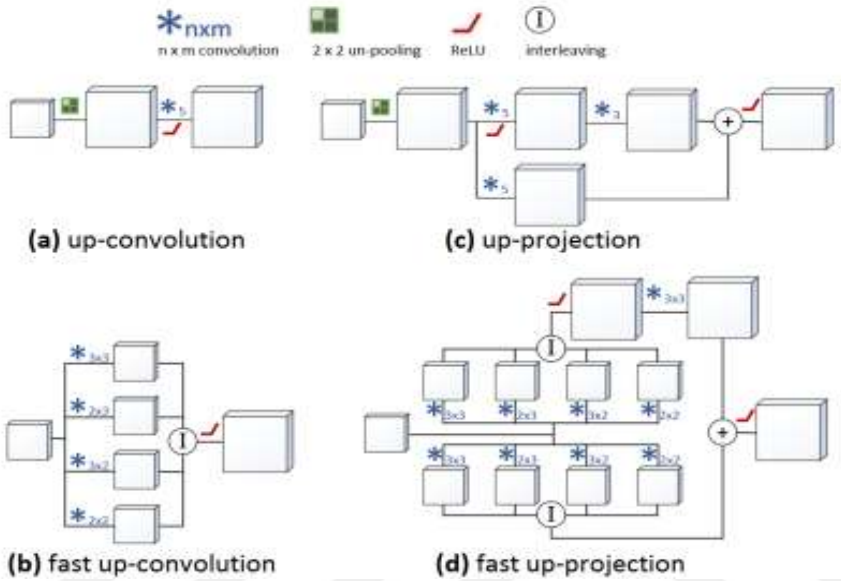

Figure 3.21. From up-convolutions to up-projections, we've come a long way. (a) Conventional up-convolution. (b) An up-convolution that is similar but quicker. (c) A unique residual logic-based up-projection block. (d) A more efficient variant of (c) (Laina et al., 2016).

3.7.3. Fast Up-Convolutions

Fast Up-Convolutions is to reformulate the up-convolution process to make it more effective, which reduces the training time for the entire network by about 15%. Because 75% of the resultant feature maps include zeros after unpooling, the following (5 × 5) convolution focuses on the zeros that can be avoided using the adjusted formula, as illustrated in Figure 3.22. The original feature map is unpooled (top center) and then convolved by a (5 × 5) filter in the upper left.

Only specified weights are multiplied by probable non-zero values in the unpooled feature map, which relies on the position (the bounding boxes of red, blue, purple, and orange) of the (5 × 5) filters.

In figure 3.22, these weights are separated into four non-overlapping groups, denoted by the distinct colors A, B, C, and D. Using a variety of filter combinations, Laina arranged the original (5×5) filter into four new filters with sizes (a) 3×3, (b) 3×2, (c) 2×3 and (d) 2×2.

Finally, The loss function used as the standard function for improvement in regression issues is the loss of L_2, which decreases the Euclidean square rule between the predictions y˜ and the ground truth y: $L_2(\tilde{y} - y) = \|\tilde{y} - y\|_2^2$. Laina found that using the inverse of Huber (berHu) (Owen, 2006; Zwald and Lambert-Lacroix, 2012) B delivers a lower final error than L_2 as a loss function.

$$B(x) = \begin{cases} |x| & x \le c \\ (x^2 + c^2)/2c & x > c \end{cases} \quad (3.\ 15)$$

Berhu's loss is $L_1(x) = |x|$, when $x \in [-c, c]$ is equal to L_2 is outside this range. BerHu achieves a good balance between the two criteria in the problem; it gives much weight to samples/pixels with much residue.

3.8. Depth Map Segmentation

This work also presents a method for segmenting a depth map. The proposed technology classifies pixels into three categories based on their distance from the camera (near, far, and faraway). A given depth map for the NYU v2 image is obtained using Kinect, and the depth map is predicted from the three algorithms, and by using three thresholds, we segment the depth map image.

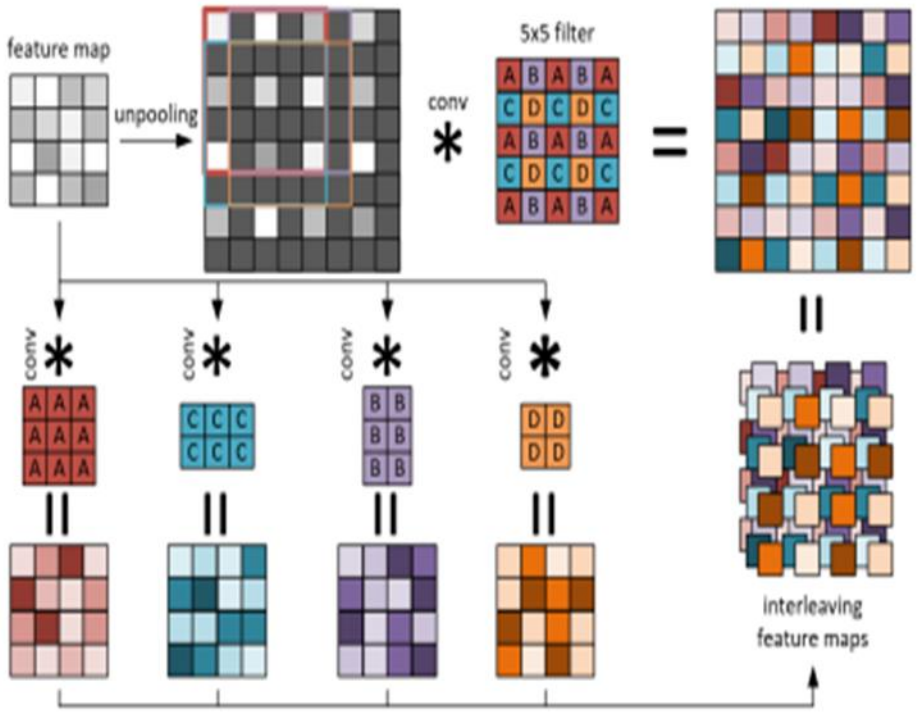

Figure 3.22. Up-convolutions that are faster: Common up-convolutional steps in the top row: The feature map is doubled in size, the gaps are filled with zeros, and the convolution (5x5) for this map is filtered. Certain sections of the filter (A, B, C, D) are multiplied by non-zero values depending on the filter's location (Laina et al., 2016).

4. RESULTS

4.1. Metrics for Evaluation

The outcomes predicted by the models were analogized with the ground truth images by applying the evaluation metrics compared with the ground truth images in order to evaluate their performance.

Selected error metrics take into account the global statistics of a predicted depth Y' and the ground truth Y with N depth pixels. Apart from visual assessments of depth maps or projected 3D points, the following error measures are only utilized in recent relevant articles.

The Mean Squared Error (MSE) is a well-known regression error statistic. The mean or average squared discrepancies between predicted and expected target values in a dataset are used to calculate the MSE.

The relative per-pixel error is calculated using the absolute relative (Rel) difference error, which is linearly proportional to the distance. In other words, a 0.1 m error at 1 m is penalized in the same manner as a 1 m error at 10 m is penalized.

In contrast, the Root Mean Square (RMS) error equally punishes an error of 0.1 m at both depths. The difference between the predicted and actual values is used to calculate the Root Mean Squared Logarithmic (log RMS). Log RMS is robust for outliers as small and large errors are treated equally. If the predicted value is smaller than the actual value, the model is punished more, whereas if the projected value is more than the actual value, the model is penalized less. Due to the log, it does not punish big mistakes. As a result, the model penalizes underestimation more than overestimation. This can be useful when we do not mind overestimation but do not want to underestimate it.

The threshold error, determines the fraction of pixels for which the relative difference between prediction and ground truth depths is less than a given threshold (thresholds are generally set to 1.25, 1.25^2, and 1.25^3), as indicated in Table 4.1.

Table 4.1. The first four metrics are particular error measurements, therefore lower is preferable; but, for the Accuracy measure, bigger is better.

