National College of Ireland

Depth Estimation for indoor environments using Augmented and Regularized Data through Knowledge Distillation

M.Sc. in Artificial Intelligence
Practicum

Utsav Pataskar
Student ID: 23195398

School of Computing
National College of Ireland

Supervisor: Kislay Raj

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Utsav Pataskar
Student ID:	23195398
Programme:	M.Sc. in Artificial Intelligence
Year:	2024
Module:	MSc Research Project
Supervisor:	Kislay Raj
Submission Due Date:	12/12/2024
Project Title:	Depth Estimation for indoor environments using Augmented and Regularized Data through Knowledge Distillation
Word Count:	7088
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Utsav Pataskar
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

National College of Ireland

AI Acknowledgement Supplement

PRACTICUM

DEPTH ESTIMATION FOR INDOOR ENVIRONMENTS USING AUGMENTED AND REGULARIZED DATA THROUGH KNOWLEDGE DISTILLATION

Your Name/Student Number	Course	Date
23195398	Msc AI	12th Dec 2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
NA	NA	NA

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Tool Name	Brief Description	Link to tool
NA	NA	NA

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

Additional Evidence:

Contents

Depth Estimation for indoor environments using Augmented and Regularized Data through Knowledge Distillation

Utsav Pataskar
23195398

Abstract

Depth Estimation is one of the important applications of computer vision which are further used in autonomous vehicles, robotics vision and AR/VR world. This research focuses on increasing generalization capabilities of depth estimation models on indoor settings which have low lightings, clustered and occluded objects and overall lack the diversity in terms of texture, has consistent and repetitive structural geometry. We deployed teacher-student framework to implement a ResNet-based pre-trained model as the teacher which will generate it's own pseudo depth maps from NYU-Depth V2 and Augmentations. The student model DenseDepth-169 based on U-Net learns from the teacher model and it's predictions. The proposal addresses overfitting and generalization problem by employing data augmentation and dropout regularization and increasing overall dataset size significantly. Edge Detection and contrast adjustment further aid in improving input feed quality. The research also provides a base for scalable and efficient indoors depth estimation models that are adaptive to diverse environments.

1 Introduction

Computer Vision with the help of OpenCV libraries has always been a topic of interest for many robotics enthusiasts. OpenCV allowed developers to explore the field of optics for Machine to mimic the visual sense of humans. Object Detection found multiple use-cases in the field of security surveillance and Autonomous Cars. Object Detection also has a military usage of Object Tracking while working with weaponry. Object Detection also gives rise to the importance of Depth Estimation. Depth Estimation is the concept of perceiving 3D structure in the surrounding world from 2D images either images being monocular or stereo vision. Depth Estimation is usually observed when we are working with Self Driving cars. We don't just want to detect obstacles, we also need to understand how far the obstacle is. Depending on these values, we decide when we start reducing the speed of the car and how fast the deceleration is expected to be. Depth estimation is also crucial for 3D reconstruction of images and helps the machine better understand the surrounding geometry. This understanding leads to enhanced interaction of the machine with the neighboring world. The Depth Estimation algorithms are broadly classified into two categories - 1. Monocular and 2. Stereo Depth Estimation.

1. **Monocular Depth Estimation** : Unlike humans who have biocular image for understanding depths, monocular images do not have second angled images for

Figure 1: Current State of Depth Estimation for Monocular Indoor Images

perceiving depth. This makes the task more challenging and thus the sole task needs to be focused on relative cues. Texture gradient from farthest object towards the image edges, object size gradient, occlusion, eliminating distortions are some of the features that are extracted in deep learning models for depth estimation. The most common use case of Monocular Depth Estimation is using it robotics which rely on single camera image data feed to process depth information efficiently.

2. **Stereo Depth Estimation** : In this approach, we have two images to mimic the stereo vision possessed by humans. The angle of two image is only slightly different. The goal is the consider this disparity to calculate the depth of the images. The depth perspection found from these data feed is more accurate than Monocular Depth Estimation, and thus find its usage in more complex and important tasks such as autonomous driving and high res 3D mapping

When it comes to indoor settings, depth estimation has a long way to go when compared to outdoor depth analysis. Outdoor settings have a variety of textures to capture when compared to limited texture variation in large surfaces like wall and ceilings making the task comparatively harder. Objects are occluded and cluttered together which increases the complexity and there is ambiguity in depth estimation. Another restriction we observe is that the object sizes vary drastically within an indoor setting from knife on the table to sofa set in the background, which further complicates the task. There is also limited diversity found for indoor depth estimation. However, despite these disadvantages, enhancing the work on indoor depth estimation can lead to many real-life applications. Older Roomba Vaccuum cleaners used to work on Infrared and/or ultrasound object detection mechanism for object avoidance. But now, newer Roombas work with Deep learning models pre-trained on images fused with traditional object detection to navigate within the house. Despite the advancements, these struggle while detecting reflecting and refracting surfaces. They also estimate depths to shorter distances and on larger objects, much research is being put for avoiding smaller and irregular objects (pet wastes). The low lighting also pose a challenge to overcome. For visually impaired people, indoor depth estimation can work miracles which will help them in navigating in enclosed places. These enclosed places will not only be the house, but malls and transportation ports as well, which has a lot of objects and bodies in a dynamically moving state. This goes to show that Depth Estimation plays a critical role in indoor navigation tasks as well. Meanwhile, exploring the Gaming Industry, in order to interact with virtual objects

in AR/VR, depth estimates are just as important. AR/VR worlds specifically are built around close proximity of the user so that the gaming experience is immersive and thus the need for indoor depth estimates. If the depth calculations are not accurate in the peripheral, gaming community might not experience the AR/VR to the fullest as misjudge causes disjointed experience. The AR/VR is important when it comes to creating sandbox environments as well. Running simulations within sandbox makes training for dangerous tasks like search and rescue operations easier for first responders. AR/VR cannot replace the actual dangers but can help them prepare better. And this training experience is heavily dependent on accurate interactions with AR/VR environment and in-turn on the accuracy of depth estimates.

