

VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Implementing GeoNet++ and Analyzing and Testing

An Industrial Internship Report

Submitted By:

Gaurav Navada
(20BKT0128)

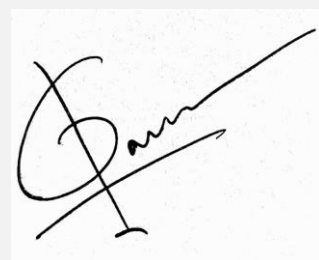
in partial fulfilment for the award of the degree of

Bachelor Of Technology
in
Computer Science and Engineering

School Of Computer Science and Engineering
October 2023

DECLARATION BY THE CANDIDATE

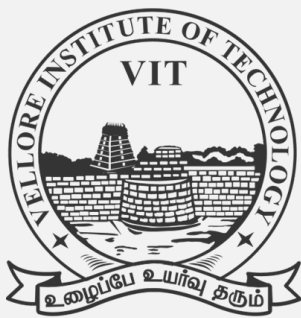
I hereby declare that the Industrial Internship report entitled “**Implementing GeoNet++ and Analysing and Testing**” submitted by me to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide industrial training undertaken by me under the supervision of **Muraleedhara Navada, MapmyIndia**. I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

A handwritten signature in black ink, appearing to read 'Gaurav', with a large, stylized initial 'G' and a long horizontal stroke extending to the right.

Signature of The Student

Name: Gaurav Navada

Reg. Number: 20BKT0128



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

BONAFIDE CERTIFICATE

This is to certify that the Industrial Internship report entitled “**Implementing GeoNet++ and Analysing and Testing**” submitted by **Gaurav Navada (20BKT0128)** to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in Computer and Science** is a record of bonafide Industrial Internship undertaken by him/her under my supervision. The training fulfils the requirements as per the regulations of this Institute and, in my opinion, meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

SUPERVISOR

Date:

Date:

Internal Examiner (s)

External Examiner (s)

Internship Certificate

MapmyIndia

Digitally mapping India since 1995

Date: 25th July, 2023

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Gaurav Navada**, a student of **B.Tech, (Computer Science), Vellore Institute of Technology, Vellore** has successfully completed his internship with **C.E. Info Systems Ltd.** from **27th June 2022 to 15th July, 2022** and from **8th May 2023 to 02nd June, 2023**.

He has worked on **"Implementing Geo Net ++ & Analysing & Testing."**

He was found punctual, hardworking and inquisitive during the period of his internship program.

We wish **Gaurav Navada**, good luck for all future endeavors in his life and career.

For C.E. Info Systems Ltd.


(Authorized Signatory) *



C.E. INFO SYSTEMS LIMITED

(Formerly known as CE Info Systems Pvt. Ltd.)

237, Okhla Industrial Estate, Phase III, New Delhi - 110020 | Phone: +91-11-4600 9900 | Fax: +91-11-4600 9920
E-mail: contact@mapmyindia.com | Website: www.mapmyindia.com | CIN No. L74899DL1995PLC065551

ACKNOWLEDGEMENT

I would like to express my special thanks to VIT University for empowering us with the knowledge to have been able to intern at a reputable organisation. I would like to express gratitude to my supervising professors at VIT University. I also wish to express my sincere gratitude to Mr Muraleedhara Navada and Mr Vinay Kumar Verma for providing me with an opportunity to do my internship.

I sincerely thank my parents, friends and seniors for their guidance and encouragement in carrying out this internship. I also wish to express my gratitude to the officials and other staff members of MapmyIndia, who rendered their help during the period of my internship.

I also thank the chancellor of VIT University, Vellore, for providing me with this opportunity to embark on this project.

I perceive this opportunity as a significant milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement to attain the desired career objectives.

Place : Vellore

(Gaurav Navada)

Date : 20/10/2023

Table of Contents

<u>LIST OF FIGURES</u>	8
<u>LIST OF SYMBOLS AND ABBREVIATIONS</u>	8
<u>SYNOPSIS OF THE REPORT</u>	9
<u>ABOUT THE COMPANY</u>	11
<u>SKILL SET INCULCATED BY THE CURRICULUM</u>	13
<u>KNOWLEDGE ACQUIRED BY THE INTERNSHIP</u>	14
INTRODUCTION	14
DATA - PREPROCESSING	14
PYTHON PROGRAMMING	14
COMPUTER VISION TOOLS	15
UNDERSTANDING HIGH-DEFINITION MAPS	15
POINT CLOUD GENERATION	16
COLMAP – SFM AND MVS TOOL	16
TRAINING AND TESTING ALGORITHMS	18
TENSORFLOW API	18
CLOUD COMPUTING (AWS)	18
DEEP LEARNING KNOWLEDGE	18
WORKING WITH GPUS AND USING CUDA TOOLKIT	19
<u>APPLICATION OF GAINED KNOWLEDGE</u>	20
PROJECT SUMMARY	20
OVERVIEW OF THE GEO.NET++ ARCHITECTURE	21
DEPTH TO NORMAL MODULE	22
RESIDUAL SUB-MODULE	22
NORMAL-TO-DEPTH MODULE	23
DEPTH AND NORMAL ENSEMBLE MODULE	23
EDGE-AWARE REFINEMENT MODULE	24
ITERATIVE INFERENCE	25
END-TO-END NETWORK TRAINING	25
OVERALL STRUCTURE OF GEO.NET++ NETWORK	27
BACK-BONE NETWORK AND STATE OF THE ART APPROACH	28
NYUD-V2 DATASET	28
GEO.NET++ IMPLEMENTATION DETAILS	29
2-D METRICS FOR QUANTITATIVE ASSESSMENT	29
TESTING GEO.NET++ WITH AUTHORS PROVIDED WEIGHTS	30
DEPTH MAP VISUALIZATION	31
SURFACE-NORMAL MAP VISUALIZATION	32

QUANTITATIVE ANALYSIS OF DEPTH AND NORMAL PREDICTIONS	33
REASON FOR WORSE PERFORMANCE	34
TRAINING THE GEONET++	35
DEPTH MAP VISUALIZATION WITH TRAINED WEIGHTS	36
SURFACE NORMAL MAP VISUALIZATION WITH TRAINED WEIGHTS	37
QUANTITATIVE ANALYSIS OF DEPTH AND NORMAL PREDICTIONS USING TRAINED WEIGHTS	38
CONCLUSION DRAWN FROM GEONET++ PERFORMANCE	40
COLMAP AS GEONET++ BASELINE MODEL	42
DEPTH MAP VISUALIZATION USING COLMAP AS BASELINE NET.	43
NORMAL MAP VISUALIZATION USING COLMAP AS BASELINE NET.	44
CONCLUSION	46
LANGUAGES AND FRAMEWORKS USED	47
 <u>COMPARISON OF COMPETENCY LEVELS</u>	 <u>49</u>
 <u>APPENDIX</u>	 <u>50</u>

List of Figures

Figure 1: MapmyIndia Logo

Figure 2: High-Definition Maps

Figure 3: Point Cloud generation

Figure 4: COLMAP implementation

Figure 5: GeoNet++ overview

Figure 6: Implementation of each GeoNet++ module

Figure 7: Overall architecture of Geonet++

Figure 9: Visualizations of different maps

Figure 10: After training of weights, visualization of different maps

Figure 11: COLMAP as baseline network visualization maps

List of Symbols and Abbreviations

HD Maps: High-Definition Maps

SfM: Structure for Motion

MVS: Multi View Stereo

Synopsis of The Report

High-definition (HD) maps are detailed digital representations of real-world environments used in applications like autonomous driving, augmented reality, and urban planning. They offer a precise depiction of road networks, landmarks, traffic signs, and other relevant features. Point clouds are collections of 3D points representing the spatial coordinates of objects or surfaces in an environment. They play a crucial role in generating HD maps by providing accurate depictions of the physical world, enabling the extraction of key features. Depth and surface normal information are vital in creating point clouds and HD maps. Depth information provides distances from the camera, facilitating 3D reconstruction, while surface normal indicate surface orientation, aiding in object shape estimation.

GeoNet++ is a geometric neural network that predicts both depth and surface normal maps from a single image. It leverages two-stream CNNs and specialized modules for depth-to-normal and normal-to-depth conversions. The "depth-to-normal" module enhances surface normal from depth using a least square solution, and the "normal-to-depth" module refines the depth map based on surface normal through kernel regression. An edge-aware refinement module further exploits boundary information. GeoNet++ excels in predicting depth and surface normal with strong 3D consistency and sharp boundaries, resulting in high-quality 3D scenes.

MapmyIndia recognizes GeoNet++'s potential in refining the high-definition map generation process. COLMAP, another technology in experimentation, employs an Image-based 3D approach involving Structure-from-Motion (SfM) for sparse scene representation and camera poses. This outputs feeds into Multi View Stereo (MVS) for dense scene reconstruction. MVS, within COLMAP, computes depth and/or normal information for each pixel based on the SfM output. By fusing depth and normal maps from multiple images, a dense point cloud of the scene is generated. Therefore, the quality of depth and normal maps directly impacts the resulting point cloud representation.

This report aims to comprehensively outline the implementation and functionality of GeoNet++. It assesses the network's performance through visual comparisons of Ground truth, Initial, and Refined depth and normal prediction maps. Additionally, it conducts qualitative evaluations of Initial and Refined depth and normal predictions against their corresponding

ground truth. The report also delves into operational aspects of the network and discusses both its enhancements and limitations. Moreover, it endeavors to utilize the predicted depth and normal maps from COLMAP as initial inputs for the baseline network. Subsequently, it assesses GeoNet++'s refinement capabilities, providing valuable insights into the strengths and limitations of these refined maps.

In summary, HD maps offer detailed digital representations of real-world environments, crucial in various applications. Point clouds, consisting of 3D points, are instrumental in generating these maps. Depth and surface normal information are key factors in this process. GeoNet++ is a powerful neural network that predicts depth and surface normal, enhancing 3D scene reconstruction. MapmyIndia acknowledges the potential of GeoNet++ in refining their map generation process. COLMAP, another technology in use, employs an Image-based 3D approach through SfM and MVS for scene reconstruction. This report provides a thorough evaluation of GeoNet++'s performance and operational aspects, shedding light on its strengths and limitations. Additionally, it explores the integration of predicted maps from COLMAP into the baseline network, offering insights into the refinement capabilities of GeoNet++.

About The Company

MapmyIndia is a prominent Indian technology company specializing in digital map data, GPS navigation, tracking, location-based services, and GIS solutions. Established in 1995 by Rakesh and Rashmi Verma, the company initially focused on developing web mapping technology and products to optimize marketing and logistics operations in existing organizations. Over time, it has evolved into the most comprehensive and accurate provider of digital map datasets covering the entirety of India.

The groundbreaking RealView service by MapmyIndia captures, analyzes, and renders the world in 3D and 360-degree photorealistic clarity, positioning it as the preferred choice for Navigation, Telematics, ADAS, GIS, and Smart City applications. The company also offers cutting-edge GPS tracking devices, in-dash car infotainment systems, and plug & play onboard diagnostics car trackers. Additionally, MapmyIndia provides a Map-based service known as Mappls, offering detailed maps for nearly 200 countries. This service assists users in finding and navigating to their destinations with step-by-step voice-guided directions, complemented by live traffic updates. The app excels in mapping terrains with detailed road networks, including advanced information like multiple names, road classifications, one-ways, turn restrictions, dividers, flyovers, tolls, ramps, and more.

MapmyIndia's online maps seamlessly integrate with ISRO Satellite Imagery, providing users with detailed satellite and hybrid views. Their offline navigation app, Navimaps, leverages offline vector data to deliver 3D terrains, city models, and 3D building renderings for in-car infotainment systems. The company's services extend beyond professional users and businesses, encompassing hyper-local mobile and web apps for consumers, including maps.MapmyIndia.com, renowned as India's first and most comprehensive, accurate, and detailed map.

In 2020, MapmyIndia launched a COVID-19 dashboard, furnishing real-time updates on the pandemic's spread across India. The company also achieved recognition by winning the Government of India's Atma Nirbhar Bharat App Innovation Challenge for their consumer app, Move. With headquarters in New Delhi and regional offices in Mumbai and Bengaluru, MapmyIndia maintains a network of smaller offices spread across India. Overall,

MapmyIndia's innovative solutions and services have solidified their position as a premier choice for location-based services and GIS AI technologies in India.