Evaluation Metric	Definition	Equation
Mean Square Error (MSE)	The mean square of the gap between actual and anticipated values.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i')^2$ Y_i is the i^{th} actual value, Y_i' is the corresponding anticipated value, n = the total amount of observations (3. 16)
Root Mean Square (RMS)	The differences between values predicted by a model or an estimator and the values observe	$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i')^2}$ (3. 17)
Absolute Relative (Rel)	The performance of a predictive model	$Rel = \frac{\sum_{j=1}^{n}\lvert Y_{ij}' - Y_j\rvert}{\sum_{j=1}^{n}\lvert Y_j - P\rvert}$ (3. 18) Y_{ij}' is the value anticipated by the individual model *i* for record *j*, Y_j is the goal value for record *j*, The formula gives *P* formula: $P = \frac{1}{n}\sum_{j=1}^{n} T_j$ (3. 19)
Logarithmic Root Mean Square (log RMS)	The Root Mean Square The difference between the log-transformed anticipated and actual numbers is the error.	$\log RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(Y_i + 1)' - \log(Y_i + 1))^2}$ (3. 20)
Accuracy (max ratio)	percentage (%) of predicted depth map	Max Ratio=$\max\left(\frac{actual}{predicted}, \frac{predicted}{actual}\right) = \delta$ <thr (thr=1.25, thr=1.25^2 , thr=1.25^3) (3. 21)

Table 4.2. A comparison of the three algorithms' performance.

Methods	Error↓ (normalized)				Accuracy↑ (normalized)		
	MSE	RMS	Log RMS	REL	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Lee	0.3025	0.55	0.243	0.173	84.2%	96.8%	98.8%
Laina (pretrain)	0.25	0.50	0.223	0.154	84.8%	97.2%	99.0%
Carvalho (pretrain)	**0.025**	**0.163**	**0.053**	**0.071**	**97.8%**	**99.4%**	**99.8%**

We utilized three metrics to evaluate the segment method rendering: Intersection over Union (IOU), a measure based on the Jaccard Index, a coefficient of similarity for two sets of data, the IOU is equivalent to the area of the overlap (intersection) between the anticipated depth map and the ground-truth divided by the area of their union. IoU = 1 indicates a perfect match, whereas IoU = 0 indicates that neither depth map intersects the other. The IoU comes closer to 1, the better the detection is thought to be. The IoU can only compare the same class's ground-truth and predicted depth maps. A metric can be more or less restricted in recognizing detections as right or wrong by defining an IoU threshold. An IoU threshold closer to 1 is more restricted since it demands near-perfect detections, whereas an IoU threshold closer to but not equal to 0 is more flexible.

The Dice coefficient (F1 score is another name for it) is calculated by dividing the area of the intersection of A and B by the total area of A and B. The dice coefficient should not exceed one. The range of a dice coefficient usually is 0 to 1. Pixel accuracy, it is the percentage of pixels in the image that are perfectly categorized. as shown in Table 4.4.

These assessment metrics are derived from the model's output, which is divided into four categories: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), which are items of the confusion matrix as shown in Table 4.3.

Table 4.3. A binary classification system's Confusion Matrix

	Model Output	
Ground Truth	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

The confusion matrix can show the performance of the classification model in tabular form by displaying the number of positive or negative cases that are

correctly or incorrectly predicted. The rows in the table equal the ground truth, and the columns equal the model's predictions. The cells in the table equal the number of items corresponding to each prediction set of the ground truth model. The True Positive (TP) in the confusion matrix indicates the number of positive events that were rightly predicted as positive. Also, False Positive (FP) indicates the number of events expected as positive when in fact they are negative. The True Negative (TN) in the confusion matrix indicates the number of negative samples rightly predicted as negative. Also, False Negative (FN) shows the number of samples that are predicted as negative when in fact, they are positive.

Table 4.4. Metrics to evaluate segmentation

Evaluation Metric	Definition	Equation
Pixel Accuracy	It is the percentage of pixels in an image that are correctly categorized.	Pixel Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ (3. 22)
Intersection-Over-Union	IoU: The overlapping area between the anticipated segmentation and the ground truth is divided by the union area between the anticipated and ground truth.	IoU=$\frac{Area\ of\ Overlap}{Area\ of\ Union}$ (3. 23)
Dice Coefficient	Dice Coefficient is double the number of items in common between the two sets divided by the whole number of items in each set.	DC =$\frac{2*Area\ of\ Overlap}{Total\ number\ of\ pixels}$ (3. 24)

Table 4.5. A. Performance comparison of the segmentation method for near area

Depth Map	Pixel Accuracy	Intersection-Over-Union	Dice Coefficient
Lee	55%	35%	50%
Laina	**90%**	**73%**	**83%**
Carvalho	60%	60%	74%

Table 4.5. B . Performance comparison of the segmentation method for far area

Depth Map	Pixel Accuracy	Intersection-Over-Union	Dice Coefficient
Lee	50%	35%	50%
Laina	**85%**	**69%**	**81%**
Carvalho	69%	65%	78%

Table 4.5. C . Performance comparison of the segmentation method for faraway area

Depth Map	Pixel Accuracy	Intersection-Over-Union	Dice Coefficient
Lee	45%	26%	41%
Laina	**78%**	69%	80%
Carvalho	77%	**70%**	**82%**

4.2. Parameter Optimization

Training a deep neural network involves a lot of challenging tasks such as defining a loss function, choosing an optimization algorithm based on gradient descent to be used and preventing numerical instability problems such as gradient fading and exploding gradients.

Carvalho took 224x224 random crops from the original images and applied a 50% probability of horizontal flip. Tests are admitted utilizing the full-resolution image. In Carvalho's method (Carvalho et al., 2019) dropouts were added (Srivastava et al., 2014) to increase generalization, the first four convolutional layers of the decoder were regularized with a probability of 0.5. For learning rate, weight decay, and momentum to 10^{-2}, 10^{-4} , and 0.9, respectively.

Lee's method (Lee and Kim, 2019) optimized the network parameters utilizing the Nestrov method (Gaviraghi et al., 1983). Also, Lee valid-cropped the test images to spatial resolution 427 × 561. Lee set the learning rate, weight decay, and momentum to 10^{-5}, 10^{-4}, and 0.9, respectively, at the start. Lee also tweaked the learning rate based on the shifted cosine function's repetition (Loshchilov and Hutter, 2016; Huang et al., 2017).

Laina cropped the image size conforming to the network trained for input size 304×228. Laina's method (Laina et al., 2016) introduced the reverse Huber loss (Zwald and Lambert, 2012) that is especially suited for the task at hand and driven by the value distributions generally present in the depth maps. Laina trained the model with a batch size of 16 for approximately 20 epochs. And for the starting learning rate is 10^{-2} for all layers, which slowly reduce every 6-8 epochs, when observing plateaus; momentum is 0.9.

The NYU v2 dataset was split into two parts: training and testing, with 55% used for training and 45% for testing. The number of times the learning algorithm passes over the training set is referred to as the epoch, and the maximum number of the epoch was set Carvalho's method, Lee's method and Laina's method to 90, 40, and 20 for all the training, respectively. We can see a sample of the comparison results in Figure 4.1.

Figure 4.1. Depth Map algorithms comparisons using NYU v2 dataset. In Laina's method, only in the first row, we put the original image, and the rest are processed (brightness, contrast) adjustment for better viewing.

Figure 4.2. Depth Map segmented comparisons using NYU v2 dataset

5. DISCUSSION

First of all, We compare the three methods with each other visually. Seven sample test images are used to make the visual comparisons.

The number of instances from which a network may learn and its performance when viewing comparable models are inextricably linked. Using all-in-focus or defocused images, we study the prediction error per depth range as well as the link between the depth data distribution and the prediction error.

As we can see in figure 5.1, In Carvalho's method, we noticed accurate reconstructions of the depth map, especially in the first row; despite having several shelves at different distances from the camera, the result of the depth map was good. At the same time, the result in the second row was wildly inaccurate for close distances. This indicates that this method deals with long distances and bright images very well and moderately with close distances and sharp angles.

The third sample is (table with chairs), we can see that the image contains very close, far, and faraway pixels; Carvalho's method best represented the depth map, where we see the far and near pixels.

The fourth row shows the image containing near and far pixels; in the predicted depth map, we can see most of the far and near pixels in the depth map.

In the fifth row, Carvalho's method was the best in representing all the image pixels. In the sixth row, this image has flat surfaces with good lighting, and as it turns out, Carvalho's method was well represented.

In the seventh row, we can see different surfaces, distances, and colors; Carvalho's method clearly shows the depth map.