Research Question:
How can advanced Deep Learning techniques be used to improve the existing depth estimation algorithms when used in indoor environment which have low lighting, object clustering and occasional occlusions and overall lack of texture and geometric diversity ?

2 Related Work

Monocular depth Estimation or Single Image Depth Estimation has always been a difficult task to achieve. The chances of losing the context information of occluded objects or occluded angles of visible objects is what makes this task error prone. And thus the need for more contextual information like stereo images was used for gauging depths. With the advancements in deep learning models, we started working with zero shot depth estimations, wherein we train our models on stereo images or semantically labeled dataset but the predictions on depth estimation is done on unseen images which are monocular in nature.
A simple undistorted image feed works with Depth Anything models, wherein we get the resultant depth estimation maps with precision. However with change in image feed changes the output. (Zhang, Juzheng et al.; 2024) worked on fisheye images. The fisheye images are taken with ultra wide angle having the capabilities to take panoramic views of 180 degrees or more. In case of circular fisheye lens, the camera can capture 360 degrees surrounding view,typically used in surveillances and AR/VR games. The images are heavily distorted such that straight lines are curved away from the center of the image (The barrel distortion). The bubble effect describes this distortion in images. (Zhang, Juzheng et al.; 2024) removed the distortion by deep learning models wherein the model was trained on left stereo images overlapped with right stereo counterparts and decoded against fisheye images. The curves with straightened out and then the straightened out images were feed to the second model of Depth Anything to get the depth estimations. This chained model resulted in depth estimation of fisheye images.
Another unique experimentation with Depth Anything model was performed by (Zhou, Keyu et al.; 2024). They used the powerful Depth Anything tool to depth estimate underwater objects. The experiment was not a huge success but they did concluded that the error in depth estimates was constant throughout the image. The error in depth maps did not spread exponentially as we moved in-depth into the images. They tried to eliminate this error by offsetting the weights but were only able to improve the results by 0.3%.
Over the years, many AI researchers and developers have tried to find more experiments

and research in the field of depth estimation and how to improve the existing technologies. One of the known issues is the loss of contextual information for occluded objects or in simple terms - the hidden side of fore-bearing objects is neglected in depth estimation. This creates a cascading problem for one of the real-life application of depth estimation, i.e., 3D reconstruction. (Bhat, S.F. et al.; 2023) faced the same challenge. ZoeDepth Algorithm used MiDaS algorithm trained NYU-Depth v2 dataset that was augmented with KITTI dataset for depth estimation. Furthermore, after acquiring the depth maps, the output was feed to reconstruct 3D mesh using Zephyr. Although they were able to reconstruct 3D mesh from Monocular images in their research, there were some information loss for occluded objects. Still their research drawbacks can be considered to be filled by (Jo, Seong-Uk et al.; 2024). Although not directly related, (Jo, Seong-Uk et al.; 2024) worked on reconstructing those occluded or partially occluded objects by splitting the images into multiple sets of images each of which contained independent objects. Occluded objects did not have any other object apart from background. And thus, it allowed for independent reconstruction of the occlusions with the help of iterative amodel depth estimations. This helped with recovering the contextual lose of information and the occluded object were already regenerated while they were in 2D format itself.

Second issue with depth estimation algorithm is its applicability on higher definition images. As we increase the resolution, so does the exponential increase in time taken while creating depth maps. (Li, Z. et al.; 2024) proposed on image segmentation. They used pre-trained models on low resolution images and targeted it for high resolution images. The way, this would work is to deploy a patch by patch depth estimation on each "patch" of the high resolution images. Numerous such patches would be targeted for depth estimation and then finally all the resultant depth maps will be collated (or fused) to create one high definition depth map. Thus the name PatchFusion. Therefore there experiment also consisted of creating a Global-Scale Awareness so that the relative depth estimation would match up with its neighboring "patches".

The most commonly referred known issues with depth estimation issues is that of the overfitting. Depth Estimations are work exceptionally well on the type of environment (indoor or outdoors) they are trained on. However, they perform poorly when exposed to others. This limits the generalization and adaptability to new images, and even the sensitivity to lighting conditions. Some known techniques to avert overfitting scenarios is by deploying regularization techniques such as data augmentation and data dropout and training models on different balanced (data distribution per class) settings. (Khanal and Sheshappanavar; 2024) proposed data augmentation enhancements via pre-trained models to enhance the image quality. These reduced the mis-classification of objects. The super-resolved enhancements were were passed to a U-Net after a series of linear transformation. The before mentioned diffusion takes place through a Variational Autoencoder (VAE). This step further aids in feature extraction. The super-resolving objects of objects are resulted in better visibility of shinier or more transparent objects which are easily missed in depth maps.