Figure 1: MapmyIndia Logo

Skill Set Inculcated by The Curriculum

While I was enrolled in the VIT program, I was allowed to gain experience in various subject areas, all of which came in very handy during my internship. The following are some of the most important aspects that I found to be especially helpful:

1. Beginning-level C, C++, Python, and Java classes sparked my interest in fundamental programming and problem-solving skills.
2. The Data Structures and Algorithms course assisted me in figuring out which data structures would be most beneficial for the projects I worked on during my internship and in developing algorithms for those structures.
3. The introductory classes I took on Computer Operating Systems, Computer Architecture and Organisation, and Networks and Communications, expanded my fundamental computer knowledge and better prepared me for the responsibilities I would be expected to fulfil during my internship.
4. The Internet and Web Programming class helped me better grasp of Database management techniques and other fundamentals of web technology. Understanding Database implementations helped me immensely in analyzing and storing datasets, as well as acquiring new datasets for my projects.
5. Programming with Data Science course helped me understand the fundamentals of data processing and application of ML models to analyze, predict or classify various datasets, along with understanding the fundamentals of data manipulation and implementation of data science algorithms in our day-to-day life
6. Machine Learning and Artificial Course enabled me to experience the benefits and shortcomings of various Machine Learning and Artificial Intelligence Algorithms and how to implement them using python, which helped me immensely during application of processes during my internships.
7. Deep Learning Course offered by VIT, helped me understand the basics of Neural network technology along with the ability to apply these various deep learning algorithms, which was useful for understanding the complex architectures used in my internship.
8. Image Processing course helped me understand concepts like convolution and basic edge detection in images which was used in my project during the internship.

Knowledge Acquired by the Internship

Introduction

As delineated in the introductory section, my internship responsibilities encompassed the training and evaluation of the GeoNet++ neural network architecture. This undertaking entailed the proficient application of diverse deep learning and image processing algorithms. Furthermore, I conducted comprehensive assessments of the aforementioned architecture in conjunction with MapmyIndia's exploration of HD maps, specifically within the framework of COLMAP. Following rigorous testing and analytical endeavors across various algorithms to fulfill this mandate, I accrued valuable insights and skills pivotal to my ongoing pursuits in the realms of deep learning and data science.

Data - Preprocessing

My internship involved pre-processing various image datasets, with the help of python libraries such as Numpy and Pandas, to enable me to clean the dataset. This involved removing any “NA” or NULL values present in the datasets, correcting the proportion ratio of all the images, as well as changing the shape of the image dataset.

Another process involved was to normalize the pixel intensities of various images before conducting any operation on it (or feeding it to the neural network). This was done using normalization libraries in numpy and pandas from python, where they have implemented algorithms like Min-max normalizations, Z-score normalizations and various other Standardization tools

Python Programming

My internship required me to use various python algorithms along with the libraries like numpy and pandas for data-preprocessing, Scipy for data access and manipulation, Tensorflow API for implementation of neural networks, and other such libraries to carry out data science workflows.

Using Python for my internship helped me understand how complex tasks can be simplified, as well as how ideas and projects can be implemented in the overall work space and deployed to be a service for the normal user. It also helped me understand the basics of algorithm

implementation in python, where I used various data structures like heap, queue, binary trees, etc.

Computer Vision Tools

During my internship, I had the privilege to use various tools from the OpenCV library in python, which is used for computer vision tasks along with image processing. I used the tools from this library to conduct and verify the images along with rectifying their dimensions and structure, and to predict depth and surface normal and understand the implementation and accuracies compared with the industry standards.

I have used the canny edge extractor tool from the OpenCV library for the implementation of the edge-aware refinement module present in the GeoNet++ framework.

Understanding High-Definition Maps

High-Definition Maps (HD Maps) are a type of digital map that offers a detailed and accurate representation of the physical world, encompassing roads, buildings, and various other features. These maps are specially crafted for utilization by autonomous vehicles and advanced driver assistance systems (ADAS) to ensure dependable navigation and heightened safety. HD maps play a pivotal role in furnishing precise and reliable navigation for autonomous vehicles, contributing to enhanced safety through the provision of real-time updates on road conditions and potential hazards. Moreover, they work towards improving traffic flow by optimizing routing strategies.



Figure2: High-Definition maps Example

Point Cloud Generation

A point cloud is a collection of data points in a three-dimensional coordinate system, where each point represents a specific position in space. These points are typically obtained through various sensing technologies such as LiDAR (Light Detection and Ranging), photogrammetry, or depth sensors. Each point contains several measurements, including its coordinates along the X, Y, and Z-axes, and sometimes additional data such as a colour value, which is stored in RGB format, and luminance value, which determines how bright the point is.

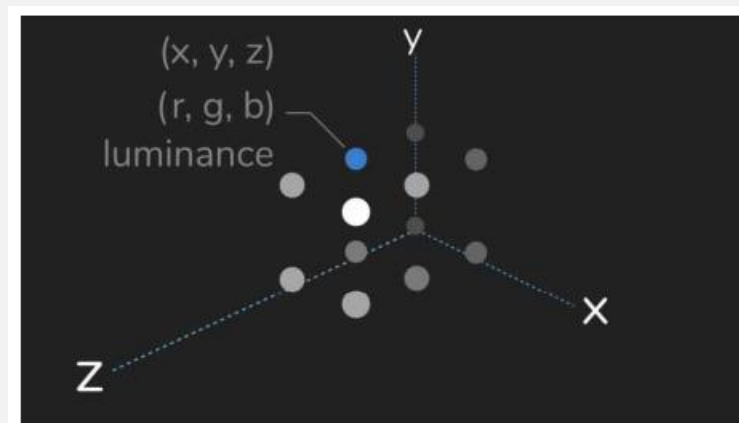


Figure 3: Point Cloud Example

Point clouds are important because they provide a detailed representation of the geometry and spatial information of a real-world scene or object. They are extensively used in fields such as computer vision, robotics, virtual reality, augmented reality, geographic information systems (GIS), and autonomous driving.

COLMAP – SfM and MVS tool

COLMAP is a versatile software package that encompasses both a graphical user interface and a command-line interface for Structure-from-Motion (SfM) and Multi-View Stereo (MVS) tasks. It provides an extensive set of functionalities for reconstructing images, whether they are ordered or unordered.



Figure 4: Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

The conventional approach to image-based 3D reconstruction involves initially generating a sparse representation of the scene along with the camera poses of the input images through Structure-from-Motion. Subsequently, this output serves as the foundation for Multi-View Stereo, aiming to recover a dense representation of the scene.

Structure-from-Motion (SfM) is the process of reconstructing three-dimensional structure from its projections onto a series of images. It operates on a set of overlapping images of the same object captured from distinct viewpoints. The result is a comprehensive three-dimensional reconstruction of the object, accompanied by the reconstructed intrinsic and extrinsic camera parameters for all images.

Multi-View Stereo (MVS) builds upon the output of SfM to compute depth and/or normal information for each pixel in an image. By merging the depth and normal maps derived from multiple images in a three-dimensional framework, a dense point cloud of the scene is generated. Subsequent algorithms, like (screened) Poisson surface reconstruction, leverage the depth and normal information from the fused point cloud to reconstruct the three-dimensional surface geometry of the scene.

Training and Testing Algorithms

In my internship, I was involved with training and testing the GeoNet++ framework to remark on the final results obtained and whether it will be useful for MapmyIndia's further experimentations in HD map generations. This involved splitting the dataset into training and testing datasets, training the architecture with custom loss functions, and testing the architecture by comparing the accuracies of the predicted values with the baseline model and Ground-Truth Dataset.

TensorFlow API

The Tensorflow API was used for building the neural network architecture for the GeoNet++. This API provides an extensive range of neural networks as implementations using python. Tensorflow allows us to implement various neural networks algorithms from dense neural layers to convolution layer, along with the implementation of optimizers (like Adam and SGD) and activation functions (Relu, softmax and sigmoid).

Cloud Computing (AWS)

Training and Testing of GeoNet++ network was done in an AWS virtual machine which had custom Nvidia Graphics card. The reason for this was that GPU within the local system along with tools like Google Colab, do not provide sufficient RAM or Cuda cores to train our model. Therefore, for the internship, I was involved with accessing an AWS machine remotely, setting up an environment and training my network with their Cuda cores.

Through this experience I had understood how to set up a remote machine on AWS, how to install python dependencies onto the machine, how to enable training analysis on the machine, and how to access various components during the testing phase.

Deep Learning Knowledge

During the course of this internship, I had the invaluable opportunity to delve into the core principles of a diverse range of deep learning algorithms. This experience provided me with a profound insight into the practical deployment of these algorithms within a production environment. Additionally, I gained substantial proficiency in pivotal concepts, such as

activation functions like RELU and Softmax, as well as optimization techniques including Adam and SGD.

Moreover, I acquired a comprehensive understanding of the application of convolutional layers in processing image data, allowing for a deeper grasp of its intricate features and functionalities. This newfound knowledge and hands-on experience have significantly bolstered my capabilities in the domain of deep learning and machine learning.

Working with GPUs and using CUDA Toolkit

During the implementation of GeoNet++, I acquired substantial expertise in harnessing the computational power of Graphics Processing Units (GPUs) and effectively leveraging the CUDA Toolkit. This technology combination proved instrumental in expediting computations and optimizing performance throughout the training and evaluation processes of the GeoNet++ neural network architecture.

By harnessing the parallel processing capabilities of GPUs, I observed a remarkable acceleration in model training and inference times. This not only expedited the experimentation phase but also allowed for more comprehensive analyses within a reasonable timeframe. The CUDA Toolkit, designed to maximize the potential of NVIDIA GPUs, provided a robust set of tools and libraries that seamlessly integrated with the GeoNet++ implementation. This synergy facilitated smooth execution and accelerated computations, further enhancing the efficiency of the neural network.

The utilization of GPUs and the CUDA Toolkit not only streamlined the training process but also enriched the depth of experimentation. This, in turn, contributed to a more refined and optimized implementation of GeoNet++, showcasing the significant impact of GPU-accelerated computing in the realm of deep learning and computer vision applications.

Application of Gained Knowledge

Project Summary

The GeoNet++ model is a geometric neural network equipped with edge-aware refinement capabilities, allowing it to predict both depth and surface normal information from a single image. This is achieved through specialized modules, "depth-to-normal" and "normal-to-depth", which enhance the quality of predictions. The former utilizes a least square solution to estimate surface normals from depth, while the latter refines the depth map based on surface normal constraints through kernel regression. Additionally, boundary information is factored in via an edge-aware refinement module.

GeoNet++ demonstrates impressive capabilities in producing depth and surface normal predictions with robust 3D consistency and sharp boundaries, resulting in superior 3D scene reconstruction. This is particularly valuable for generating point clouds and creating high-definition maps. Moreover, as a versatile model, GeoNet++ can be integrated with other depth and surface normal estimation models to further enhance prediction quality and application efficiency.

The estimation of depth and surface normals is pivotal in generating precise 3D point clouds, a process integral to MapmyIndia's creation of highly detailed HD maps. The adoption of GeoNet++ is expected to refine depth and surface normal predictions, potentially leading to the development of even finer HD maps compared to their current model.

This report aims to delve into the methodologies employed by the authors to leverage the geometric relationship between depth and surface normal, with the goal of improving estimation quality. It also seeks to assess the validity of the authors' claims by conducting tests and visualizing the refined depth and surface normal predictions.

Overview of the GeoNet++ Architecture

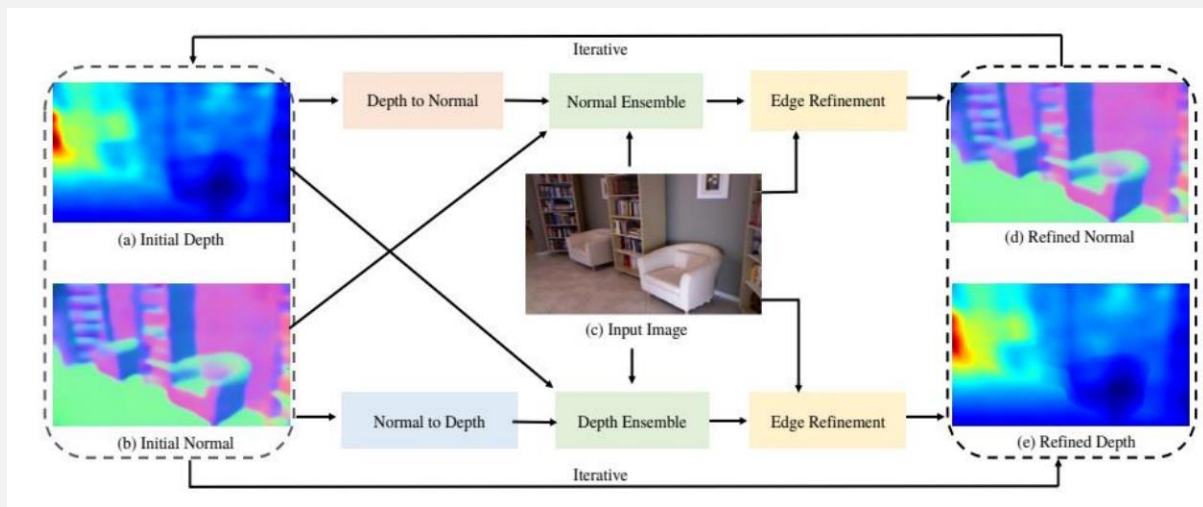


Figure 5: Overview

The overall architecture of GeoNet++ is illustrated in the above figure. It has a two-stream backbone CNN, which predicts initial depth and surface normals from a single image respectively. Based on the initial depth map predicted, we apply the depth-to-normal module to transfer the initial depth map to the normal map. This module refines the surface normals with the initial depth map considering geometric constraints. Similarly, given the initial surface normal estimation, we generate the depth using the normal-depth module. This enhances the depth prediction by incorporating the inherent geometric constraints to the estimation of depth from normals. Our “depth-to-normal” module relies on least-square and residual submodules, while the “normal-to-depth” module updates the depth estimates via kernel regression.