RGB Image Ground Truth Carvalho Method

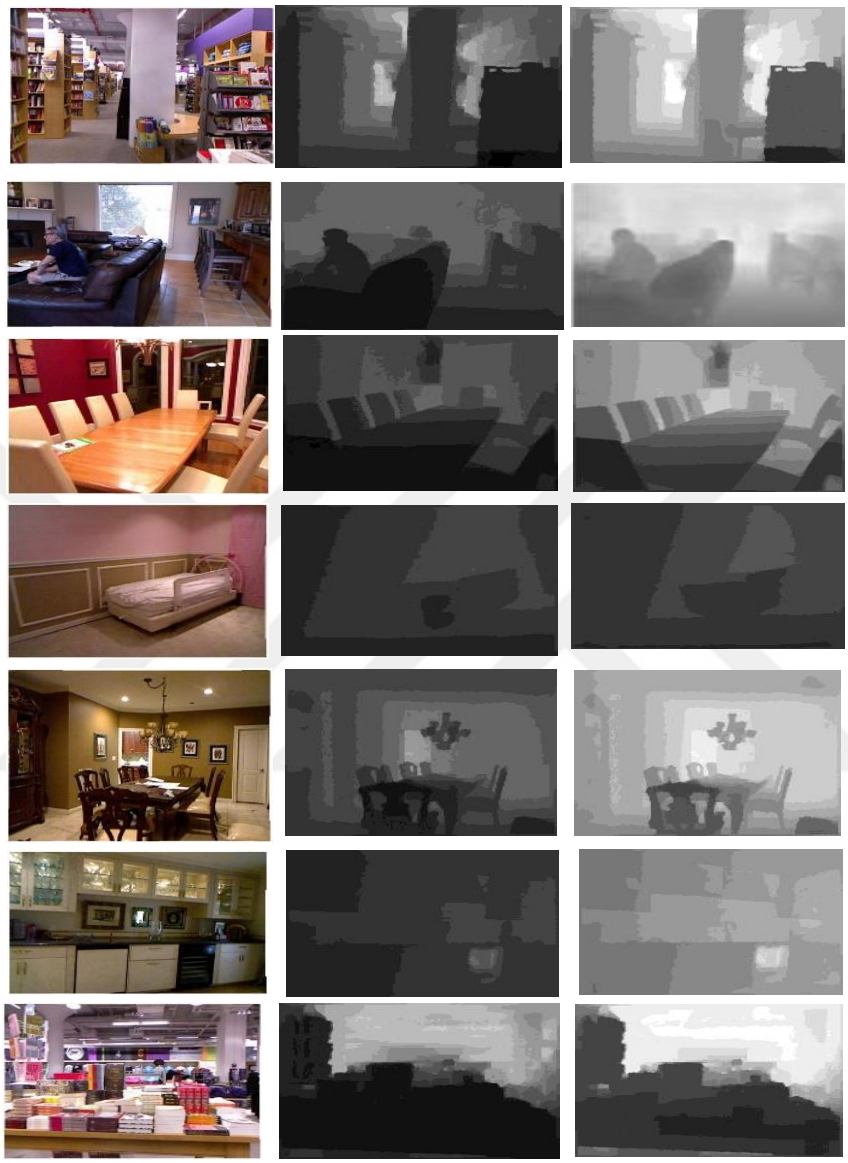

Figure 5.1. Predicted Depth Map for Carvalho's algorithm utilizing NYU v2 dataset

As we can notice in figure 5.2. In Laina's method, we noticed good reconstructions of the depth map. However, it handles the near distances perfectly; the first row, the predicted depth map, does not notice all the shelves, especially the far shelf. In the second row, the representation of the chairs and the seated man was very close to the ground truth.

The third sample is (table with chairs); we can see that the image contains very close, far, and faraway pixels. Laina'sLaina's method represented the depth map better than Lee'sLee's method; we can see most of the pixels in the depth map, at the same time, we do not notice the white color of the distant pixels.

The fourth row shows the image containing near and far pixels; Laina'sLaina's method did not represent most of the pixels in the depth map.

In the fifth row, Laina'sLaina's method, the representation was better, but it contained some noise and overlooked the library's details and the last chair.

In the sixth row, this image has flat surfaces with good lighting, and as it turns out, Laina'sLaina's method was well represented.

We can see different surfaces, distances, and colors; the depth map is clearly represented in the Laina method.

RGB Image Ground Truth Laina Method

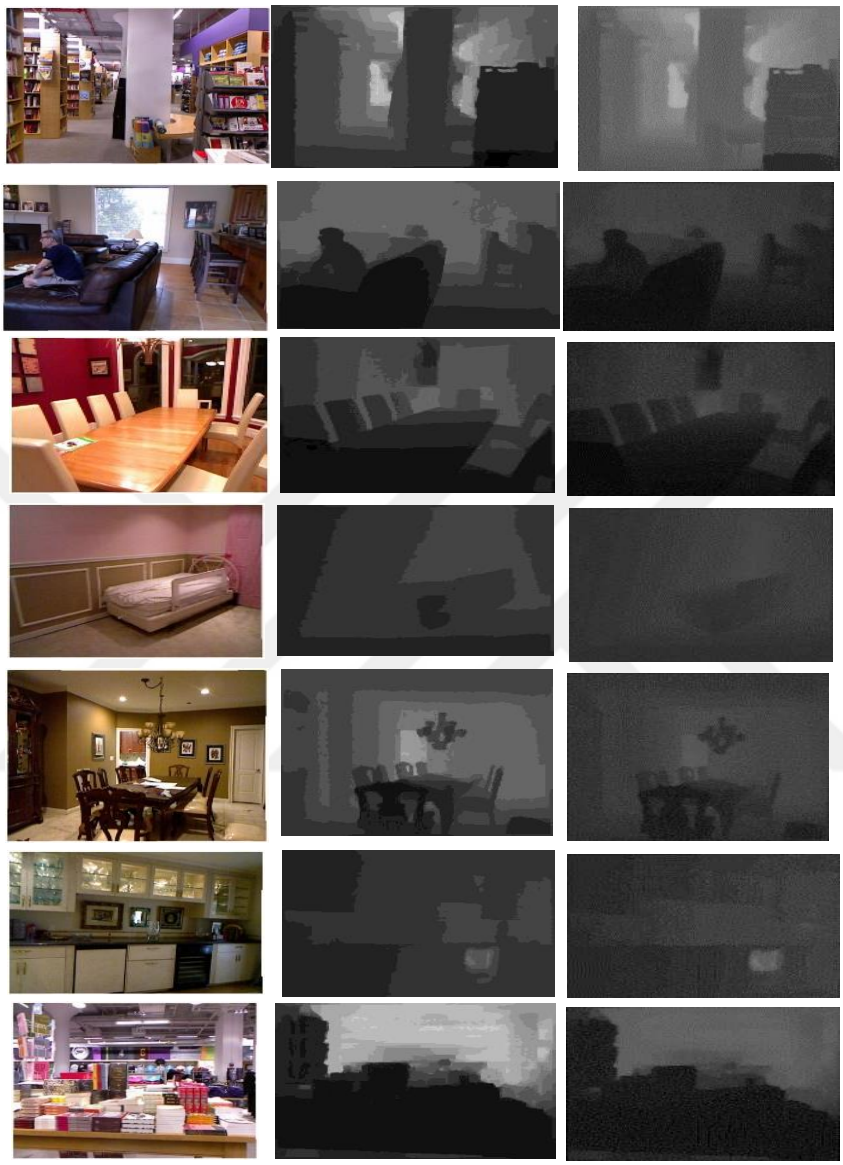

Figure 5.2. Predicted Depth Map for Laina's algorithm using NYU v2 dataset

In Lee's method, we noticed acceptable reconstructions of the depth map. However, it seems that it handles just the near distances; as shown in figure 5.3, the first and second row, the predicted depth map did not notice the far-distance pixels at all, and for the near pixels, it was acceptable. Still, for all distances, the representation was blurred. Furthermore, by calculating relative depth maps, incorporating order information and conventional depth maps, Lee's technique properly reveals the depth ordering of pixels.

The third sample is (table with chairs); we can see that the image contains very close, far, and faraway pixels. Lee's method represented the depth map acceptably because it did not notice the far points well. We cannot see the chair at the end of the table, also the spaces between the chairs to the left of the table.