Alternatively, (Yoo et al.; 2024) explored Conventional as well as Synthetic data augmentations techniques. Flipping (or Mirroring) data, Resizing after Cropping, Adding Gaussian Noising to create new data and adjusting brightness are some of the conventional methods. On the other hands, Synthetic-based data augmentation techniques use Masking - where the images are segmented into multiple different regions and put under spot light, and these masks are overlapped onto original images to create new data. Secondly, Mark-scale is similar to what Masking combined with Resizing and rescaling

can do. Mark-scale essentially masks regions of interests and re-attaches it to image with different scale and original. Smaller objects can be used as an indicative of distant objects. These are some great methods for generalizing the model and reducing over-fitting scenarios. Although (Yoo et al.; 2024) experiment did not show much significant improvement, but it still was an improvement.
(Jun et al.; 2024) proposed a new methodology for data augmentation by synthesizing more data using a combination of Masking, Mark-scaling and using CutFlips. They got a pair of left sided and right sided images and then they flipped the Cutflips L to get symmetric images. Same going for Cutflips. This process essentially doubled the training dataset. Now this process is repeated for respective Depth maps as well, resulting in corresponding depth maps of newly created dataset. They concluded that using multiple combinations of data augmentations techniques generated the best results for training.
Another Methodology that is widely used to avoid over-fitting extreme machine learning scenarios is to use data dropout regularization process. Few such strategies are (firstly) Random Pixel Dropout, forcing model to divert the focus from individual pixels and onto the overall structure. Secondly approach is to drop random spatial squares from the images for it to simulate black dropped out patches as occlusions. This will train the model to understand and predict depth maps when partial views are present. And lastly, selecting those features which are too complex and consider them to be noises, will allow the model to focus on significant areas. (Zhao et al.; 2024) pointed out depending on dataset with small size, information cannot be extracted beyond few layers. Thus emphasizing on the need of data dropout regularization techniques to improve generalization on image feed that is smaller in size. (Martinho et al.; 2024) also deployed dropout regularization while experimenting with depth estimation on underwater images. They randomly dropped out image units while training to avoid the situation of co-adaptation and thus reduce over fitting as well.

Critical Analysis and Limitations:

- **Advancements in Monocular Depth Estimation:**
  Even with the advancements in Monocular Estimations, it still facing many problems due to context information loss and occluded or partially visible objects, is crucial task to work on. The Zero-shot estimation that uses semantic label is a noteworthy technical leap. ZoeDepth and MiDas are some great examples that showcased the potential for depth estimation in different settings. Still an holistic contextual information loss remains a problem when reconstruction of occluded or partially visible objects is concerned. Even though works like (Jo, Seong-Uk et al.; 2024), provided remediate for depth estimation in occluded objects, the task remains to be computationally heavy.

- **Distorted and Specialized Image Depth Estimates:**
  With advancements in variations of camera and their respective photographs, we observed more complex system to depth decode these complex images. (Zhang, Juzheng et al.; 2024) applying chain modeling, one for correcting the barrel effect and second for depth estimation, allowed us to explore the real-life application of AR/VR surveillence which uses fisheye images. (Zhou, Keyu et al.; 2024) explored complex domain of underwater depth estimation which can be considered as an specialized area. However, their project findings still need further refinement to handle complex visual cues. We infer that pre-processing steps remain domain specific and cannot be extensively generalized. Specialization also applies to Evaluation metrics

as no one metrics are fit to evaluate generalized model

- **Overfitting and Generalization Challenges:**
  Depth estimation models are notoriously known for their ability to exhibit overfitting behavior and their lack of ability to adapt. Depth Model work particularly well for the environment they are trained on, but give face challenges when models try to generalize, it is specially observed when indoor train model are generalized on outdoor testing dataset and vice-a-versa. (Khanal and Sheshappanavar; 2024) enhanced shiny object detection for handling the uneven lighting indoor settings and (Martinho et al.; 2024) work on dropping the same shiny objects from depth estimation increased the accuracies. (Zhao et al.; 2024) dropout also showed robustness post data dropout of smaller objects.

- **Data Augmentation Challenges:**
  Even though we are able to synthesize complex data, we fail to mimic the real world indoor simplicities. The Data Augmentation can introduce more diversity in terms of unstructured geometry, it can also introduce biases. And even though we observe increase in generalization capabilities of the model, the gain is incremental. We observe the trade-off here as well, synthesizing data can help us better generalize the model, but we stray away from real-life scenarios resulting in difficult testing phase, but if we do not augment the data, we face data shortage issue for indoor settings and fail to more generalize the model.

Our Project will focus on handling generalization problems for which we will be using an outdoor pre-trained model to infer depth information from indoor settings which can then be instilled into another model for depth predictions.

3 Methodology

3.1 Teacher-Student Framework

We are deploying Teacher-Student model that will be used to train our model on the basis of a pre-trained model weights presented by (Kumar et al.; 2024). They presented a multitude of approaches for Data Augmentation techniques and Data dropout techniques which helped them in generalization of depth estimation models and handled the lack of diversity issues with exterior images. We are going to add data augmentation techniques to enhance depth estimation techniques to increase the algorithms scalability on the student model. Since we are working towards the same goal/task of depth estimation, we are transferring knowledge between teacher-student model and thus has an advantage over transfer learning. We are more concerned with the model compression and refinement tasks on student model, rather than focusing on domain adaptation task. The way this "learning" works is that the results that are produced by teacher (pseudo depth maps in our case). Need to be duplicated by student model as well. Considering the teacher model did not misproduced the results, student model will learning without significant amount of performance loss.