The depth/normal maps generated with the above components are then adjusted via the depth (normal) ensemble module. Furthermore, guided by the learned propagation weights, our “edge-aware refinement module” sharpens boundary predictions and smooths out noisy estimations. Finally, GeoNet++ can be applied iteratively by taking the refined results from previous iteration as inputs. Our framework enforces the final depth and surface normal prediction to follow the underlying 3D constraints, which directly improves 3D surface reconstruction quality.

Depth to Normal Module

This module proposes a depth to normal transformation that explicitly incorporates depth-normal consistency into deep neural networks. This module consists of two sub-modules - **the Least square module**, viewed as a fix-weighted neural network, and the **Residual sub-module**, that aims at smoothing and combining results with the initial surface normal.

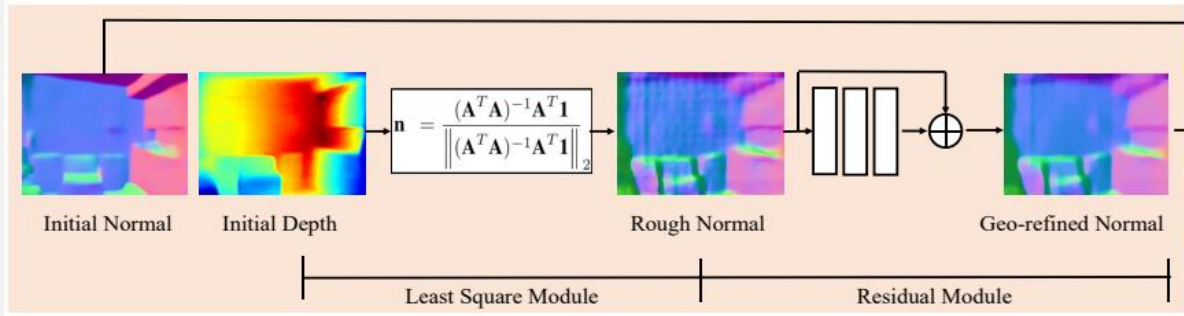
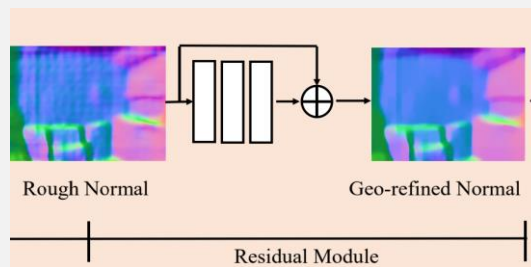


Figure: Initial depth and Surface normal getting passed into the Least Square model and Residual module to produce a Geo-refined Surface Normal estimation.

By explicitly leveraging the geometric relationship between depth and surface normals, GeoNet++ network circumvents the difficulty in learning geometrically consistent depth and surface normals. The Authors also believe that the module can be incorporated and jointly fine-tuned with other networks that predict depth maps from raw images.

Residual Sub-Module

The Least-Square module, mentioned in the previous step, occasionally produces noisy surface normal estimation (Rough Normal) due to issues like noise and improper neighbourhood size. To further improve the quality, we propose a residual module, which consists of a **3-layer CNN with skip-connections** as shown in the below figure. The goal is to smooth the noisy estimation from the least square module.



Normal-to-Depth Module

The main goal of the normal-to-depth module is to refine the depth of any pixel \mathbf{i} , given its surface normal $(\mathbf{n}_{ix}, \mathbf{n}_{iy}, \mathbf{n}_{iz})$ and an initial estimate of depth \mathbf{z}_i .

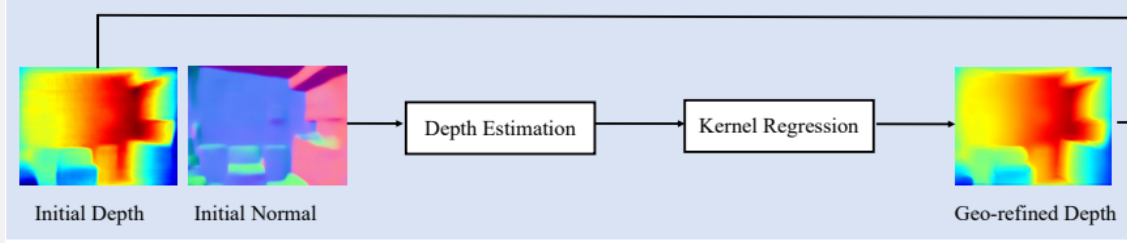


Figure: Initial depth and normal go through the depth estimation with respect to its neighbours and next the Kernel regression module

The idea of this module is to find neighbours of the current point, with the same surface normal. These neighbours might give an idea of what the general depth of the current point is. Using Kernel regression, we aggregate the properties of the neighbour point along with the current point, to find the optimal depth of this point. Using the formulas of the depth-to-normal module, we can elaborate on the specific point and compare the initial depth with the predicted depth to further enhance the estimation.

Depth and Normal Ensemble Module

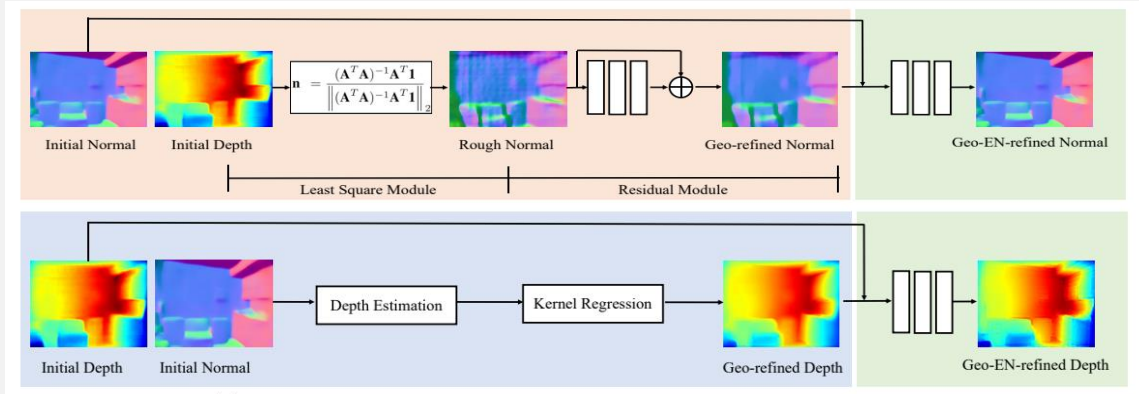


Figure: Image (on top) refers to the depth-to-normal module with the ensemble module producing Geo-EN-refined Normal. Image (on bottom) - refers to the normal-to-depth module with ensemble module producing Geo-EN-refined Depth.

To further enhance the prediction quality, the Authors of GeoNet++ proposed to combine the “Initial Depth/ Normal” from the backbone network and the “Geo-refined Depth/ Normal” from the geometric refinement to an ensemble module illustrated in the above figure for both depth and normal.

Both depth and normal ensemble modules have similar architectures. The depth ensemble module takes as inputs “Initial Depth” from the backbone network and “Geo-refined Depth” from the geometric module and produces a **refined depth** – “**GeoEN-refined Depth**”.

To enlarge the receptive field of the ensemble module, the input is firstly processed with 3 convolution layers with a dilation rate of 2, kernel size 3×3 , and channel number 128. This is followed by another 2 dilation-free convolution layers with kernel size 3×3 and channel number 128.

Edge-aware refinement Module

The Authors believe that their above two modules - “depth-to-normal” and “normal-to-depth” can refine the depth and surface normal prediction from a given image, but it still might contain noise and inconsistencies. To solve this problem, the Authors have designed an edge-aware refinement module (shown in the below figure) to further enhance the prediction. This module **enhances the boundary prediction and removes noisy predictions** by gradually aggregating the information from neighbouring pixels. This process is guided by a set of learned weight maps (known as “Weight Maps”). The edge-aware refinement contains two submodules, i.e., **the weight map predictor** and **the recursive propagator**. This module uses the same architecture for both the depth and normal prediction so that we only elaborate on the details for the depth.

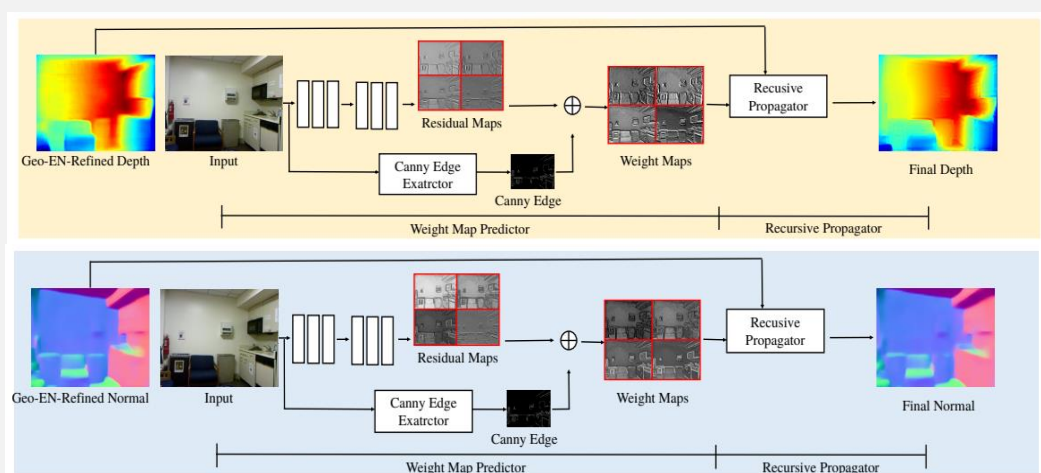
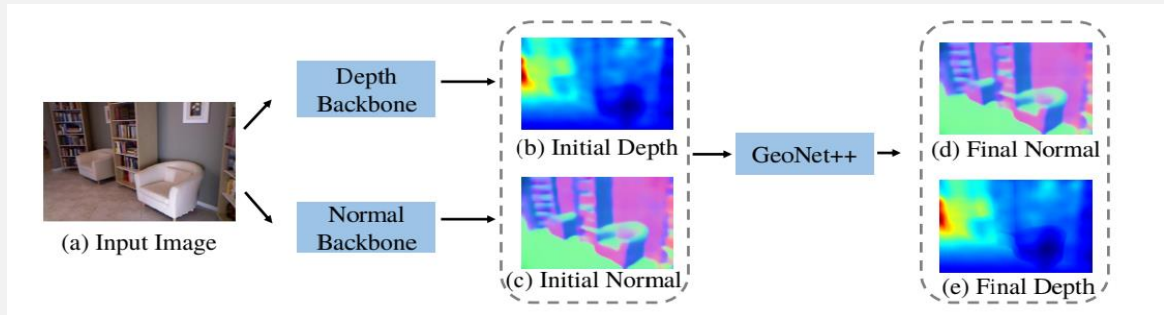


Figure: Edge-aware refinement module for both depth (top figure) and surface normal (bottom figure). Residual (weight) maps include “left to right”, “right to left”, “top to bottom”, “bottom to top”.

Iterative inference

The Authors of GeoNet++ have made the network so that it can be applied iteratively to further improve the results as shown in Fig. . The refined depth and normal maps from previous iterations can further serve as the inputs to GeoNet++ for iterative refinement. Note that the Authors have only applied this iteratively during the inference. In the training phase, the Authors have applied GeoNet++ only once to reduce the memory consumption and improve the training efficiency.

End-to-End Network Training



GeoNet++'s full system can be illustrated from the above figure. The backbone network produces the initial depth and surface normal maps, which are further refined with GeoNet++ by incorporating the geometric constraint and the edge information. The whole system can be trained end-to-end. The Authors have defined a loss function for training the full system.