The fourth row shows the image containing near and far pixels; Lee's method did not represent precise edges or angles.

In the fifth row, we can see Lee method did not represent most of the image pixels; although from image contain a good light source, we cannot see all the chairs, the door and the chandelier hanging from the ceiling completely.

In the sixth row, this image has flat surfaces with good lighting, and as it turns out, Lee's method represented the depth map with blurry and without any sharp edges.

In the seventh row, we can see different surfaces, distances, and colors; the depth map is clearly represented.

RGB Image Ground Truth Lee Method

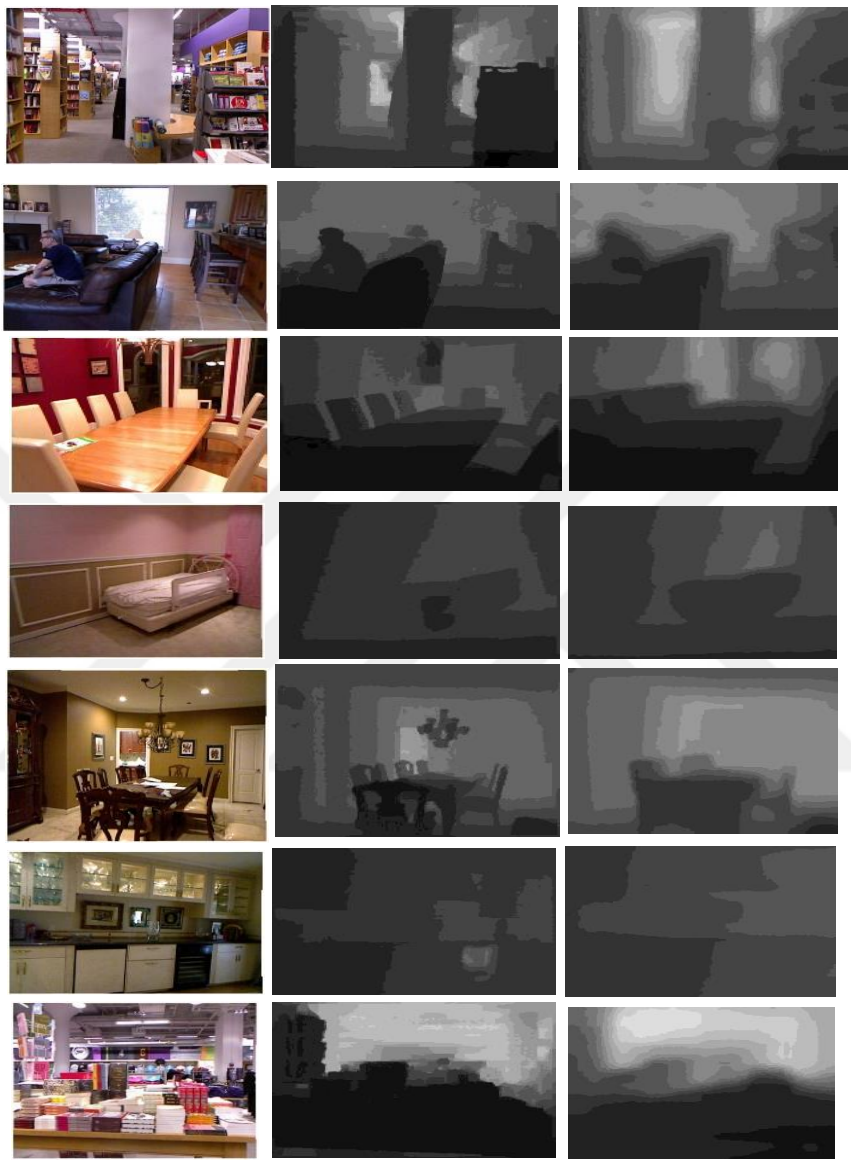

Figure 5.3. Predicted Depth Map for Lee's algorithms using NYU v2 dataset

In figure 5.4, we zoomed in on three samples to see the accuracy of the depth map estimation for the three methods. We can see all the details in the ground truth depth map, while in Lee's approach, we cannot see most of the details as it does not have any visible edges of any object in the images. In Laina's method, nearby pixels are represented flawlessly, but most distant pixels cannot be seen because they do not represent most of the shelves behind the chair in the first row. The depth maps still include noise, even though the Laina model enhances edge sharpness and body description considerably. Also, Carvalho's method represents the best depth map for both near and far pixels in these images.

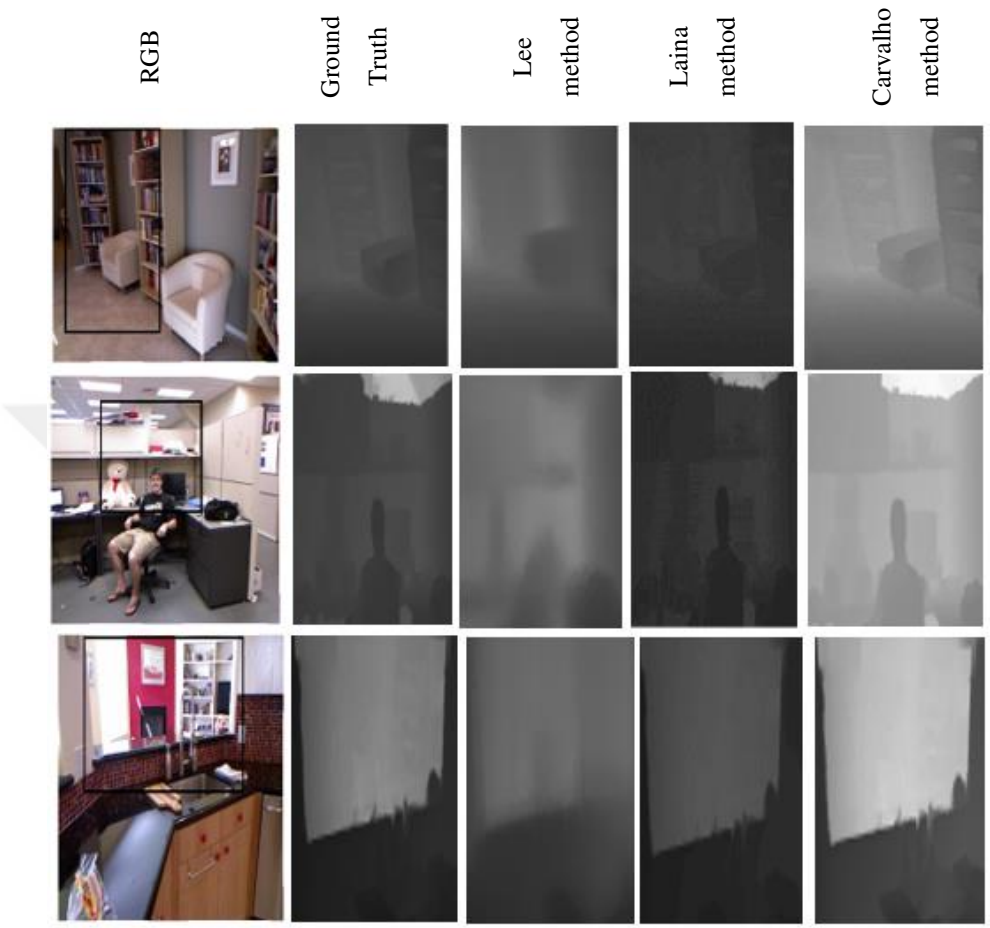

Figure 5.4. Predicted Depth Map for three algorithms using NYU v2 dataset (zoomed in images).

Quantitative evaluations have been performed using five well-known metrics: MSE, RMS, Log RMS, REL, and Accuracy. Several findings can be drawn from Table 4.2. First, there is a significant difference in depth estimate performance when employing out-of-focus images rather than all-in-focus images. Second, D3-Net perform better than other networks (ResNet-50, Encoder-Decoder), this is also shown in figure 5.1, without the need for a complex scene model or precise blur calibration. Third, Lee's method was the worst on all metrics. Fourth, Laina's approach does not explicitly depend on visual elements to estimate depths; therefore,

it might be used in situations with low-textured surfaces like walls, floors, and other structures that are common in interior contexts. Even though Laina et al. (Laina et al., 2016) yield smaller errors than Lee et al. (Lee and Kim, 2019), their depth maps look noisier. According to the quantitative results, Carvalho's method is superior to the other methods for the five metrics; moreover, utilizing defocus blur, the error repartition is additional akin to the quadratic rise in error with depth, which is the typical error repartition of inactive depth estimation.