3.2 Pre-trained ResNet Model

We are using the pre-trained model with pre-trained weights of resNet "as-is" to generate pseudo depth maps on the augmented indoors data. The teacher generated depth maps will be used as training input for the student model. The predictions created by teacher model are saved in an .npy (Numpy) file. Here-in teacher student model, the above mentioned .npy will serve for supervised learning of the student. It is imperative to note down that this process will not fine-tune or re-tune the teacher model. Teacher model has 2 purposes in this process, firstly to use it's experience to extract features and secondly to predict or generate pseudo-depth maps. We resize the data augmented RGB images' resolution to feed that of teacher's accepted input resolution. The images are then converted into tensor suitable format for inferencing results. The inferencing steps comprises of features extraction and decoding for generating imitations of depth maps. For post-processing we use bilinear interpolation to resize the image back to it's original resolution.

3.3 Student Model Architecture

The student model is a DenseNet model which is based on U-Net structure with attention mechanism (The Squeeze and Excitation block) . DenseNet is known to avoid vanishing gradient-problems, can strengthen the propagation and re-usage of features that are extracted and all of this can be performed over much lesser number of training parameters (as compared to resNet).

- **Backbone: DenseNet-169:**
  This acts as the encoder that is responsible to extract multi scale features from the image feed. The DenseNet-169 is pre-trained on ImageNet. We draw skip connections from in-between feature/depth maps from the image feed. The *conv1_relu* handles low-level features, *conv2_block6_concat* for Early intermediate features, *conv3_block12_concat* for Deeper intermediate features and *conv4_block6_concat* for high-level semantic features. The Image feed expects the shape to be 320x320

- **Decoder:**
  The task of decoder is to reconstruct depth maps from feature maps that were encoded DenseNet-169 encoder.
  First Upsampling Block takes High-level features from *conv4*. We double the spatial resolution here and combine features having skip connection that derive from *conv3*. This layer has 512 filters, *ReLU* as the activation function with a kernel size of (5,5). We introduced an SE block to focus on attentions channel-wise and then dilated thee convolution to have 256 fiter with kernel size of (3,3)
  Second Upsampling Block takes input from the first block. The Upsampling process is repeated with double the resolution and combines features from *conv2* skip connection now. In this block, we further reduce the filter count and kernel siz from 256, (5,5) to 128, (3,3).
  Third Upsampling Block is feed for second block output and we repeat the Upsampling again with *conv1* skip connection, the filter count is still halved to 64 but the kernel size is kept constant in this block.
  Final Upsampling Block has only doubling task of the resolution so that it matches the input image size of 320x320.

Throughout the initial testing, the dilation rate is kept at 2 and the activation function is ReLU.

- **Output Layer:**
  This layer has only 1 filter and works with sigmoid activation function. This layer produces a single channel depth map scaled from 0 to 1.

- **Attention Mechanism:**
  The SE block enhances the model's capabilities to focus on informative features to establish channel-wise relations. It summarizes the spatial information.

- **Loss and Optimization:**
  we used RMSE, The which inflicts even more penalty of outlier but still is receptive to interpretations derived from deviations. In RMSE, we see the errors in the same unit to that of depth values.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d}_i)^2} \tag{1}$$

  - $\boldsymbol{N}$: number of pixels or observations.
  - d_i : the actual ground truth value of the i^{th} pixel.
  - $\hat{d}_i$: the observed, model predicted value of the same pixel

  We utilized the Adam optimizer because it can adjust the learning rates for each trainable parameters in the encoder and decoder based on the mean and the variance of the gradients of the feature maps. The learning rate adjustment is crucial to depth estimation domain. Certain regions within the image require more focus and precision to learning rate updates than other regions. Depth Estimation is also a processing heavy task thus Adam optimizer is known to converge faster than other optimizer. We reduce the training time while maintaining the accuracy of the model.

3.4 Exploratory Data Analysis

3.4.1 Dataset Description

We intend to use the NYU-Depth V2 dataset which is most frequently used when working with indoor depth estimation models. The dataset also contains semantic labels. It is a very high quality depth map collection specialized to capture indoor settings. The dataset contains 1449 labeled images which will be used for supervised learning. The dataset is represented in a triplet format - An RGB Image, A Depth Map and Semantic Labels (ref 2). The Depth The images (RGBs and Ground Truths) are in the resolution of 640x480 pixels and the images are stored in as a .mat file (MATLAB file format), we choose the 1449 image dataset over the raw image dataset of over 220,000 image dataset due to lack of computation power accessible at the moment. Another reason to skip raw dataset is it requires even higher level of pre-processing on removing the noise and handling lighting conditions. The challenges that we encountered with using this dataset is the lack of diversity in geometry, occlusion of objects and lighting conditions. This dataset is also not useful towards application which expects outdoor images.