They suggest denoting the ground-truth depth of pixel \mathbf{i} as

$$z_i, \hat{z}_i \text{ and } z_i^{\text{gt}}$$

respectively. Similarly, we denote the initial, refined, and ground-truth surface normals as

$$\mathbf{n}_i, \hat{\mathbf{n}}_i, \text{ and } \mathbf{n}_i^{\text{gt}}$$

respectively. The overall loss function is the summation of two losses, one for the depth and one for the normals,

$$L = l_{\text{depth}} + l_{\text{normal}}$$

The depth loss, \mathbf{l}_{depth} is expressed as

$$l_{\text{depth}} = \frac{1}{M} \left(\sum_i \|z_i - z_i^{\text{gt}}\|_2^2 + \eta \sum_i \|\hat{z}_i - z_i^{\text{gt}}\|_2^2 \right), \quad (9)$$

with M the total number of pixels.

The surface normal loss, \mathbf{l}_{normal} is,

$$l_{\text{normal}} = \frac{1}{M} \left(\sum_i \|\mathbf{n}_i - \mathbf{n}_i^{\text{gt}}\|_2^2 + \lambda \sum_i \|\hat{\mathbf{n}}_i - \mathbf{n}_i^{\text{gt}}\|_2^2 \right). \quad (10)$$

Here λ and η are hyperparameters which balance the contribution of individual terms.

Overall Structure of GeoNet++ Network

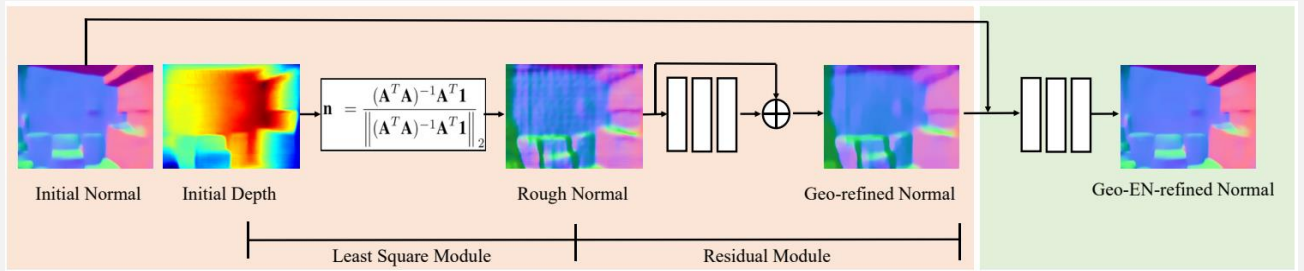


Figure: The depth-to-normal module (L) first estimates “Rough Normal” from the “Initial Depth” with least square fitting; normals are then refined by the residual module producing “Geo-refined Normal”; a normal ensemble network (R) is utilised to fuse the initial and Geo-refined normals generating “Geo-EN-refined normal”

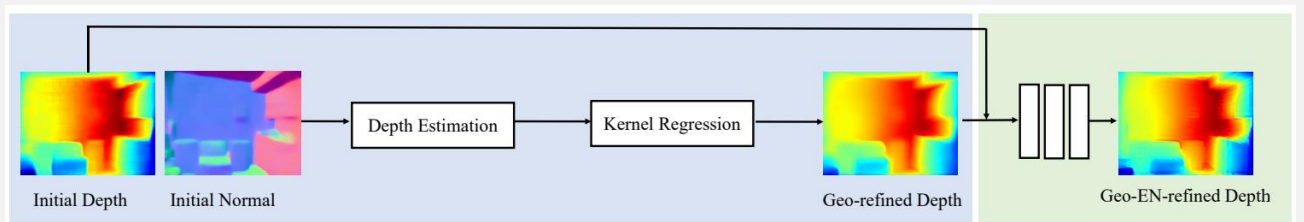


Figure: The normal-to-depth module (L) takes the “Initial Depth” and “Initial Normal” as inputs; the normal map helps propagate the initial depth prediction to neighbours; depth estimates are aggregated by the kernel regression module producing “Geo-refined Depth”. The depth ensemble module (R) taking “Geo-refined Depth” and “Initial Depth” as inputs further improves prediction generating “Geo-EN-refined Depth”.

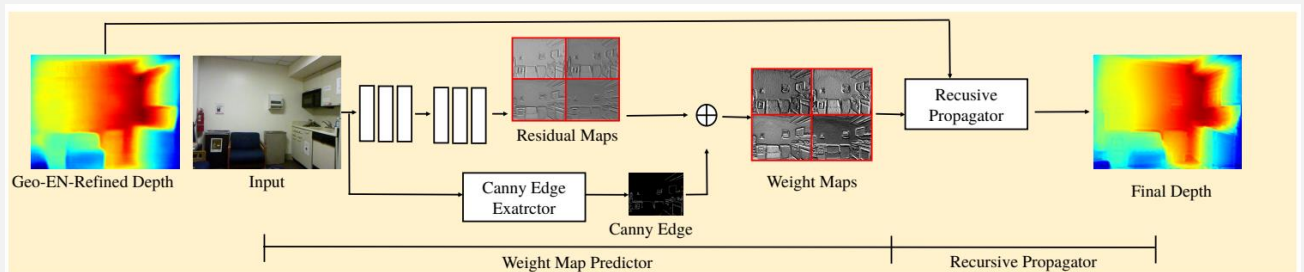


Figure: The edge-aware refinement module first constructs direction-aware propagation “Weight Maps” by combining low-level edges with “Residual Maps”; the recursive propagator utilises the learned weight maps to refine “Geo-EN-refined Depth” producing “Final Depth”.

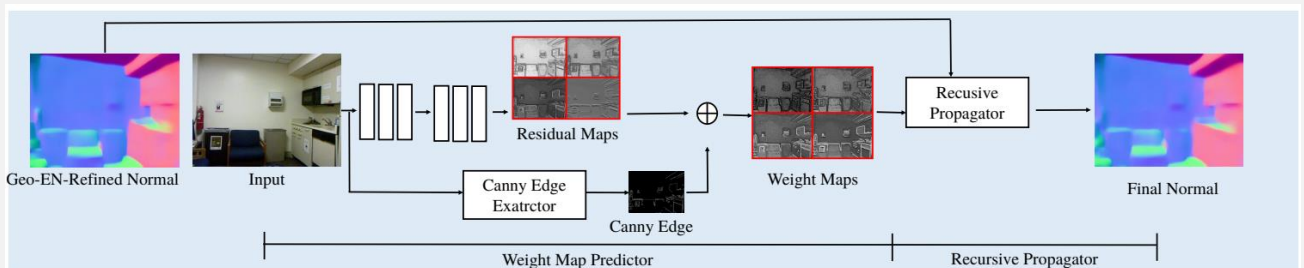


Figure: The edge-aware refinement module for surface normal.

Back-bone network and State of the art approach

For most of their experiments, they utilise a modified VGG-16, i.e., deeplab-LargeFOV with dilated convolution and global pooling, for initial depth and surface normal prediction. This is their baseline backbone network for comparison with VGG-based methods.

To further evaluate the effectiveness of the system, they have also adopted state-of-the-art methods to produce the initial prediction of depth and surface normal. We experiment with Multi-scale CNN V1 [24], Multi-scale CNN V2 [26], FCRN [33], Multi-scale CRF [27], and DORN [28] for initial depth estimation. For initial normal estimation, they have employed the initial normal map from SkipNet [30].

NYUD-V2 Dataset

This dataset contains 464 video sequences of indoor scenes, which are further divided into 249 sequences for training and 215 for testing. The Authors sample 30, 816 frames from the training video sequences as the training data.

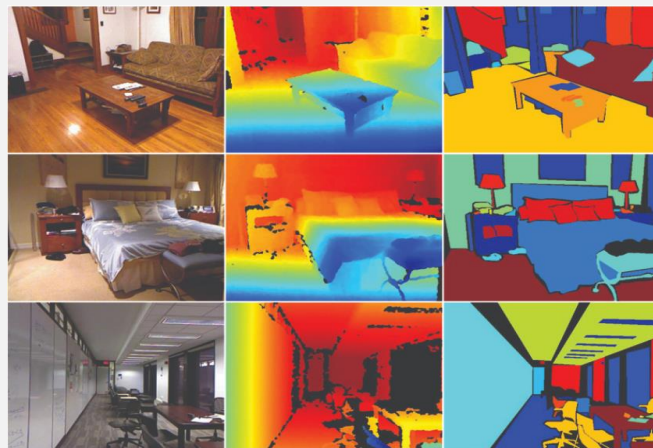


Figure: Some of the images with their depth and image segmentation maps from the NYUD-V2 Dataset

For the training set, they use an in-painting method to fill in invalid or missing pixels in the ground-truth depth map. The in-painting method is based on Colorization, which is a computer-assisted process of adding colour to a monochrome image or movie. Authors of this paper argue that colorization typically involves segmenting images into regions and tracking these regions across image sequences, but requires considerable user intervention and remains a tedious, time-consuming, and expensive task. The Authors propose a simple

colorization method based on the neighbouring pixels in space-time that have similar intensities should have similar colours, thus removing the limitations of previous mentioned methods.

The Authors then generate a ground-truth surface normal map following the procedure of [29], which uses a CNN architecture, which has been incorporated with several constraints (man-made, Manhattan world) and meaningful intermediate representations (room layout, edge labels) for the task of surface normal estimation.

GeoNet++ Implementation Details

- The GeoNet++ is implemented in TensorFlow v1.5 (converted to TensorFlow v.2.5).
- Their VGG baseline network is initialized with two-stream CNNs with networks pre-trained on ImageNet. Other baseline approaches are initialized with their corresponding pre-trained models, which are fixed in the procedure of fine-tuning GeoNet++.
- The Mean of the RGB values of the image is set to [104.008, 116.669, 122.675]
- Adam optimizer is being used.
- The norm of gradients is clipped, so that they are no larger than 5.
- The initial learning rate is $1e-4$. It is adjusted following the polynomial decay strategy with the power parameter 0.9.
- Random horizontal flip is utilized for augmentation.
- The whole system is trained with batch-size 1 for 40K iterations (where each iteration includes 10 steps) on the NYUD-V2.
- Hyperparameters $\{\alpha, \beta, \gamma, \lambda, \eta\}$ are set to $\{0.95, 9, 0.05, 0.01, 0.5\}$ according to validation on 5% randomly split training data.

2-D metrics for quantitative assessment

Following various existing approaches, the Authors adopt four metrics to evaluate the **resulting depth map** quantitatively. They are:

- root mean square error (RMSE)
- mean log 10 error (Log 10)
- mean relative error (REL),

- pixel accuracy as percentage of pixels with

$$\max(z_i/z_i^{gt}, z_i^{gt}/z_i) < \delta \text{ for } \delta \in [1.25, 1.25^2, 1.25^3]$$

The evaluation metrics for **surface normal prediction** are:

- mean of angle error (mean)
- median of angle error (median)
- root mean square error (RMSE)
- pixel accuracy as percentage of pixels with angle error below threshold t where:

$$t \in [11.25^\circ, 22.5^\circ, 30^\circ]$$

Testing GeoNet++ with Authors provided weights

The authors' empirical findings point towards a discernible enhancement in the accuracy of depth and surface normal predictions facilitated by the GeoNet++ model. To corroborate these assertions, a meticulous comparative analysis against ground truth data is imperative. Graciously, the authors have made available a comprehensive code implementation on GitHub, affording an invaluable resource for rigorous testing and validation procedures. Upon meticulous scrutiny by MapmyIndia, it has been ascertained that the code adeptly incorporates all discussed modules, including the crucial depth-to-normal, normal-to-depth, ensemble, and edge-aware refinement components. Notably, the code demonstrates a commendable adherence to the mathematical formulations and equations elucidated in the authors' papers, reinforcing its fidelity to the intended functionalities of GeoNet++. This comprehensive validation process serves as a testament to the model's robustness and reliability in generating refined depth and surface normal predictions, establishing it as a promising asset for advanced mapping applications.