Table 5.1. Comparing Run Times for the three algorithms on our setup.

Algorithm	Run time(sec.)
Lee's Algorithm	93.5
Laina's Algorithm	55
Carvalho's Algorithm	110

As shown in Figure 5.3, we segmented the predicting depth map images into three regions: near (yellow), far (blue), and faraway (green). We see that Lee's method gives an awful result in the segmentation, however it was better in near area than the other areas. While Carvalho's method gives a good result, espesially for the faraway area, but Laina's approach gives the best result in general.

As for the quantitative comparison, it has been done using three well-known metrics: Pixel Accuracy, IoU, and DC. According to these metrics, Laina's method is the best in near and far areas, but Carvalho's method is the best in faraway areas.

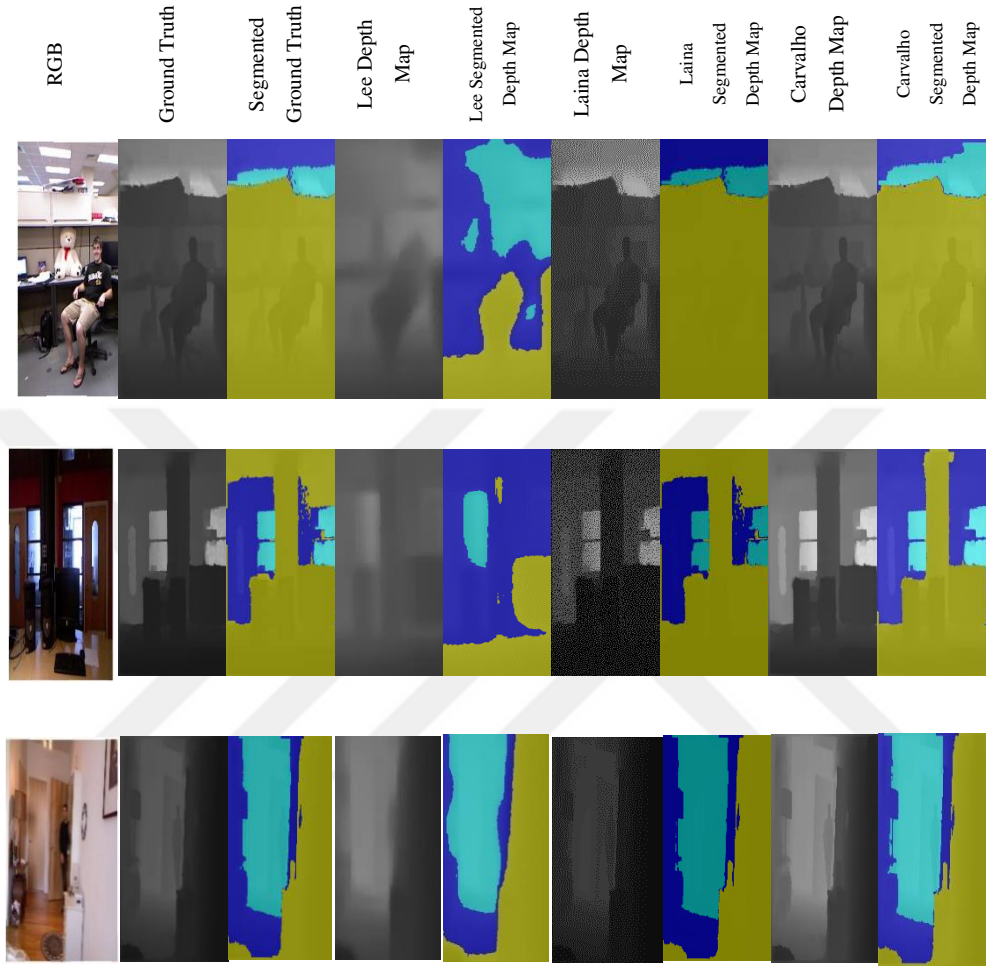

Figure 5.5. Depth Map segmented for three algorithms using NYU v2 dataset

According to the quantitative comparison results for depth map estimation, Carvalho's method appeared to generate the most accurate depth maps for the five metrics but turned out to be a long time intensive. We got the depth map in about 20 hours where; which is considered successful with most interior images. The depth maps created with out-of-focus images are crisper than those created with in-focus ones. Laina's method performed moderately well for depth map estimation. We got the result in about 10 hours as it succeeded in representing the near pixels from the

camera very well, unlike the far pixels. Lee's method is the worst representation of the depth map, both for the far and near points, as most of its results were blurry; although it is the only network trained on NYU v2, it did not require a long computational time.

For the quantitative comparison results of the depth map segmentation, Liana's method was the best for the three measures in near and far areas. Even though defocusing provides network-local depth information resulting in better depth map segmentation, it did not beat Laina's way except in faraway areas. As for Lee's method, it completely fails to segmentation as its results can never be used.

6. CONCLUSION AND FUTURE WORKS

6.1. Conclusion

The goal of this thesis was to examine and implement known techniques for estimating a depth map from a single image using the NYU v2 dataset. Moreover, we segment the predicted depth map into three areas.

Unlike standard CNN algorithms, Laina's approach requires a multi-step procedure to correct their initially imprecise depth estimates. Following residual learning, Laina's technique consists of a robust, single-scale CNN architecture. Because the network is fully convolutional, including up-projection relations, the number of learning parameters and training samples required is considerably reduced. The fully convolutional model suggested by Laina significantly increases edge sharpness and structure description in the expected depth maps. Even though the depth predictions are generated from a single model that has been trained end-to-end without any additional post-processing procedures, they have a noticeable visual quality. The single network generalizes better than previous CNN depth estimation techniques, overcoming the coarseness problem.

Lee's method uses relative depth maps. In figure 3.17, Lee's technique generates conventional depth maps D_n and relative depth maps R_n using up to 10 decoders. First, it is an encoder-decoder network with numerous decoder blocks for predicting relative and ordinary depths at various scales. Second, Lee used the ALS technique to reconstruct a complete relative depth map from selectively estimated data, reducing the complexity. Experiments have shown that relative depth maps are more effective than standard depth maps at preserving a scene's depth ordering.

Carvalho's method used a deep learning methodology to investigate the effect of defocus blur as a cue in monocular depth map estimation. This demonstrates that blurry images perform better than all-in-focus images without the need for scene modelling or blurring calibration. Furthermore, the deep network's simultaneous use of defocus blurring and geometrical structural information on the

image eliminates the traditional constraints of Depth from Defocus with a traditional camera, like dead zones and depth ambiguity.

The pre-trained algorithms with another dataset have succeeded in achieving better results than the pre-trained algorithm on the NYU v2 dataset. This has not been a straightforward journey because of the inner difficulties of debugging deep neural networks and handling big data. Experiments are performed using 1449 pairs of images. A comparison was made between the three methods and the ground truth; the methods are examined and compared visually and quantitatively.

According to the visual results, it is seen that:

- The method trained and tested on the same dataset does not have to give better results than methods trained and tested on a different dataset.
- Carvalho's method that uses defocus blur as a cue for depth map estimation achieved better results than other methods, as it was pre trained on another dataset. And its results were good in the segmentation especially in faraway area.
- Laina's method achieved relatively good results, especially with the pixels near the camera and less accurate results with the far pixels. also, this method was pre-trained on another dataset, however, in the segmentation process, it produced the best results.
- Lee's method achieved feeble results compared to the previous two methods, where most of the results were a blur. As for its results in the segmentation, it was very weak.

Five different quantitative metrics are used for quantitative evaluation in-depth map estimation and three metrics for the segmentation process. The conclusions from the quantitative evaluations are as follows:

- Carvalho's method is the most successful in five metrics. The only drawback is that it takes a long time.
- Laina's method is the most successful of the three metrics in the segmentation process just in near and far areas.

6.2. Future works

First, The obtained results open the way for more research into 3D estimation utilizing defocusing and deep learning but with attention to the time taken by the algorithm.

Second, it would be interesting to benchmark the three models with entirely different datasets, where there is more variation in illumination, fine detail, etc.