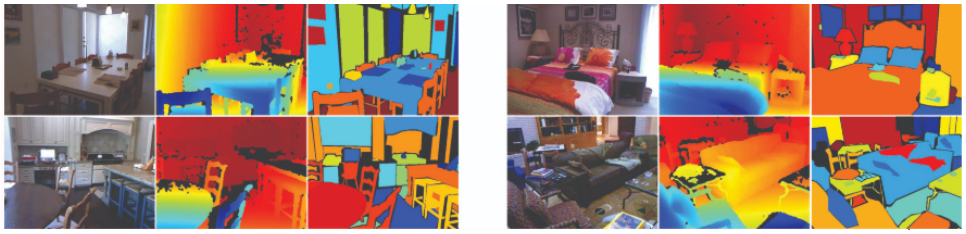

Figure 2: Extract from NYU dataset website

3.4.2 Pre-Processing

We propose incorporating a pipeline for normalizing, to adjust the brightness levels within the indoor settings and enhancing edge detection capabilities. Each of these steps will help us to extract meaningful features for depth estimation tasks. Normalizing the depth images makes the depth maps (ground truth) to a 8-bit scale i.e. 0 to 255 range. This ensures compatibility with further pre-processing for images. The normalized images then are fed to adjusting brightness, so that shiny reflecting surface do not disturb the neighboring depth map density while predicting depths. On the other hand, adjusting contrast allows us to detect edges much more distinctly. Once we have prominent edges to detect, we can apply segmentation and work towards enhancing the depth map predictions

3.4.3 Data Augmentation and Data Dropout Regularization Techniques

The limitation of indoor dataset is the lack of diversity of unstructured geometry to help train the model generalize better depth maps as well as the lack of quantity of images we are using from NYU-Depth dataset,it is imperative to create more diverse dataset using Augmentation and Dropout techniques. We will apply geometric transformation like shearing and advance techniques like cutFlip and block & stipe dropouts to unstructure the geometry from RGB and it's corresponding depth maps.

3.4.4 Feature Analysis

One of the key features which are visually perceived from ground truth depth maps from NYU-Depth V2 is the map has a sharp transition in shades when it crosses an edge. Within the boundary of an object (edge bound objects) there is only a monotonous smooth gradient observed.
When considering spatial and statistical properties of the images. In any average indoor setting, The depth varies within 0 to approximately 10m. Majority of the images have a maximum depth of 5m and very few images have deeper settings. This cannot be considered as an imbalanced dataset because the aim is to work the depth estimation within the room itself.
It is also noteworthy that the encoder feed have varying intensities of RGB with sufficiently noticeable differences in brightness and contrast.

3.4.5 Data-Driven Model Selection

Based on the above mentioned feature characteristics, we concluded on the following model selection.

- Teacher Model : We scoured for a model which has was pretrained on outdoor settings and came across monodepth2 (Kumar et al.; 2024). ResNet is robust and scalable in nature when it comes to capturing structural patterns within depth maps. The monodepth2 (Kumar et al.; 2024) was trained on huge dataset which made it an ideal candidate from the knowledge distillation onto our DenseDepth-169 student model.

- Student Model : We selected DenseDepth-169 model for it's capabilities of efficient learning and reusing features through a series of layers which are densely connected (the encoder-decoder mesh). For depth estimation, we need to extract multi-scale features and the DenseNet can efficiently resuse these. To Student Model, we also added Squeeze-Extraction block which further allowed the model to focus on relevant feature blocks.

3.4.6 Data Challenges

One key challenge while working with NYU-Depth is that the images with near depth objects are over-represented, meaning we have a huge chunk of dataset which has objects within 5m range. On the other hand, the images with further distances are under-represented. And this problem cannot be entirely solved with our proposed data augmentation techniques. NYU is also inherently noise polluted due to the limitations of capturing depth indoor precision with the kinnect sensor.