Depth Map Visualization

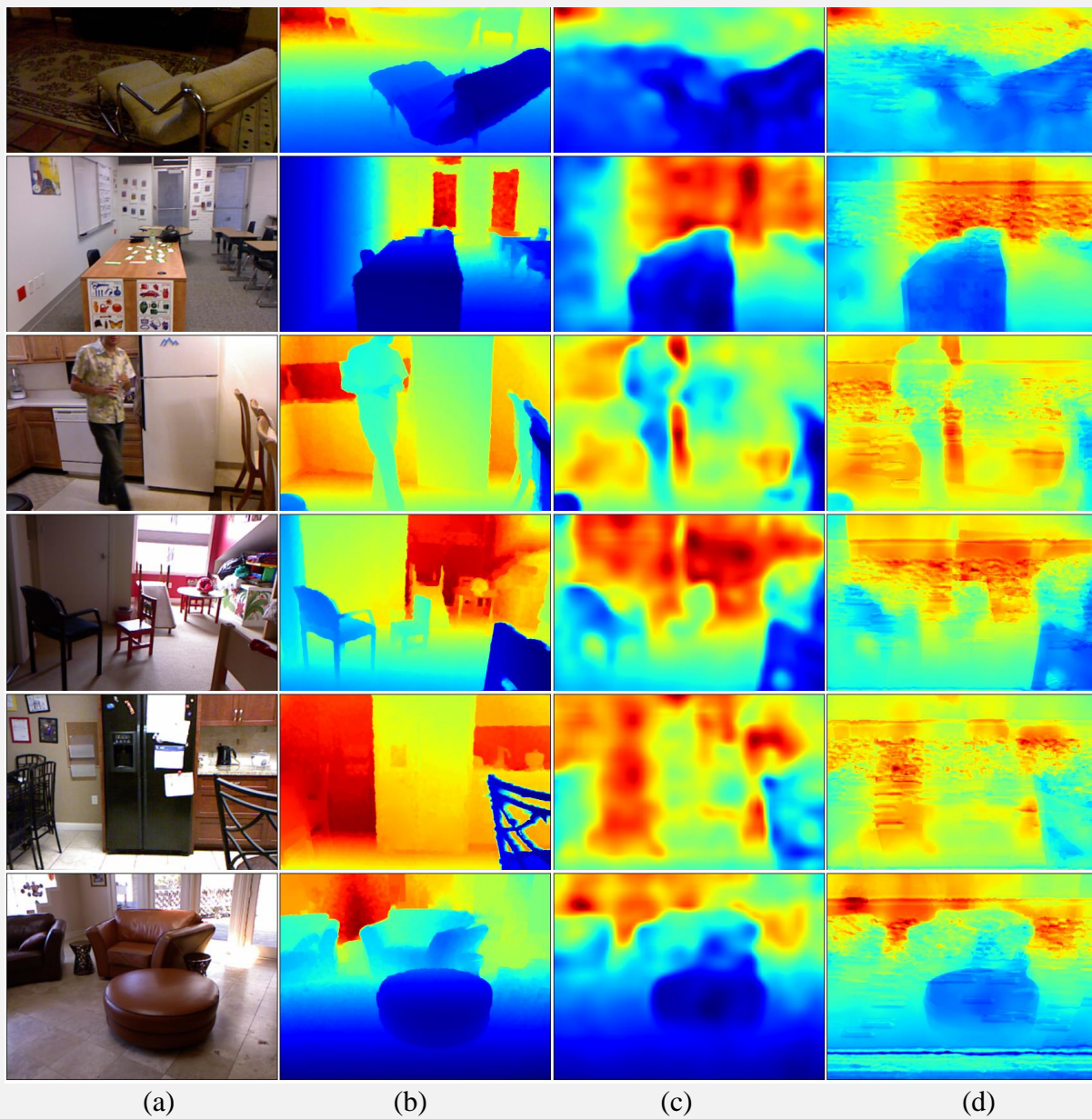


Figure: Images from left to right - (a) RGB original Image, (b) Ground Truth Depth map, (c) Initial Depth map prediction from baseline network, (d) Refined Depth Map estimation from GeoNet++ network.

Surface-normal Map Visualization

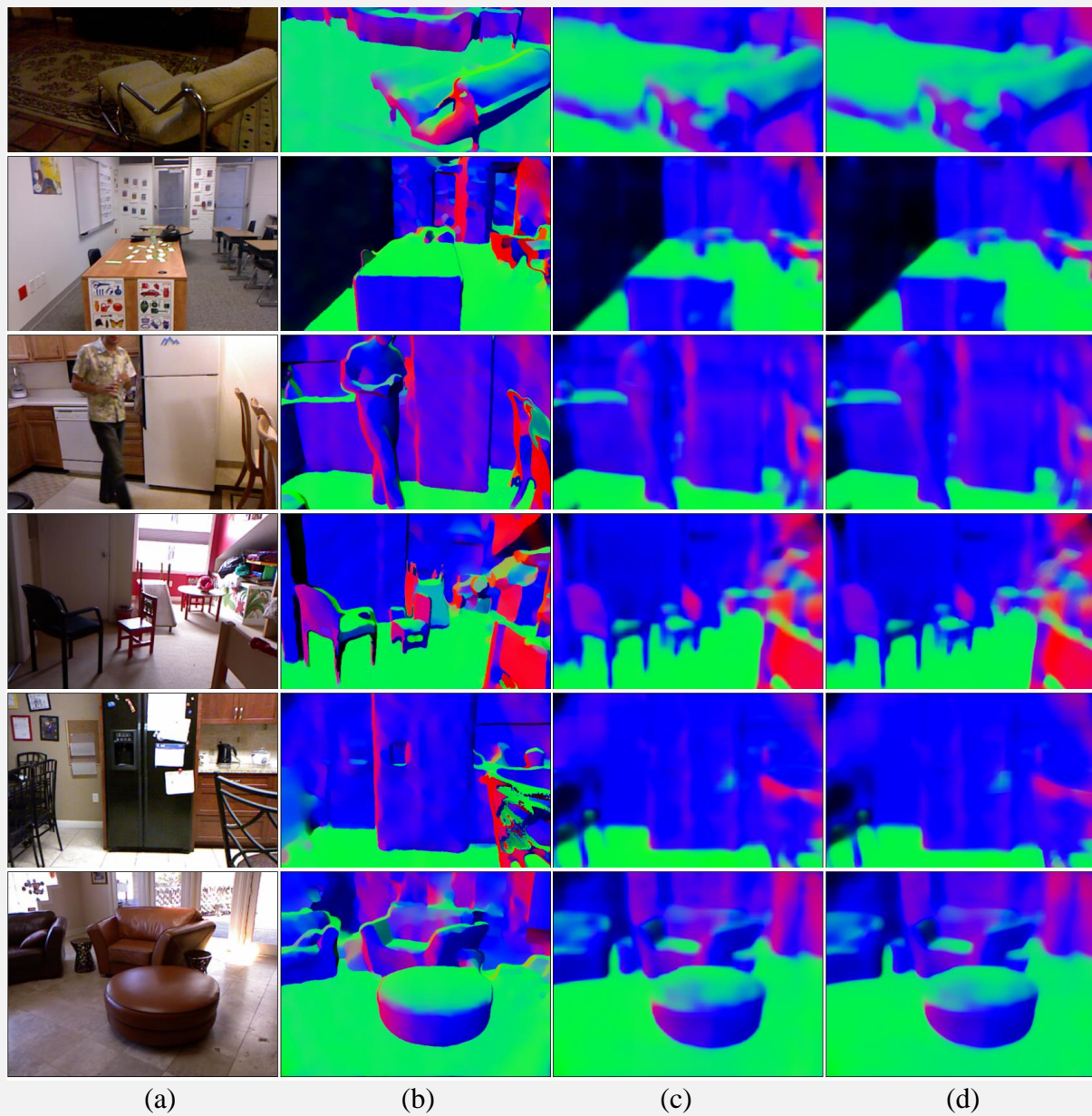


Figure: Images from left to right - (a) RGB original Image, (b) Ground Truth Surface Normal map, (c) Initial Surface Normal map prediction from baseline network, (d) Refined Surface Normal Map estimation from GeoNet++ network.

Quantitative analysis of Depth and Normal predictions

- Evaluation of Initial and Refined Depth Maps

Method	Error			Accuracy		
	rmse	log 10	rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline Net.	0.615	0.208	0.065	0.781	0.955	0.989
GeoNet++ Net.	0.664	0.255	0.079	0.712	0.923	0.975

Based on our previous discussions and visual comparisons, it is apparent that despite the GeoNet++ network's attempt to refine the initial depth map using the mentioned modules, it still suffers from numerous distortions that result in a visually inferior output compared to the initial depth map. The quantitative evaluation presented in the table further reinforces this observation, as the GeoNet++ network shows no improvement and performs worse than the original Baseline network in terms of the provided metrics. The low error and accuracy values can be attributed to these distortions, which cause deformations and inaccuracies in the depth pixels across various regions. As a result, these pixels exhibit smudged or malformed values that deviate from the intended predictions, ultimately leading to inaccurate depth estimations when compared to the ground truth. In conclusion, the quantitative evaluation strongly contradicts the claims made by the authors of GeoNet++, as it demonstrates that the network has failed to effectively refine the depth prediction and has produced inferior results compared to the baseline network.

- Evaluation of Initial and Refined Surface Normal maps

Method	Error			Accuracy		
	Mean	Median	rmse	11.25°	22.5°	30°
Baseline Net.	18.839	11.614	26.735	48.830	71.842	79.803
GeoNet++ Net.	19.538	12.748	27.004	45.159	70.545	79.313

Based on the visual comparisons, it is evident that there is no discernible difference between the initial and refined depth maps. Additionally, the qualitative evaluation metrics presented above clearly indicate that the refined normal map produced by the GeoNet++ network performs significantly worse than our baseline model. This discrepancy could be attributed to either the malfunctioning of the modules mentioned by the authors of GeoNet++ or the improper configuration of the weights within the GeoNet++ layers. These factors have likely contributed to the lack of improvement in the refined normal predictions, highlighting the need for further investigation and refinement of the GeoNet++ network.

Reason For worse performance

The subpar performance of the GeoNet++ network in comparison to the baseline model can be attributed to various factors. One plausible explanation is that the additional modules incorporated in GeoNet++ to refine the depth and surface normal predictions may not be effectively capturing the desired features or enhancing the accuracy of the results. Limitations in the design or implementation of these modules could impede their ability to improve the predictions significantly. This suggests that the claimed benefits of these modules by the authors may not hold true in practice.

Additionally, the presence of distortions and inaccuracies in the refined depth and surface normal maps suggests that there may be underlying issues with the network architecture or the quality of the training data. If the network is not able to effectively capture the complex relationships between depth, surface normals, and the corresponding images, it can result in degraded performance.

Contrary to the distortions observed in the depth map, the surface normal map does not exhibit such issues. But upon meticulous evaluation of the Refined Normal map in comparison to its corresponding Ground Truth Normal map, it becomes evident that the performance of GeoNet++ is notably inferior to that of the Baseline Network in terms of the normal map. In fact, as we have discussed earlier, the refined depth map utilises these modules to enhance the initial depth map and refine its edges. However, due to the presence of distortions, the depth pixel regions become smudged and deviate from their intended values. While one might argue that these distortions are caused by the modules themselves, it is noteworthy that the surface normal maps, which also utilise the same refinement modules, do not exhibit such distortions. Therefore, it is not plausible to attribute the network's underperformance to the modules proposed by the authors being ineffective.

Another possible factor contributing to the inferior performance of GeoNet++ could be associated with the training process and network configuration. The precise fine-tuning of weights and parameters is crucial to optimise the network's performance for the specific task at hand. Inadequate tuning of weights or a training process that lacks sufficient data or regularisation techniques can result in suboptimal outcomes. Therefore, it is reasonable to consider that the worse performance of GeoNet++ may be attributed to improper weight configurations employed by the authors. In order to draw conclusive findings, it is imperative

to conduct thorough training and evaluation of the model, taking into account the specific requirements and characteristics of the dataset. By addressing these aspects and appropriately optimising the weights, a more accurate assessment of GeoNet++ can be achieved.

Training the Geonet++

The GeoNet++ network was trained over a span of approximately 40,000 iterations, with each iteration consisting of 10 training steps. As a precautionary measure, the Authors took regular snapshots of the network's checkpoints every 1,000 iterations and stored them locally. This practice was implemented to ensure that in the event of any unforeseen issues causing the training process to halt abruptly, the latest saved checkpoint could be utilised to resume training seamlessly from where it left off. However, it has been noted that the most recent checkpoint provided by the Authors, denoted as "SRCNN.model-399999" in the "Testing the GeoNet++" section, may have potential faults or corruption. Considering this, our approach will involve training the system end-to-end and evaluating its performance based on the newly trained checkpoints, enabling a comprehensive assessment of the model's capabilities.