Third, it will be better during the comparing checking depth and network structure.

Fourth, It is possible to improve Lee's method by replacing the encoding part with pre-trained DenseNet.

Fifth, when looking at the literature review, none of them is talking about shiny surfaces in the image, and this feature can be utilized to calculate the depth map.

REFERENCES

Abbas, O. M. 2015. Neural Networks in Business Forecasting, International Journal of Computer (IJC), 19(1): 114-128. https://ijcjournal.org/index.php/InternationalJournalOfComputer/article/view/483.

Abdul, A., Dawood, M., and Saleh, M. F. 2019. Review Article Image Deblurring Techniques:A Review. 6(3): 94–98. https://www.academia.edu/40363052/Image_deblurring_tecniques

Alrammahi, A., and Radif, M. 2019. Neural networks in Business Applications Journal of Physics: Conference Series, 1294(4): 042007. https://doi.org/10.1088/1742-6596/1294/4/042007

Anwar, S., Hayder, Z., and Porikli, F. 2017. Depth estimation and blur removal from a single out-of-focus image. British Machine Vision Conference 2017, BMVC 2017, 1–12. https://doi.org/10.5244/c.31.113

Badrinarayanan, V., Kendall, A., and Cipolla, R. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12): 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Bahadur, S., Shrestha, R., Sumaharshini, Y., Teja, G. R., and N, K. 2017. Literature Review on Various Depth Estimation Methods for an Image. International Journal of Research -GRANTHAALAYAH, 8–13. https://doi.org/10.29121/granthaalayah.v5.i4racsit.2017.3342

Bihl, T. J., Young II, W. A., and Weckman, G. R. 2018. Artificial Neural Networks and their applications in business. Encyclopedia of Information Science and Technology, Fourth Edition, 6642–6657. https://doi.org/10.4018/978-1-5225-2255-3.ch576

Biswas, J., and Veloso, M. M. 2012. Depth Camera Based Indoor Mobile Robot Localization and Navigation.

https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=2332&context=cs_faculty_pubs

Brownlee, J. 2018. Encoder-Decoder Recurrent Neural Network Models for Neural Machine Translation. Machine Learning Mastery. https://machinelearningmastery.com/encoder-decoder-recurrent-neural-network-models-neural-machine-translation/

Anonymous. 2021. CAFFEMODEL File Extension - What is a caffemodel file and how do I open it. Fileinfo. https://fileinfo.com/extension/caffemodel

Cao, Y., Wu, Z., and Shen, C. 2018. Estimating Depth from Monocular Images as Classification Using Deep Fully Convolutional Residual Networks. IEEE Transactions on Circuits and Systems for Video Technology, 28(11): 3174–3182. https://doi.org/10.1109/TCSVT.2017.2740321

Carvalho, M., Le, B., Trouvé, P., Almansa, A., and Champagnat, F. 2019. Deep depth from defocus: How can defocus blur improve 3D estimation using dense neural networks? Lecture Notes in Computer Science, 11129 LNCS, 307–323. https://doi.org/10.1007/978-3-030-11009-3_18

Carvalho, M., Saux, B. Le, Trouvé-Peloux, P., Almansa, A., and Champagnat, F. 2018. On regression losses for deep depth estimation. https://hal.archives-ouvertes.fr/hal-01925321

Chakrabarti, A., Shao, J., and Shakhnarovich, G. 2016. Depth from a Single Image by Harmonizing Overcomplete Local Network Predictions.

Chakrabarti, A., and Zickler, T. 2012. Depth and Deblurring from a Spectrally-varying Depth-of-Field. http://www.springerlink.com/.

Chen,J.2020.Neural Network Definition. Investopedia. https://www.investopedia.com/terms/n/neuralnetwork.asp

Chen, W., Fu, Z., Yang, D., and Deng, J. 2016. Single-Image Depth Perception in the Wild. Advances in Neural Information Processing Systems, 29. http://www-personal.umich.edu/~wfchen/depth-in-the-wild.

Cheng, C. M., Lin, S. J., Lai, S. H., and Yang, J. C. 2008. Improved Novel View

Synthesis from Depth Image with Large Baseline. http://cv.cs.nthu.edu.tw/upload/projects/31/index.php?project=31

Contributors, W. 2021. Bokeh - Wikipedia. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Bokeh

D'Andres, L., Salvador, J., Kochale, A., and Susstrunk, S. 2016. Non-parametric blur map regression for depth of field extension. IEEE Transactions on Image Processing, 25(4), 1660–1673. https://doi.org/10.1109/TIP.2016.2526907

Delage, E., Lee, H., and Ng, A. Y. 2006. A Dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image. IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06): 2418–2428. https://doi.org/10.1109/cvpr.2006.23

Delbracio, M., Musé, P., Almansa, A., and Morel, J. M. 2011. The Non-parametric Sub-pixel Local Point Spread Function Estimation Is a Well Posed Problem. International Journal of Computer Vision 96(2): 175–194. https://doi.org/10.1007/S11263-011-0460-0

Depth map -Wikipedia. 2021. Wikipedia. https://en.wikipedia.org/wiki/Depth_map

Dosovitskiy, A., Springenberg, J. T., Tatarchenko, M., and Brox, T. 2014. Learning to Generate Chairs, Tables and Cars with Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4): 692–705. https://doi.org/10.1109/TPAMI.2016.2567384

Dunham, H. 2020. Cheat Detection using Machine Learning within Counter-Strike: Global Offensive (thesis). SENIOR INDEPENDENT STUDY THESES. P: 8948.

Eigen, D., and Fergus, R. (2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter: 2650–2658. https://doi.org/10.1109/ICCV.2015.304

Eigen, D., Puhrsch, C., and Fergus, R. 2015. Depth Map Prediction from a Single

Image using a Multi-Scale Deep Network.

Fumo, D. 2017. Types of Machine Learning Algorithms, Towards Data Science. https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

Garg, R., Vijay Kumar, B. G., Carneiro, G., and Reid, I. 2016. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. Lecture Notes in Computer Science, 9912 LNCS: 740–756. https://doi.org/10.1007/978-3-319-46484-8_45

Gaviraghi, B., Schindele, A., Annunziato, M., and Borzì, A. 1983. On Optimal Sparse-Control Problems Governed by Jump-Diffusion Processes. Soviet Mathematics Doklady. https://www.scirp.org/(S(i43dyn45teexjx455qlt3d2q))/reference/ReferencesPapers.aspx?ReferenceID=1893751

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. 2013. Vision meets robotics: The KITTI dataset; 32(11):1231-1237. https://doi.org/10.1177/0278364913491297

Godard, C., Mac Aodha, O., and Brostow, G. J. 2016. Unsupervised Monocular Depth Estimation with Left-Right Consistency. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017: 6602–6611. https://doi.org/10.1109/CVPR.2017.699

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. 2020. Generative adversarial networks. Communications of ACM, 63(11):139–144. https://doi.org/10.1145/3422622

Gupta, A., Efros, A. A., and Hebert, M. 2010. Blocks world revisited: Image understanding using qualitative geometry and mechanics, 6314LNCS(PART4):482–496.https://doi.org/10.1007/978-3-642-15561-1_35

Han, J., Shao, L., Xu, D., and Shotton, J. 2013. Enhanced computer vision with

Microsoft Kinect sensor: A review. IEEE Transactions on Cybernetics, 43(5): 1318–1334. https://doi.org/10.1109/TCYB.2013.2265378

Hassan, M. ul. 2018. VGG16 - Convolutional Network for Classification and Detection. Neurohive. https://neurohive.io/en/popular-networks/vgg16/

He, K., Zhang, X., Ren, S., and Sun, J. 2015. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem: 770–778. https://doi.org/10.1109/CVPR.2016.90

He, K., Sun, J., and Tang, X. 2011. Single Image Haze Removal Using Dark Channel Prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(12): 2341–2353. https://doi.org/10.1109/TPAMI.2010.168

Heath, N. 2021. What is AI? ZDNet. ZDNet. https://www.zdnet.com/article/what-is-ai-heres-everything-you-need-to-know-about-artificial-intelligence/

Heo, M., Lee, J., Kim, K. R., Kim, H. U., and Kim, C. S. 2018. Monocular Depth Estimation Using Whole Strip Masking and Reliability-Based Refinement. Undefined,11208 LNCS:39–55.https://doi.org/10.1007/978-3-030-01225-0_3