4 System Architecture

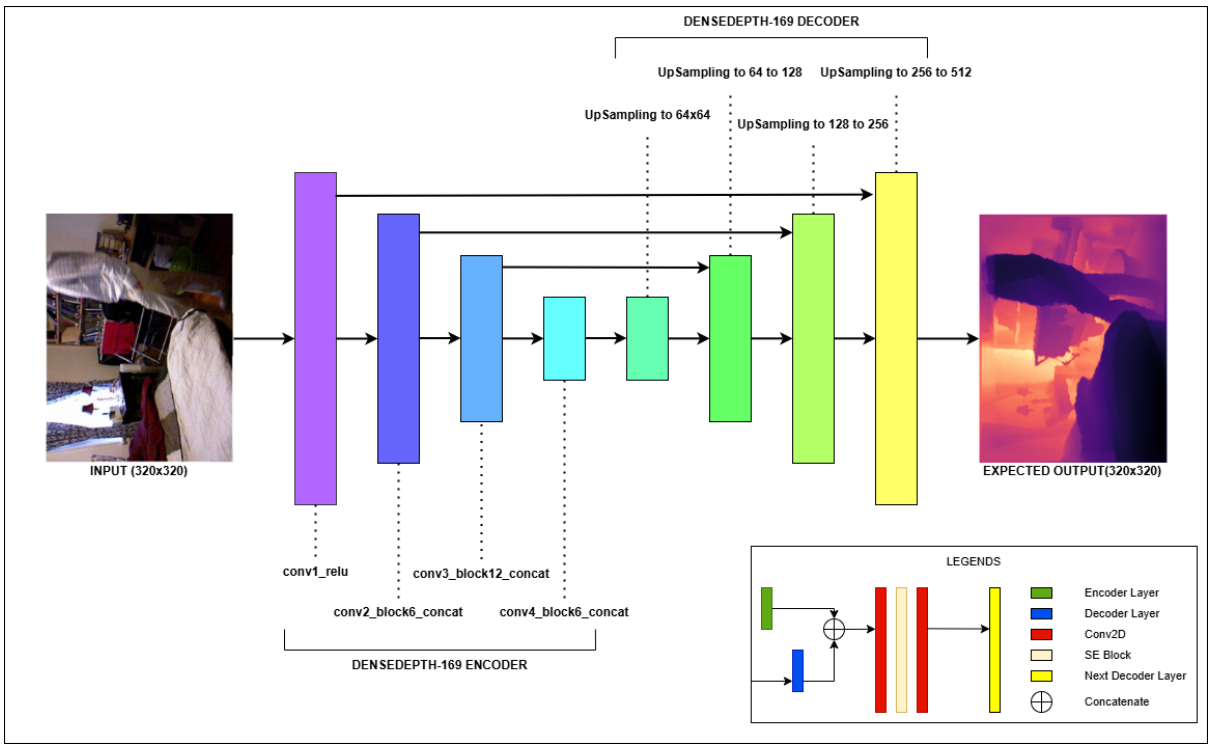

Figure 3: Student Model DenseNet Encoder-Decoder Architecture

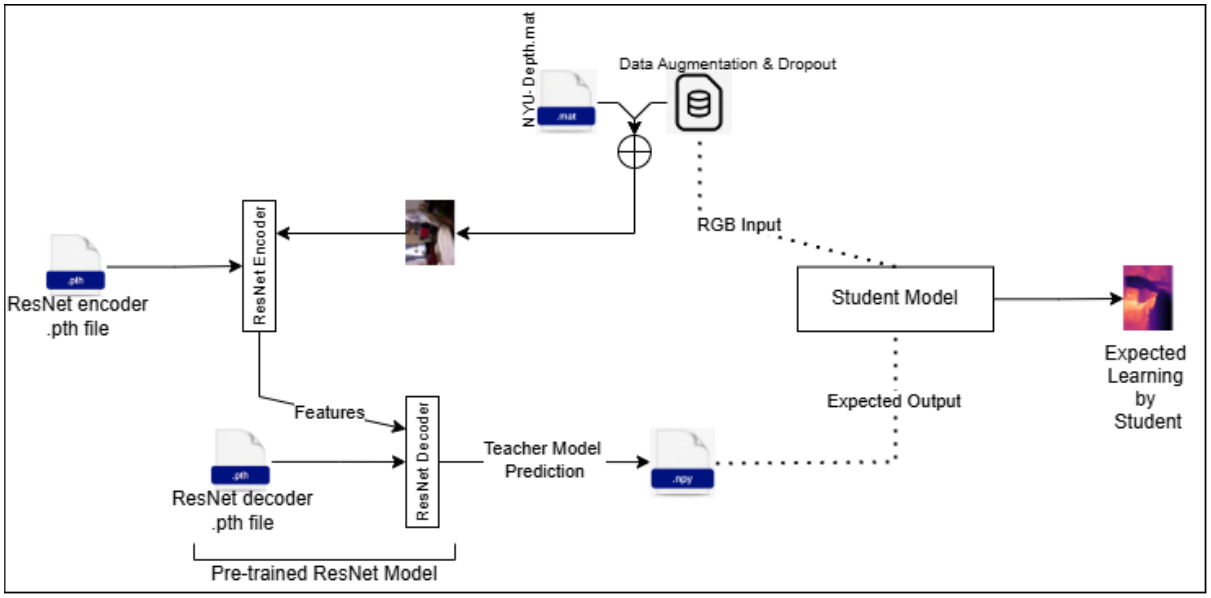

Figure 4: Teacher Model - Knowledge Distillation

5 Implementation

We divided the project into following tasks.

5.1 Raw Image Dataset Pre-processing

- Building Depth Estimation Models on following Libraries: U-Net, DenseDepth and ResNet-18 without most pre-processing. This did not yield us even remotely satisfactory results. We only adjusted the brightness of the images before predicting the depth map of the test image. By leveraging the capabilities of PIL (Python Imaging Library), we adjusted the brightness of the image by using ImageEnhance to multiply the image with a constant factor. Having the factor 1 indicates the image in original lighting. Reducing the value below 1, will decrease the brightness and eventually we get a pitch black image at 0. Similarly increasing the value above 1 will add brightness and eventually will white-wash the depth maps. As evident

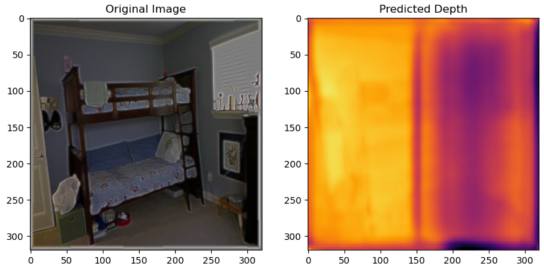

Figure 5: Output with Pre-processing - Reducing brightness

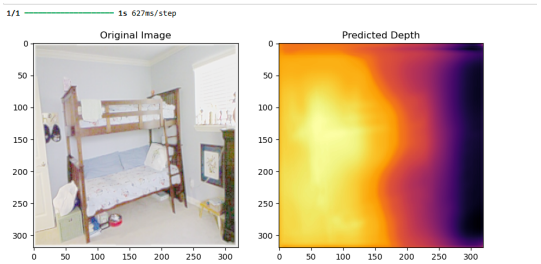

Figure 6: Output with Pre-processing - Increased brightness

from the images, the brightness of the image distorts the depth map prediction. It became evident that we need to tone down the brightness in order to get better results on our predictions. The brightness within the original images can also be

indicative of the presence of shiny objects which reflect high intensity lights back onto the camera and thus can lead to even more distortions while predicting depth maps. We observed U-Net gave us good results with darker images. And thus we proceed with that model for further processing.