Depth Map Visualization with trained weights

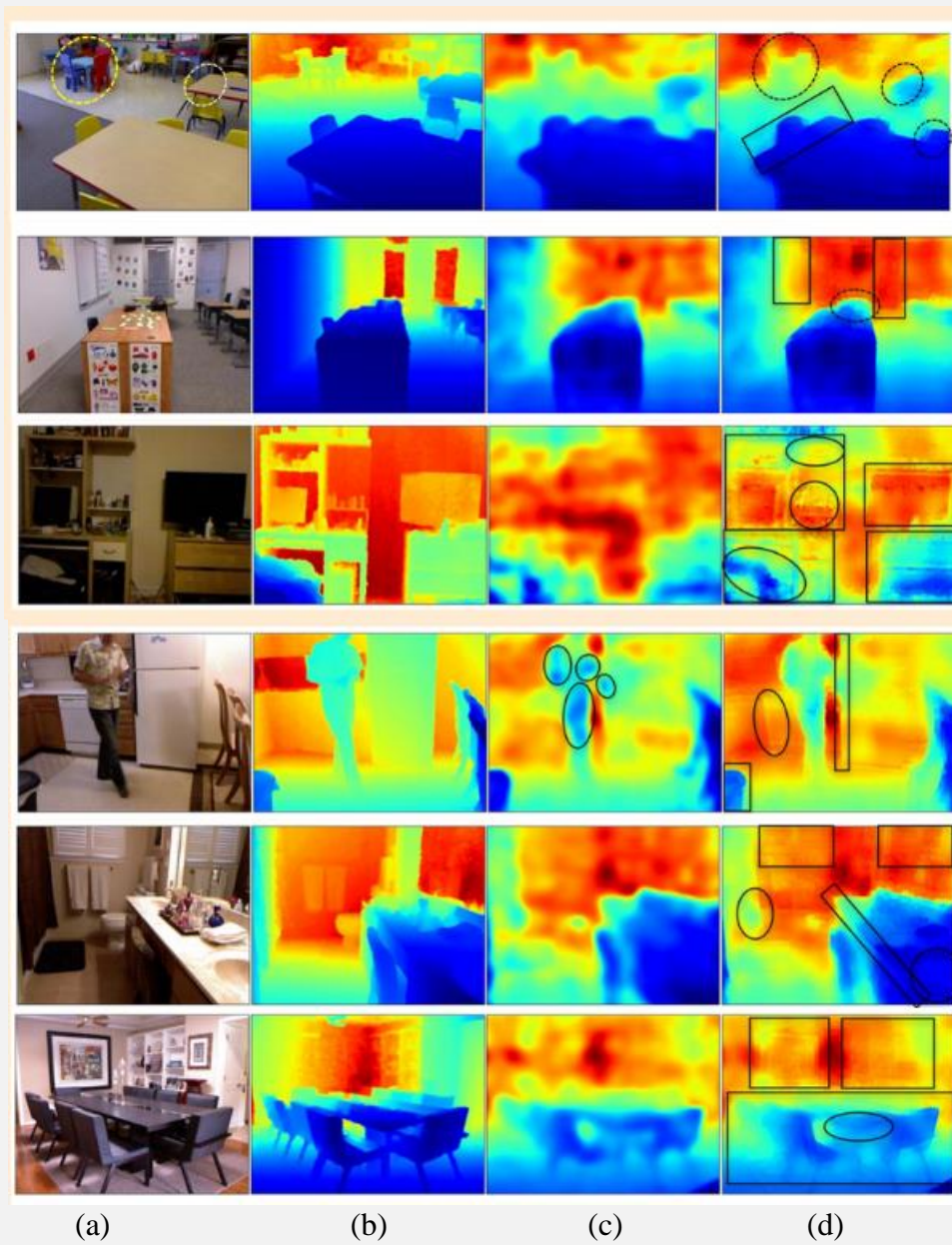


Figure: Images from left to right - (a) RGB original Image, (b) Ground Truth Depth map, (c) Initial Depth map prediction from baseline network, (d) Refined Depth Map estimation from GeoNet++ network.

Surface normal Map Visualization with trained weights

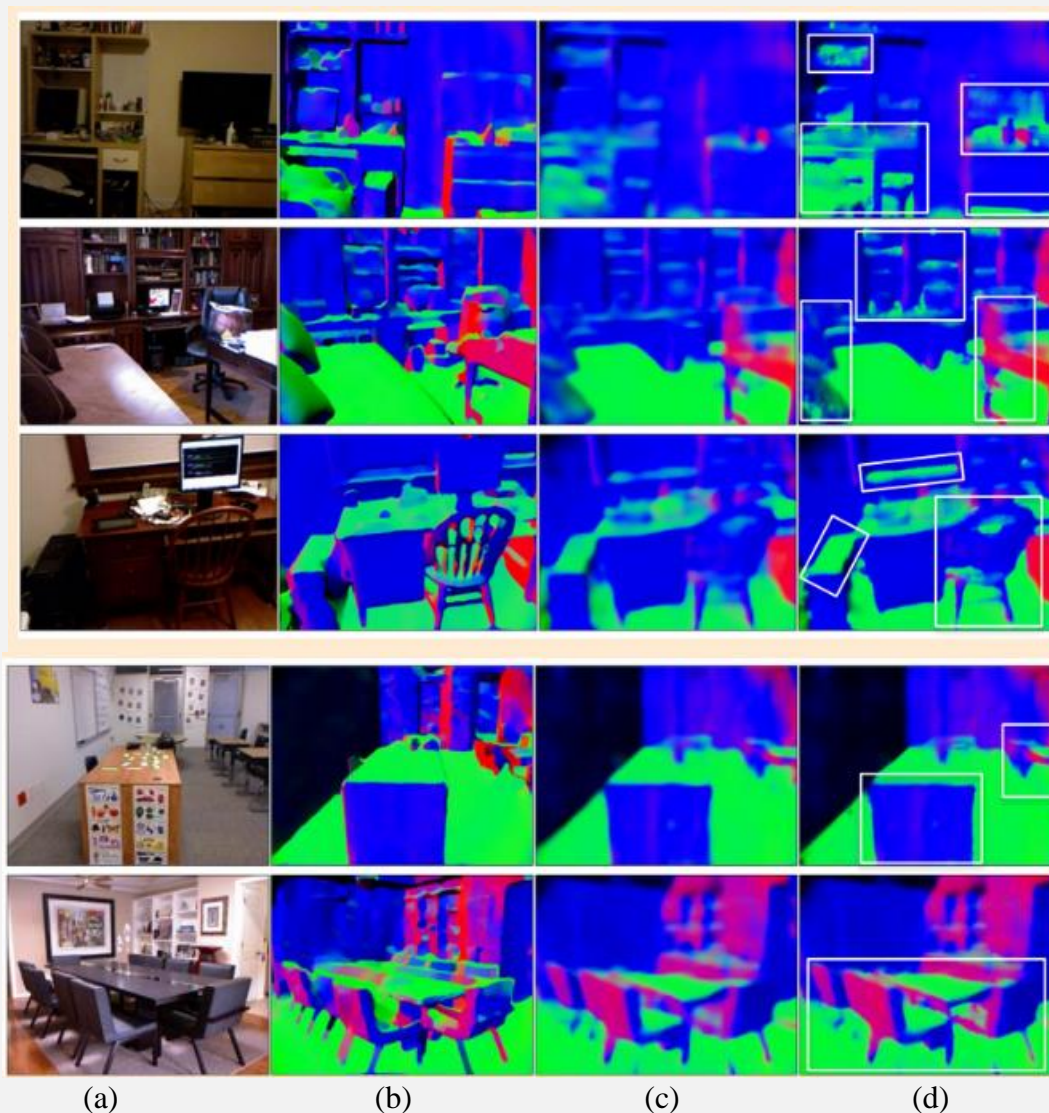


Figure: Images from left to right - (a) RGB original Image, (b) Ground Truth Normal map, (c) Initial Normal map prediction from baseline network, (d) Refined Normal Map estimation from GeoNet++ network.

Quantitative analysis of Depth and Normal predictions using trained weights

- Evaluation of Initial and Refined Depth Maps

Method	Error			Accuracy		
	rmse	log 10	rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline Net.	0.625	0.212	0.067	0.773	0.953	0.989
GeoNet++ Net.	0.595	0.209	0.065	0.784	0.954	0.988

Based on the qualitative evaluation metrics, the performance of the GeoNet++ network with our newly trained weights surpasses that of the baseline network. The Depth map generated by GeoNet++ demonstrates superiority over the baseline network in various aspects, except for the pixel accuracy as a percentage of pixels with $\delta < 1.25^3$, where the difference is minimal, possibly due to variations in the normalization of depth values across the maps. However, when considering the accuracy values in other sections and the error metric of the refined depth map, which outperforms the initial depth map, as well as the visual representation that shows significant improvement over the baseline network's Depth map, it is evident that the refined Depth map produced by GeoNet++ significantly enhances the quality of the initial depth map. In summary, the qualitative evaluation highlights the remarkable performance of the GeoNet++ network with our trained weights compared to the baseline network. The refined Depth map demonstrates superior results through improved accuracy and error metrics, along with a visually enhanced representation. These findings support the conclusion that the refinement process of GeoNet++ effectively enhances the quality of the initial depth map produced by the baseline network, resulting in superior depth estimation outcomes.

- Evaluation of Initial and Refined Surface Normal Maps

Method	Error			Accuracy		
	Mean	Median	rmse	11.25°	22.5°	30°
Baseline Net.	18.441	11.463	26.141	49.297	72.548	80.502
GeoNet++ Net.	18.726	11.150	27.144	50.334	72.894	80.326

The qualitative evaluation of the metrics reveals notable improvements in the performance of the refined Surface Normal Maps obtained through the newly trained weights compared to the Initial Surface Normal map. Particularly, the Refined Surface Normal map exhibits significant advancements over the Refined Surface Normal map obtained from the Authors' provided

weight. Analysing the error metrics, we observe that the Baseline network's Normal maps demonstrate lower mean error and root mean square error compared to the GeoNet++ network's Normal maps. However, the median error of the GeoNet++ network's Normal map is lower than that of the Baseline network. It is worth noting that the mean error is lower in the initial normal map, potentially attributed to variations in the normalization of surface normals within the maps. The visual representations of the refined normal map reveal instances where additional object normals are captured, which are not depicted in the Ground truth normal map. This suggests that the refinement process enhances the representation of surface normals, resulting in a more accurate depiction of the scene. Another possible explanation is that the presence of outliers or disturbances in the Initial depth map causes the average or mean to be lower than that of the Refined depth map. However, the median, which indicates the central tendency of the distribution compared to the Ground truth, is lower in the refined Normal map compared to the initial Normal map. This signifies that the refined depth map exhibits a smoother and more defined representation of surface normals with a uniform distribution, surpassing the quality of the initial normal map.

Furthermore, it is evident that the Initial Normal map exhibits a higher pixel accuracy, represented as the percentage of pixels with an angle error below 30 degrees, compared to the Refined Normal map. This discrepancy could arise due to the absence of normalization for the normal maps (Ground truth, Initial, and Refined) within a specific range, as mentioned in the preceding evaluation of depth maps. As discussed earlier, another plausible reason could be attributed to the instances where the Refined Normal map captures object normals that are not depicted in the Ground truth normal map, indicating GeoNet++'s endeavour to refine the deficiencies of the Initial Normal map and potentially resulting in divergent outcomes compared to the Ground truth map. Consequently, the evaluation of these regions based on the Ground truth may yield different outcomes compared to the evaluation performed by the Initial Normal map. Nevertheless, it is noteworthy that two out of three sections of the accuracy metrics exhibit significant improvements over the Initial Normal map. Considering the observations made during the visual representation analysis of the Refined Normal map, it can be reasonably inferred that the refined normal map indeed outperforms the Initial Normal map, albeit with subtle enhancements that may not be as pronounced as the improvements observed in the evaluation of the depth map.

Conclusion drawn from GeoNet++ performance

In conclusion, the qualitative assessment of depth and surface normal maps, utilizing the provided weights, initially yielded suboptimal results in terms of accuracy and error metrics. It was evident that further training was required to enhance model performance. Introducing newly trained weights led to significant improvements in the refined depth and surface normal maps.

The evaluation of depth maps underscored the superiority of the refined depth map generated by GeoNet++. It exhibited lower error metrics compared to the baseline network. Although pixel accuracy showed a slight advantage in the initial depth map, this difference may be attributed to variations in depth value normalization. The visual representation of the refined depth map further affirmed its ability to capture more accurate depth information, surpassing the baseline network's performance.

Similarly, the evaluation of surface normal maps demonstrated the effectiveness of the refined normal map obtained through GeoNet++. While mean error and root mean square error were slightly higher in the refined normal map compared to the initial normal map, the median error exhibited notable improvement. This indicated that the refined normal map achieved a more uniform distribution of surface normals, resulting in smoother and well-defined representations. Visual analysis corroborated these findings, showcasing improved geometric structures and enhanced clarity in object depiction.

Notably, discrepancies emerged between the refined normal maps and the Ground truth normal map. This suggested that GeoNet++ attempted to refine deficiencies in the initial normal map by producing different results. Despite this, two out of three sections of the accuracy metrics exhibited significant enhancements over the initial normal map, confirming the refined normal map's superiority.

Comparing the provided weights with the newly trained weights solidified the necessity for further training. The refined depth and surface normal maps derived from the newly trained weights demonstrated superior performance, presenting clearer and more accurate representations compared to the results obtained with the provided weights. These

improvements underscored the significance of training and fine-tuning the network to achieve optimal results.