MatConvNet. 2014. MatConvNet. https://www.vlfeat.org/matconvnet/

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. 2017. Snapshot Ensembles: Train 1, get M for free. 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings. https://arxiv.org/abs/1704.00109v1

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. 2016. Densely Connected Convolutional Networks. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017: 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Introduction to Python. 1999.https://www.w3schools.com/python/python_intro.asp

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 32nd International

Conference on Machine Learning, ICML 2015: 1, 448–456. https://archive.org/details/arxiv-1502.03167

Jaiswal, S. 2011. Analog Image Processing vs Digital Image Processing - Javatpoint. https://www.javatpoint.com/analog-image-

Jung, H., Kim, Y., Min, D., Oh, C., and Sohn, K. 2017. DEPTH PREDICTION FROM A SINGLE IMAGE WITH CONDITIONAL ADVERSARIAL NETWORKS School of Electrical and Electronic Engineering , Yonsei University , Seoul , Korea Department of Computer Science and Engineering , Chungnam National University , Daejeon , Korea. Icip, 1717–1721. http://cvl.ewha.ac.kr/assets/conference/2017-ICIP-Jung.pdf

Jwalapuram, N. 2020. A Gentle Introduction to PyTorch Library for Deep Learning.https://www.analyticsvidhya.com/blog/2021/04/a-gentle-introduction-to-pytorch-library/

Karaali, A., and Jung, C. R. 2018. Edge-based defocus blur estimation with adaptive scale selection. IEEE Transactions on Image Processing, 27(3): 1126–1137. https://doi.org/10.1109/TIP.2017.2771563

Karsch, K., Liu, C., and Kang, S. B. 2012. DepthTransfer: Depth Extraction from Video Using Non-parametric Sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(11): 2144–2158. https://doi.org/10.1109/TPAMI.2014.2316835

Kendall, A., and Gal, Y. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? Advances in Neural Information Processing Systems, 2017-Decem, 5575–5585. https://arxiv.org/abs/1703.04977v2

Kenton, W. 2020. Risk assessment. Investopedia. Retrieved from https://www.investopedia.com/terms/r/risk-assessment.asp

Kinect- Wikipedia. 2020. Wikipedia. https://en.wikipedia.org/wiki/Kinect

Koch, T., Liebel, L., Fraundorfer, F., and Körner, M. 2018. Evaluation of CNN-based single-image depth estimation methods, 11131 LNCS, 331–348.

https://doi.org/10.1007/978-3-030-11015-4_25

Konrad, J., Wang, M., and Ishwar, P. 2012. 2D-to-3D image conversion by learning depth from examples. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 16–22. https://doi.org/10.1109/CVPRW.2012.6238903

Koren, Y., Bell, R., and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37. https://doi.org/10.1109/MC.2009.263

Kundu, A., Li, Y., Dellaert, F., Li, F., and Rehg, J. M. 2014. Joint Semantic segmentation and 3D reconstruction from monocular video. Computer Vision – ECCV, 703–718. https://doi.org/10.1007/978-3-319-10599-4_45

Ladický, L., Shi, J., and Pollefeys, M. 2014. Pulling Things out of Perspective. In CVF (pp. 89–96). https://openaccess.thecvf.com/content_cvpr_2014/html/

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. 2016. Deeper depth prediction with fully convolutional residual networks. Proceedings - 2016 4th International Conference on 3D Vision, 239–248. https://doi.org/10.1109/3DV.2016.32

Lee, D., Gupta, A., Hebert, M., and Kanade, T. 2010. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. http://www.cs.cmu.edu/~dclee/pub/nips2010.pdf

Lee, J. H., Heo, M., Kim, K. R., and Kim, C. S. 2018. Single-Image Depth Estimation Based on Fourier Domain Analysis. https://openaccess.thecvf.com/content_cvpr_2018/CameraReady/2873.pdf

Lee, J. H., and Kim, C. S. 2019. Monocular Depth Estimation Using Relative Depth Maps. In CFV (pp.9729–9738). https://openaccess.thecvf.com/content_CVPR_2019/html

Lenz, I., Lee, H., and Saxena, A. 2015. Deep Learning for Detecting Robotic Grasps. The International Journal of Robotics Reasearch, 34: 705–724.

https://www.semanticscholar.org/paper/Deep-learning-for-detecting-robotic-grasps-Lenz-Lee/260b98e772c4785fa06a5e8fe1c205eb05ec01e2

Levin, A., Fergus, R., Durand, F., and Freeman, W. T. 2007. Image and depth from a conventional camera with a coded aperture. ACM Transactions on Graphics (TOG), 26(3). https://doi.org/10.1145/1276377.1276464

Levin, A., Weiss, Y., Durand, F., and Freeman, W. T. 2010. Understanding and evaluating blind deconvolution algorithms. 1964–1971. https://doi.org/10.1109/CVPR.2009.5206815

Li, B., Shen, C., Dai, Y., Van Den Hengel, A., and He, M. 2015. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1119–1127. https://doi.org/10.1109/CVPR.2015.7298715

Li, L., and Lin, H. 2006. Ordinal Regression by Extended Binary Classification. Advances in Neural Information Processing Systems, 19. https://papers.nips.cc/paper/2006/hash/019f8b946a256d9357eadc5ace2c8678-Abstract.html

Liu, B., Gould, S., and Koller, D. 2010. Single image depth estimation from predicted semantic labels. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1253–1260. https://doi.org/10.1109/CVPR.2010.5539823

Liu, F., Shen, C., Lin, G., and Reid, I. 2016. Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(10): 2024–2039. https://doi.org/10.1109/TPAMI.2015.2505283

Loshchilov, I., and Hutter, F. 2016. SGDR: Stochastic Gradient Descent with Warm Restarts. 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings. https://arxiv.org/abs/1608.03983v5

Machine learning -Wikipidia 2021.https://en.wikipedia.org/wiki/Machine_learning

Malik, J., and Rosenholtz, R. 1997. Computing Local Surface Orientation and Shape from Texture for Curved Surfaces. International Journal of Computer Vision, 23(2): 149–168. http://web.mit.edu/rruth/www/Papers/1997-SFT.pdf

Meechan, K., Palmquist, D., Schiller, U., Turner, R., Becker, E., and Hodges, T. 2016. Blurring images – Image Processing with Python. https://datacarpentry.org/image-processing/06-blurring/

Mun, J. H., and Ho, Y. S. 2019. Depth Estimation from Light Field Images via Convolutional Residual Network. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 1495–1498. https://doi.org/10.23919/APSIPA.2018.8659639

Ndjiki-Nya, P., Köppel, M., Doshkov, D., Lakshman, H., Merkle, P., Muller, K., and Wiegand, T. 2011. Depth image-based rendering with advanced texture synthesis for 3-D video. IEEE Transactions on Multimedia, 13(3): 453–465. https://doi.org/10.1109/TMM.2011.2128862

Neural Network: Architecture, Components and Top Algorithms | upGrad blog. 2020.https://www.upgrad.com/blog/neural-network-architecture-components-algorithms/

Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., and Hanrahan, P. 2020. Light field photography with a hand-held plenoptic camera. HAL Open Science. https://hal.archives-ouvertes.fr/hal-02551481

Olivier. 2019. How does refocusing in a dual lens work? - Photography Stack Exchange. Photography. https://photo.stackexchange.com/questions/76322/how-does-refocusing-in-a-dual-lens-work

Owen, A. 2006. A robust hybrid of lasso and ridge regression. 59–71. https://doi.org/10.1090/CONM/443/08555

Pentland, A. 1987. A New Sense for Depth of Field. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(4): 523–531.

https://doi.org/10.1109/TPAMI.1987.4767940

Pykes, K. 2020. The Vanishing/Exploding Gradient Problem in Deep Neural Networks.Towards Data Science. https://towardsdatascience.com/the-vanishing-exploding-gradient-problem-in-deep-neural-networks-191358470c11

Qi, X., Liao, R., Liu, Z., Urtasun, R., and Jia, J. 2018. GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 283–291. https://doi.org/10.1109/CVPR.2018.00037

Rani, N., Raj, P., Tech Scholar, M., and Parnami, P. 2016. REVIEW OF IMAGE DEBLURRING TECHNIQUES. International Journal For Technological Research In Engineering, 3(10). www.ijtre.com

Recht, B. 2009. A simpler approach to matrix completion. arXiv.org. https://arxiv.org/abs/0910.0651

Ren, X., Bo, L., and Fox, D. 2012. RGB-(D) Scene Labeling: Features and Algorithms. https://research.cs.washington.edu/istc/lfb/paper/cvpr12.pdf

Residual neural network - Wikipedia. 2021. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Residual_neural_network

Rocha, D. S. 2019. " Estimation of Depth Maps from Mo- nocular Images using Deep Neural Networks " Daniel Sarmiento Rocha.