- Edge Detection trained on U-Net: After adjusting the brightness of the images, we turned to enhancing the edge detection within the image. The more identifiable and detectable the images are, the more easier it will be for model to train. The Edge Detection algorithm is applied on both the images - the RGB image and the Depth image (Ground Truth) Images. By visualizing clear boundaries in the depth images, we make the model detect objects which are smaller or occluded from vision to be considered for depth estimation. We deployed the Canny library which performs really well when it comes to edge detection.

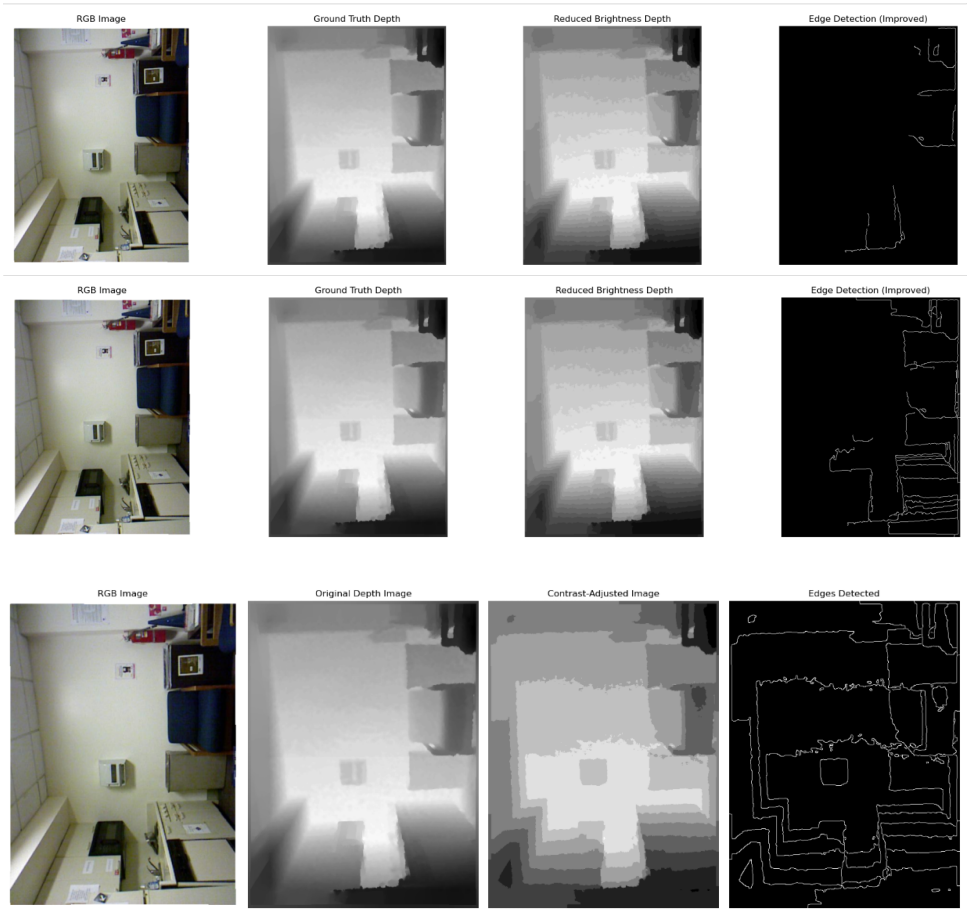

Figure 7:
Top - Preprocessing without Gaussian Filter - Less clarity output,
Middle - Preprocessing with Gaussian Filter - Slightly Better results,
Bottom - Preprocessing with Edge Detection with Contrast Adjusted and Gaussian Filter - much better results

The top layer shows raw edge detection with no change in brightness, which results in less edge being detected. In the bottom row, we applied Gaussian filter to reduce noise in the image by smoothening pixel density. We get better edge detection

and more robust image feed for depth estimation. We also decide the threshold for minimum and maximum intensity which can be considered for a line being an edge in the image.

Gaussian Filter gave better results for our dataset and experiments as compared to Bilateral filtering. As we can see, standalone Edge Detection do not provide us with necessary results, it also is insufficient in detecting edges in the ground truth.

- Thirdly we applied contrast adjustment to refine the boundaries which further aid the cause of depth estimation. Ref Figure 7. Contrast works similar to adjusting brightness, by adjusting thresholds in the ImageEnhance function, we can create sharper or duller images. For Sharp contrast, we see vibrant colors and distinct borders, and results in better edge detection but it also amplifies noise thus we see random borders within the images which are not true edges. Conversely, low contrast can cause us to reduce noise but consequently miss out some important edge. The resultant edge detection is somewhat closer to what we want in depth estimation maps. By Adjusting contrast within the RGB image, we get sharper features that can be extracted by our U-Net student model.

We need to balance out contrast, Gaussian Filter threshold and brightness factors to achieve better results.