Additionally, it's crucial to acknowledge that the refinement process of the GeoNet++ network heavily relies on the quality of the initial baseline network. A superior initial baseline network would provide a stronger foundation for the refinement process, allowing GeoNet++ to further enhance the accuracy and fidelity of the depth and surface normal maps. By employing a more advanced baseline network, the refinement process could capture finer details and nuances, resulting in even higher accuracy and visual quality in the refined maps.

In conclusion, while the newly trained weights have demonstrated significant improvements over the initial baseline network, further advancements can be achieved by enhancing the performance of the baseline network itself. Investing in the development of a superior baseline network would lay a solid groundwork for the refinement process, ultimately leading to even better results in terms of accuracy, error metrics, and visual fidelity of the depth and surface normal maps generated by GeoNet++.

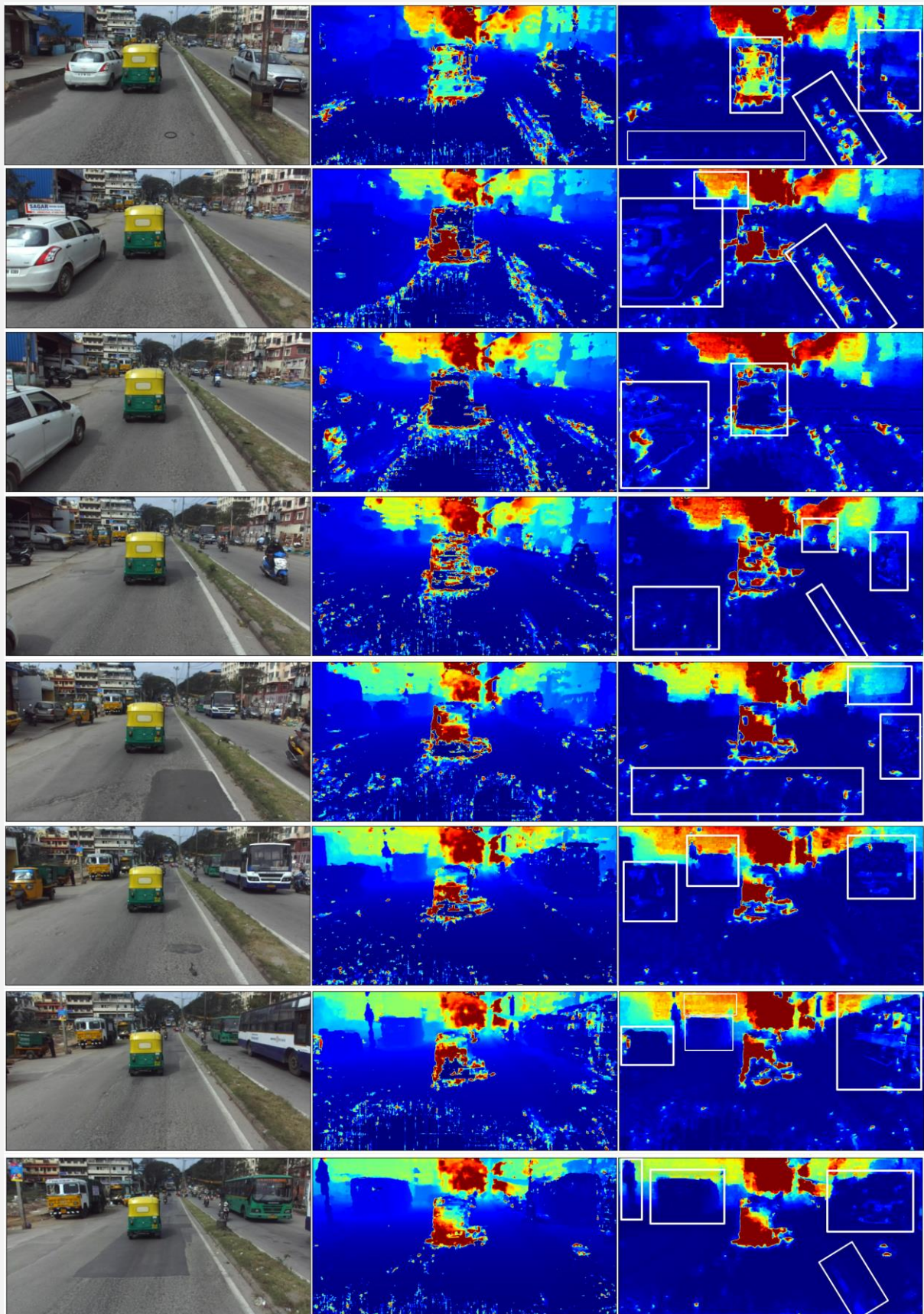
COLMAP as GeoNet++ baseline model

COLMAP is a robust framework designed to address the challenges of structure-from-motion and multi-view stereo tasks, aiming to generate high-quality point clouds from a collection of images. Leveraging advanced multi-view stereo algorithms, COLMAP effectively combines depth and normal maps obtained from multiple images to reconstruct a dense and accurate three-dimensional representation of the scene. The resulting point cloud serves as a foundation for the generation of intricate and precise high-definition (HD) maps. Consequently, it becomes apparent that enhancing the quality of the depth and normal maps used as references for these scenes would significantly improve the fidelity of the resultant point clouds and, consequently, the quality of the HD maps produced.

The authors of GeoNet++ have put forth a claim stating that their framework can enhance the initial depth and normal maps generated by a baseline network, resulting in a more refined map of the given image. However, our extensive testing of the GeoNet++ framework has revealed that the authors' claim does not align with our observations. Upon thorough analysis, we have determined that the issue may be attributed to the weights or checkpoints provided by the authors for our Refinement network in GeoNet++. To address this concern, we conducted training on the network using our own dataset, resulting in new trained weights and checkpoints. Subsequent testing utilizing these updated parameters has demonstrated significant improvements in the system's performance. The refined depth maps not only exhibit visually more appealing results but also yield superior qualitative evaluation outcomes compared to the initial depth and normal maps and the refined depth and normal maps obtained from the authors' provided weights, thus partially supporting the authors' claim.

Hence, based on their assertion, it is conceivable to substitute the existing baseline network with an alternative network for obtaining the initial depth and normal maps. By doing so, the GeoNet++ network should be capable of generating refined depth and normal maps for the provided images, leading to enhanced visual outcomes compared to the initial depth and normal maps acquired from COLMAP. Consequently, we intend to assess the efficacy of GeoNet++ by subjecting it to a testing phase involving 117 outdoor scene images depicting Bengaluru roads featuring two lanes, along with various vehicles, buildings, trees, and other elements. These images were captured by MapmyIndia. Following the image processing and refinement of initial depth and normal maps by GeoNet++, we will proceed to discuss our observations pertaining to both the initial and refined maps, culminating in our final conclusions.

Depth map visualization using COLMAP as Baseline net.



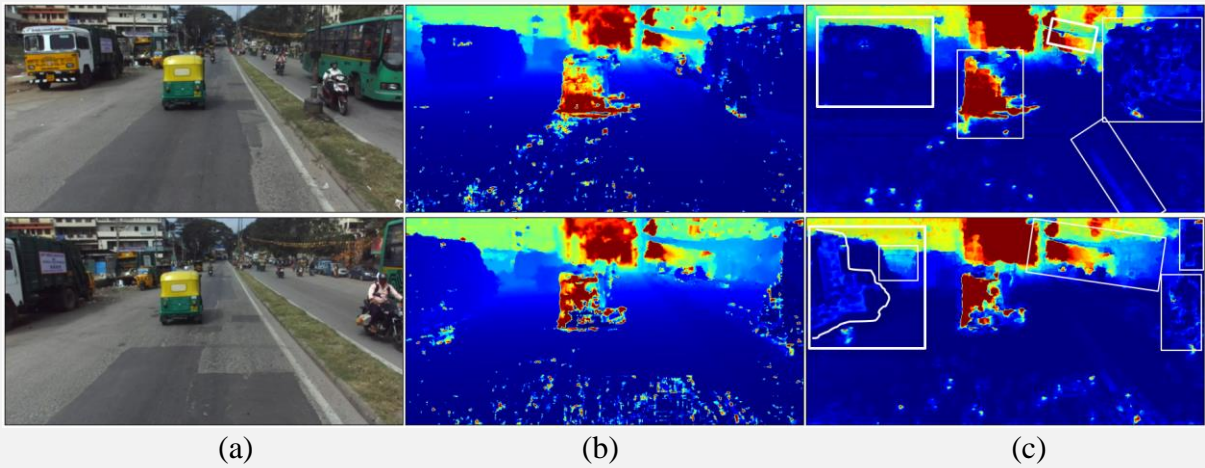
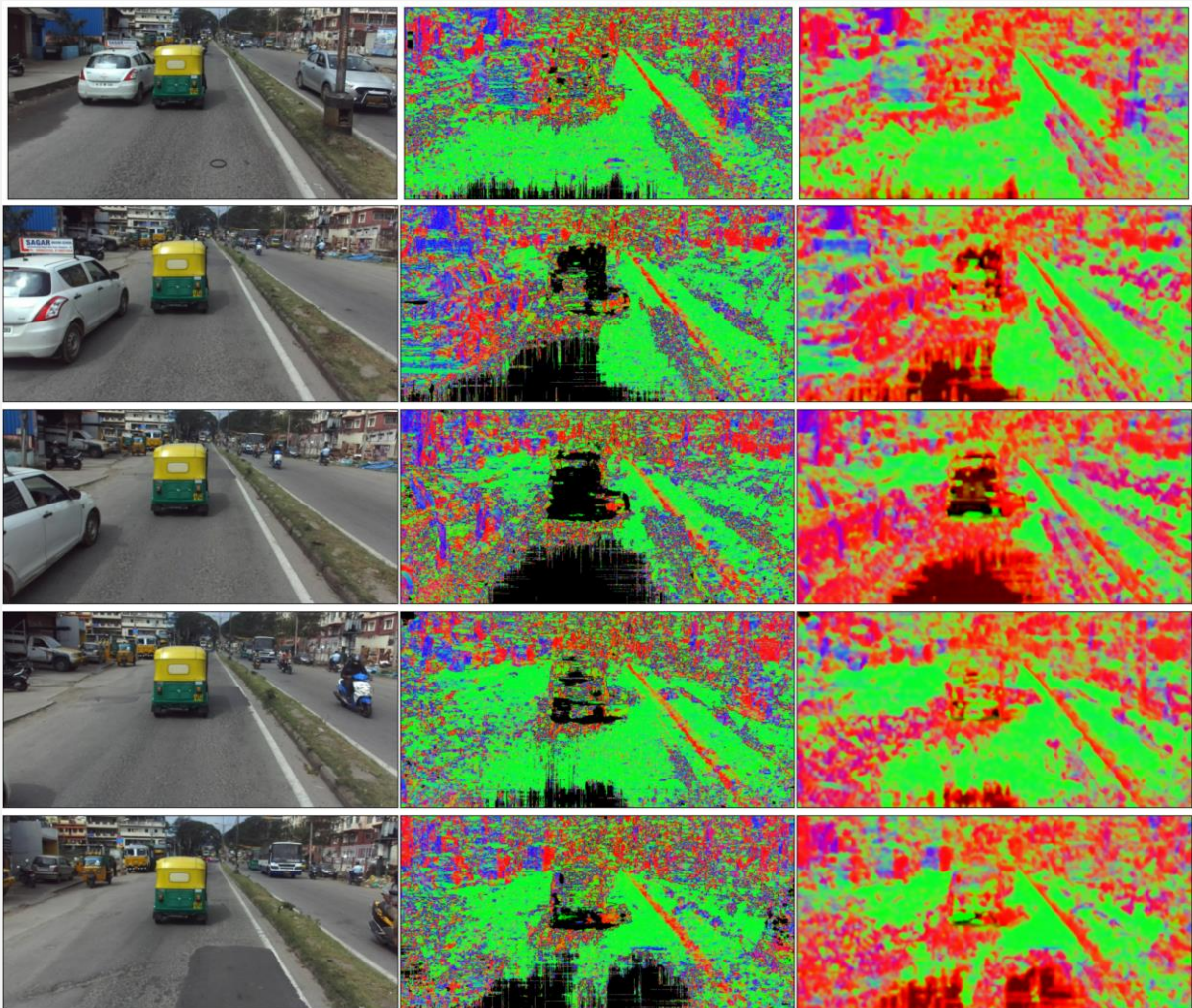


Figure: Images from left to right - (a) RGB original Image, (b) Initial Depth map prediction from COLMAP as out baseline network, (c) Refined Depth Map estimation from GeoNet++ network.

Normal map visualization using COLMAP as Baseline net.