Roy, A., and Todorovic, S. 2016. Monocular Depth Estimation Using Neural Regression Forest. https://openaccess.thecvf.com/content_cvpr_2016/papers/Roy_Monocular_Depth_Estimation_CVPR_2016_paper.pdf

Saxena, A., Chung, S. H., and Ng, A. Y. 2005. Learning depth from single monocular images. Advances in Neural Information Processing Systems, 1161–1168. http://www.cs.cornell.edu/~asaxena/learningdepth/NIPS_LearningDepth.pdf

Saxena, A., Sun, M., and Ng, A. Y. 2009. Make3D: Learning 3D scene structure

from a single still image. IEEE Transactions on Pattern Analysis andMachine Intelligence, 31(5): 824–840. https://doi.org/10.1109/TPAMI.2008.132

Sellent, A., and Favaro, P. 2014. Which side of the focal plane are you on? IEEE International Conference on Computational Photography, ICCP 2014. https://doi.org/10.1109/ICCPHOT.2014.6831818

Shachmurove, Y. 2002. Applying Artificial Neural Networks to Business, Economics and Finance.

Shelhamer, E., Long, J., and Darrell, T. 2017. Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4): 640–651. https://doi.org/10.1109/TPAMI.2016.2572683

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. 2013. Real-Time Human Pose Recognition in Parts from Single Depth Images. ACM, 56(1): 116–124. https://www.microsoft.com/en-us/research/wp content/uploads/2016/02/BodyPartRecognition.pdf

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. 2012. Indoor Segmentation and Support Inference from RGBD Images. https://doi.org/10.1007/978-3-642-33715-4_54

Silberman, N., Kohli, P., Hoiem, D., and Fergus, R. 2012. NYU DepthV2 https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Simonyan, K., and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations. http://www.robots.ox.ac.uk/

Srivastava, N., Hinton, G., Krizhevsky, A., and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15, 1929–1958.

https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. 2014. Going Deeper with Convolutions. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Tang, C., Hou, C., and Song, Z. 2013. Defocus map estimation from a single image via spectrum contrast. Optics Letters, 38(10): 1706–1708. https://doi.org/10.1364/OL.38.001706

Taylor, J., Shotton, J., Sharp, T., and Fitzgibbon, A.2012. The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation. http://www.cs.toronto.edu/~jtaylor/papers/cvpr2012.pdf

Trouvé, P., Champagnat, F., Le Besnerais, G., Sabater, J., Avignon, T., and Idier, J. 2013. Passive depth estimation using chromatic aberration and a depth from defocus approach. Applied Optics 52(29): 7152–7164. https://doi.org/10.1364/AO.52.007152

Tulsiani, S., Zhou, T., Efros, A. A., and Malik, J. 2017. Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 209–217. https://doi.org/10.1109/CVPR.2017.30

Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., and Brox, T. 2017. DeMoN: Depth and motion network for learning monocular stereo. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 5622–5631. https://doi.org/10.1109/CVPR.2017.596

Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., and Tumblin, J. 2007. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. Proceedings of the ACM SIGGRAPH Conference on Computer Graphics. https://doi.org/10.1145/1275808.1276463

Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., and Yuille, A. 2015. Towards

unified depth and semantic prediction from a single image. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2800–2809. https://doi.org/10.1109/CVPR.2015.7298897

Wedel, A., Franke, U., Klappstein, J., Brox, T., and Cremers, D. 2006. Realtime depth estimation and obstacle detection from monocular video. Lecture Notes in Computer Science, 475–484. https://doi.org/10.1007/11861898_48

Anonymous.1994.What Is MATLAB-MATLAB and Simulink. https://ww2.mathworks.cn/en/discovery/what-is-matlab.html

Witkin, A. P. 1981. Recovering surface shape and orientation from texture. Artificial Intelligence, 17:(1–3).https://doi.org/10.1016/0004-3702(81)90019-9

Xie, J., Girshick, R., and Farhadi, A. 2016. Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks. Lecture Notes in Computer Science, 9908 LNCS, 842–857. https://doi.org/10.1007/978-3-319-46493-0_51

u, D., Ricci, E., Ouyang, W., Wang, X., and Sebe, N. 2019. Monocular Depth Estimation Using Multi-Scale Continuous CRFs as Sequential Deep Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(6): 1426–1440. https://doi.org/10.1109/TPAMI.2018.2839602

Xu, D., Wang, W., Tang, H., Liu, H., Sebe, N., and Ricci, E. 2018. Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 3917–3925. https://doi.org/10.1109/CVPR.2018.00412

Yang, J., Research, A., Price, B., Cohen, S., Lee, H., and Yang, M. H. 2016. Object Contour Detection with a Fully Convolutional Encoder-Decoder Network. https://openaccess.thecvf.com/content_cvpr_2016/papers/Yang_Object_Contour_Detection_CVPR_2016_paper.pdf

Yatsenko, M., and Zh, V. 2020. How to Implement Artificial Intelligence for Solving Image Processing Tasks | Apriorit. https://www.apriorit.com/dev-blog/599-

ai-for-image-processing

Ye, M., Wang, X., Yang, R., Ren, L., and Pollefeys, M. 2011. Accurate 3D pose estimation from a single depth image. Proceedings of the IEEE International Conference on Computer Vision, 731–738. https://doi.org/10.1109/ICCV.2011.6126310

Yin, Z., and Shi, J. 2018. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1983–1992. https://doi.org/10.1109/CVPR.2018.00212

Zeiler, M. D., Taylor, G. W., and Fergus, R. 2011. Adaptive deconvolutional networks for mid and high level feature learning. Proceedings of the IEEE International Conference on Computer Vision, 2018–2025. https://doi.org/10.1109/ICCV.2011.6126474

Zhang, C., Wang, L., and Yang, R. 2010. Semantic Segmentation of Urban Scenes Using Dense Depth Maps, 6314 LNCS(PART 4): 708–721. https://doi.org/10.1007/978-3-642-15561-1_51

Zhou, H., Ummenhofer, B., and Brox, T. 2018. DeepTAM: Deep Tracking and Mapping. Lecture Notes in Computer Science, 11220 LNCS: 851–868. https://doi.org/10.1007/978-3-030-01270-0_50

Zhu, X., Cohen, S., Schiller, S., and Milanfar, P. 2013. Estimating spatially varying defocus blur from a single image. IEEE Transactions on Image Processing, 22(12): 4879–4891. https://doi.org/10.1109/TIP.2013.2279316

Zhuo, S., and Sim, T. 2011. Defocus map estimation from a single image. Pattern Recognition, 44(9): 1852–1858. https://doi.org/10.1016/J.PATCOG.2011.03.009

Zoran CSAIL, D., Isola CSAIL, P., Krishnan Google, D., and Freeman Google, W. T. 2015. Learning Ordinal Relationships for Mid-Level Vision. https://openaccess.thecvf.com/content_iccv_2015/papers/Zoran_Learning_Ordinal_Relationships_ICCV_2015_paper.pdf

Zwald, L., and Lambert-Lacroix, S. 2012. The BerHu penalty and the grouped effect. https://arxiv.org/abs/1207.6868v1

AUTOBIOGRAPHY

WASAN THAKER NADHIM, Iraq. She attended her elementary and secondary school, respectively, at Al Fayhaa Elementary School and Al Mutamayizat Secondary School. She completed her bachelor's degree in Computer Science from Mosul University in 2009, Iraq. Immediately after graduation, she worked as a lecturer at the Northern Technical University, Iraq. In 2019, she started her master's degree in Computer Engineering at Cukurova University, Turkey, after completing the special student stage. Her research interests include Machine Learning, Artificial Intelligence, Deep Learning, Image vision.