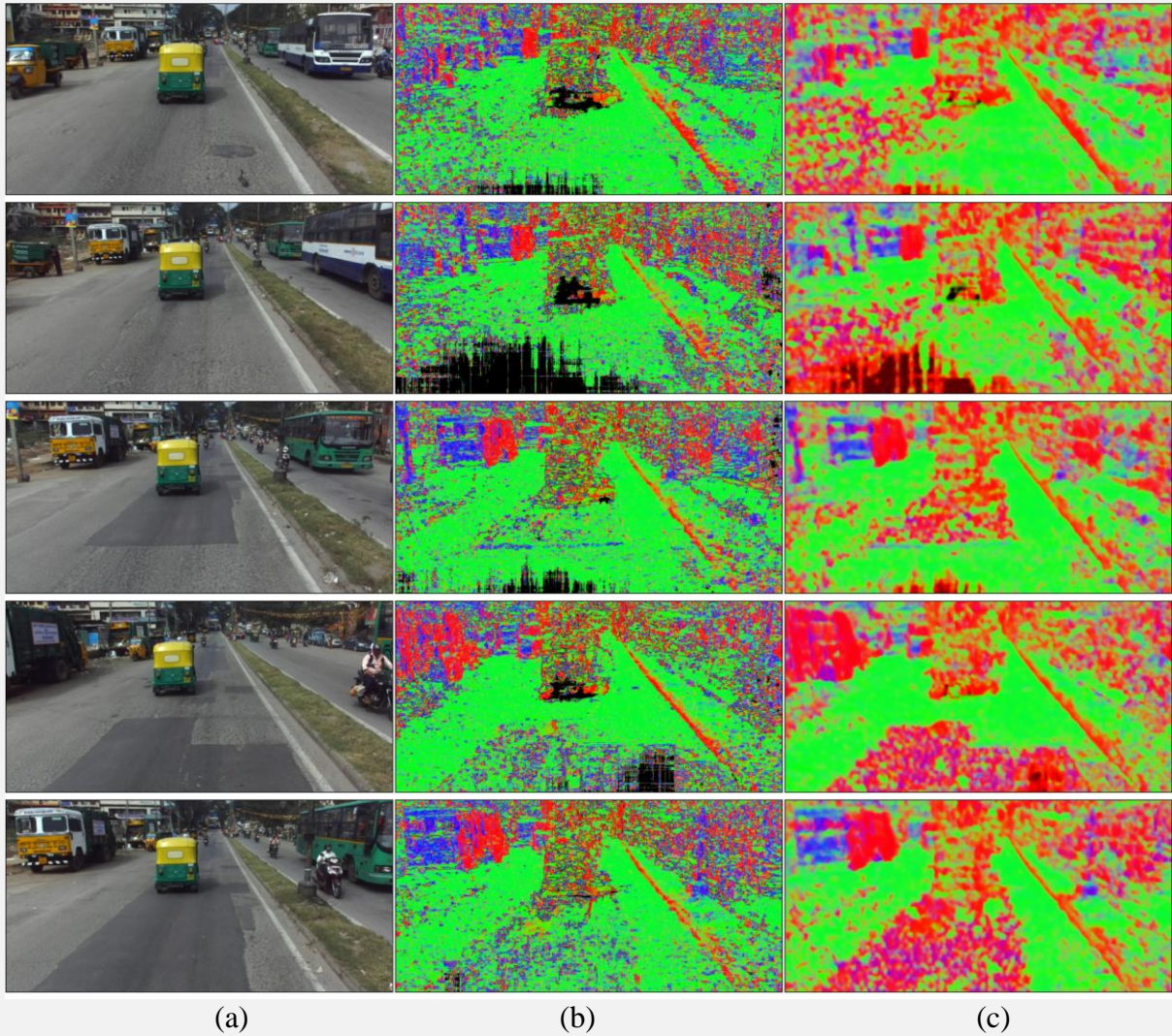


Figure: Images from left to right - (a) RGB original Image, (b) Initial Surface normal map prediction from COLMAP as out baseline network, (c) Refined Surface normal Map estimation from GeoNet++ network.

Conclusion

This report thoroughly analyzes GeoNet++, its underlying principles, and performance evaluation. It delves into the mathematical foundations and logical principles of GeoNet++, providing a deep understanding of its functionality. The training, testing procedures, and results on NYUv2 datasets were meticulously examined. The study concludes that GeoNet++ refinement enhances existing frameworks, leading to accurate point cloud representations.

However, our evaluation of the provided GeoNet++ implementation showed limited improvement in depth and normal map refinement compared to authors' claims. While GeoNet++ attempted to enhance edges and geometric structures, refined depth maps exhibited distortions, hindering the refinement process. Refined surface normals yielded inferior results compared to initial maps.

To address this, we conducted our own training, obtaining new weights. Utilizing these parameters, we observed significant enhancements in depth and normal maps. This emphasized the importance of fine-tuning network parameters for specific datasets. Substantial improvements in depth map accuracy were observed, highlighting the efficacy of authors' modules in refinement. Refined surface normal maps showed visual improvements, though less significant than in depth maps.

Integrating GeoNet++ with COLMAP as our baseline network showed promising advancements in depth and normal map depiction. Refined depth maps exhibited improved object geometry. However, challenges arose due to distorted normal predictions and missing pixel values in the initial normal map, hindering the refinement process.

In conclusion, thorough examination and dedicated training underscore the importance of appropriate weights and data for improved results. While refined depth and normal maps address certain limitations, it's essential to note that not all limitations can be overcome, as GeoNet++ relies on initial maps for refinement. Nevertheless, GeoNet++ shows potential as a powerful tool for depth estimation and surface normal refinement. Collaboration with COLMAP holds promise for more accurate scene representations, opening avenues for enhanced applications in 3D reconstruction, augmented reality, and autonomous navigation.

Languages and Frameworks Used

During the course of this internship, I had the opportunity to work with a wide range of programming languages and technologies. The following are some of the primary technologies that I used:

- **Nvidia CUDA Toolkit:** Leveraged for training machine learning algorithms with GPU acceleration, significantly enhancing processing speeds and performance.
- **AWS (Amazon Web Services) Virtual Machine:** Utilized for cloud-based computing, allowing you to deploy and run machine learning models at scale.
- **TensorFlow API:** Employed for implementing Convolutional Neural Networks (CNNs) and other machine learning models, providing a robust framework for deep learning tasks.
- **Python Programming:** Served as the foundational language for coding and implementing various machine learning algorithms and workflows.
- **NUMPY and Pandas:** Used for efficient data manipulation, enabling you to work with large datasets and perform complex operations seamlessly.
- **MATPLOTLIB:** Applied for data visualization, aiding in the interpretation and presentation of results and insights derived from your models.
- **SCIPY:** Employed for data conversion to MAT files, facilitating compatibility and integration with other tools and platforms.
- **OpenCV (Computer Vision Library):** Utilized for a range of computer vision tasks, including Canny edge detection, a fundamental technique for feature extraction.
- **Google Colab and Jupyter Notebook:** Employed for interactive and collaborative coding environments, facilitating experimentation and documentation of your machine learning projects.

- Conda Environment: Utilized for managing virtual environments, allowing you to isolate dependencies and configurations for different projects.
- COLMAP (Structure from Motion Software): Employed for 3D reconstruction and point cloud generation, vital for understanding and processing high-definition maps.
- Octave: Utilized for evaluation metrics, providing a comprehensive environment for numerical computations and data analysis, particularly in the context of machine learning models.

In general, applying various programming languages and technologies contributed to MapmyIndia's HD map generation experiments.

Comparison of Competency Levels

Throughout my internship tenure, I had the privilege of refining both my technical proficiencies and soft skills, resulting in a notable elevation of my expertise in Deep Learning and Computer Vision Technologies. Prior to this experience, my endeavors primarily revolved around personal or smaller-scale projects with peers, offering limited exposure to large-scale corporate environments. The internship, however, provided a remarkable opportunity to engage in collaborative efforts within a team, working on substantial assignments and thereby fostering a comprehensive professional experience.

Within this internship, I made significant strides in enhancing my proficiency in Deep Learning. I actively collaborated with my advisors in the implementation of various algorithms integral to GeoNet++. This endeavor entailed a comprehensive grasp of the rationale behind each strategic decision. Moreover, I gained invaluable insights into the practical applications of these Deep Learning algorithms, emphasizing the critical importance of thorough model testing before deployment. Additionally, I familiarized myself with the operational workflows inherent to organizational environments.

Despite entering the internship with limited prior exposure to Computer Vision technology, I swiftly assimilated the foundational principles of this field, enriching my comprehension of how it aids in comprehending the intricacies of our surroundings. My proficiency in utilizing Computer Vision tools saw a notable advancement through my contributions to the GeoNet++ project.

Furthermore, my supervisors instilled in me the significance of punctuality and effective communication. This emphasis on timeliness and clear lines of communication proved instrumental in developing not only my work ethic but also my soft skills. The encouraging atmosphere cultivated by my superiors facilitated an open dialogue about any encountered challenges, ultimately contributing to the refinement of my communication abilities. Team cohesion and seamless communication were paramount throughout the training, ensuring the collective efficiency of the team and the steady progress of the project.

Appendix

GeoNet and GeoNet++ papers:

1. <https://arxiv.org/pdf/2012.06980.pdf>
2. <https://xjqj.github.io/geonet.pdf>

HD map references:

3. <https://www.MapmyIndia.com/hd-maps/>
4. <https://www.mapz.com/en/>
5. https://en.wikipedia.org/wiki/High-definition_map
6. <https://www.geospatialworld.net/article/hd-maps-autonomous-vehicles/>
7. <https://www.diva-portal.org/smash/get/diva2:1578797/FULLTEXT01.pdf>
8. <https://arxiv.org/pdf/2206.05400>
9. <https://www.here.com/platform/HD-live-map>

Point clouds:

10. <https://www.gigabyte.com/Glossary/point-cloud#:~:text=Point%20Cloud-.What%20is%20a%20Point%20Cloud%3F,external%20surface%20of%20an%20object.>
11. <https://flyguys.com/point-cloud/>
12. <https://holocreators.com/blog/what-is-a-point-cloud/>
13. <https://medium.com/analytics-vidhya/what-are-point-clouds-3655d565e142>
14. <https://www.anolytics.ai/blog/applications-challenges-with-3d-point-cloud-data-for-lidars/>
15. <https://www.sigarch.org/point-clouds-are-eating-the-world/>

Framework for HD maps:

16. <https://towardsdatascience.com/how-baidu-apollo-builds-hd-high-definition-maps-for-autonomous-vehicles-167af3a3fea3>
17. <https://www.mrt.kit.edu/z/publ/download/2018/Poggenhans2018Lanelet2.pdf>
18. https://www.ntnu.edu/documents/1284037699/1285579906/Gran-ChristofferWilhelm_2019_Master_NAP_HDMaps.pdf/79ef2eec-c9e2-454b-bf14-08d585cf8826
19. <https://www.asam.net/index.php?eID=dumpFile&t=f&f=4089&token=deea5d707e2d0edeeb4fccd544a973de4bc46a09>

Authors related work references:

20. <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/torralba-pami-02.pdf>
21. http://www.cs.cornell.edu/~asaxena/learningdepth/NIPS_LearningDepth.pdf
22. <https://users.cecs.anu.edu.au/~sgould/papers/cvpr10-depth.pdf>
23. http://www.cse.cuhk.edu.hk/leojia/projects/sblurdetect/papers/single_depth_shi.pdf
24. <https://arxiv.org/pdf/1406.2283>
25. https://openaccess.thecvf.com/content_iccv_2015_workshops/w10/papers/Shelhamer_Scene_Intrinsics_and_ICCV_2015_paper.pdf
26. <https://arxiv.org/pdf/1411.4734>

27. <https://arxiv.org/pdf/1704.02157>
28. <https://arxiv.org/pdf/1806.02446>
29. <https://www.cs.cmu.edu/~xiaolonw/papers/deep3d.pdf>
30. <https://arxiv.org/pdf/1604.01347>
31. https://papers.nips.cc/paper_files/paper/2016/file/65ded5353c5ee48d0b7d48c591b8f430-Paper.pdf
32. <https://arxiv.org/pdf/1805.04409>
33. FCRN: <https://arxiv.org/pdf/1606.00373>
34. Loss function: <https://arxiv.org/pdf/1711.03665>

COLMAP references:

35. COLMAP documentation: <https://colmap.github.io/index.html>
36. COLMAP GitHub link: <https://github.com/colmap/colmap>
37. Structure for motion:
https://openaccess.thecvf.com/content_cvpr_2016/papers/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.pdf
38. Multi-view Stereo: <https://demuc.de/papers/schoenberger2016mvs.pdf>

Other references:

39. http://www.math.iit.edu/~fass/477577_Chapter_5.pdf
40. NYUv2 Dataset: https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html
41. All images and figures taken from google images and above references mentioned.