5.2 Data Augmentation Techniques

Data Augmentation Techniques are frequently used when there is a quantity shortage for any image processing tasks as well as when there is a dataset imbalance. For our task of depth estimation, the teacher model has been extensively trained on outdoor or exterior images. Outdoor images are scaled widely (in terms of meters to kilometers), have majority of non-structured geometric environments and have less controlled lighting. When such a model is used to train our U-Net student model trained on indoor settings, the model has a limitation of overfitting when it comes to indoor images. Indoor settings also have a limitation on quantity of repetitive geometric structures (rooms and corridors) and thus harder to train. We increase our odds on model training by using Data Augmentation Techniques such as geometric transformations and synthetic data generation for increasing the existing already limited dataset. NYU-Depth v2 dataset which is being used to train has only 1449 image x 2 (RGB image and Ground Truth Depth maps) for training purpose. We apply the following techniques:

5.2.1 Geometric Transformations

- **Image Rotations and Image Flipping:** Even though the task is simple, it effective to introduce generalization of the real-world images. We remove the orientation dependency while training the model by introducing mirrored and rotated versions of the same images.

- **Shearing,Elastic Stretching and Cropping:** Image Shearing is responsible for simulating perspective distortions in monocular images. Cropping helps us the focus (or defocus) from certain regions of interest. In indoor settings, if there are shiny reflective surfaces, the depth estimation model distorts the ground truth and needs to be considered noise. On the other hand, if have larger volume of noise, but some

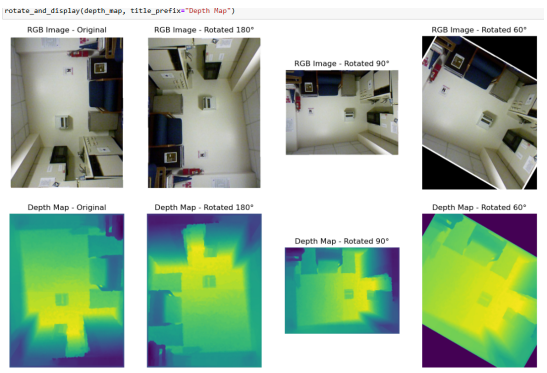

Figure 8: Rotation Geometric Transformation

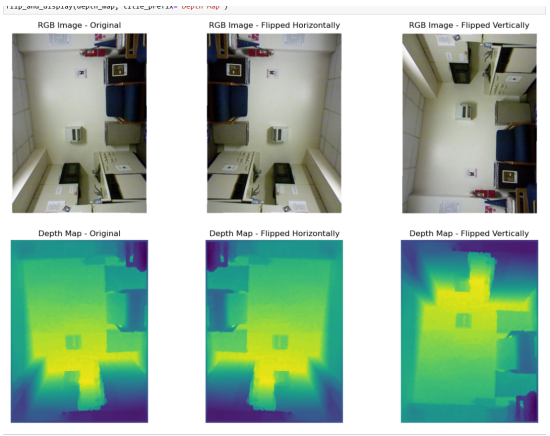

Figure 9: Flipping Geometric Transformation

regions of the image are infact useful, we focus only on those images. Training the model to omit or view certain regions only is another model itself. Ref [10]

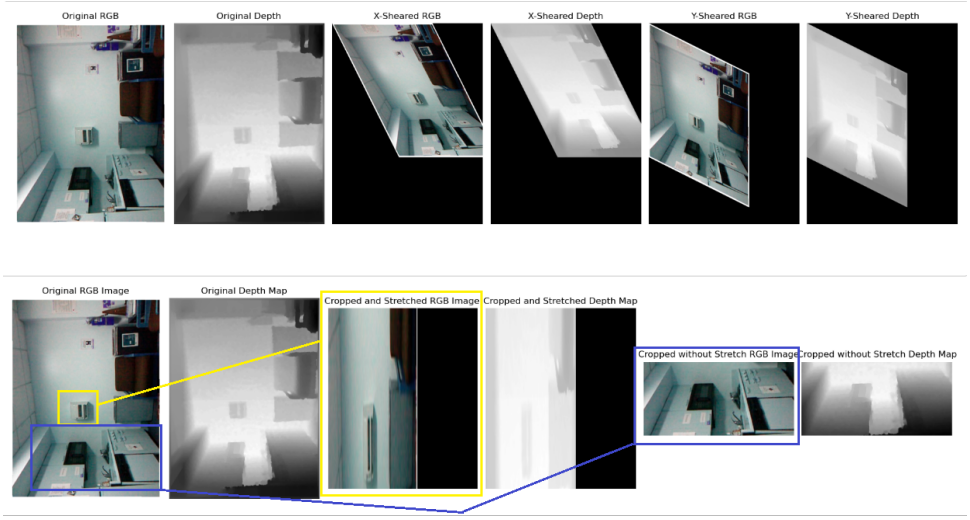

Figure 10:
Top row - Shearing Along Axes
Bottom row - Cropping and Elastic Transformation

The new dataset we get from Geometric data transformation is saved in host location which will further be used as training material for teacher as well as student model.

5.2.2 Synthetic Data Generation

As stated earlier, we do have a quantitative limitation on the training set when it comes to indoor images. And even if we overcome this challenge by adding more and more data points, we eventually are faced with repetition of content of the same geometrical structures (room, furniture, corridors, cabinets, etc). Thus reducing the ability of the model to generalize depth estimates.

- **Slicing and reconstructing Images:** One approach we used was to slice and dice the images and depth maps into 4 quadrants and reconstruct the images using

geometric alterations to create a new image. In order to slice, we convert the images into numpy array, halved the width and height and stored the image in an object. Since we are working with numpy arrays, we can easily shuffle the indices

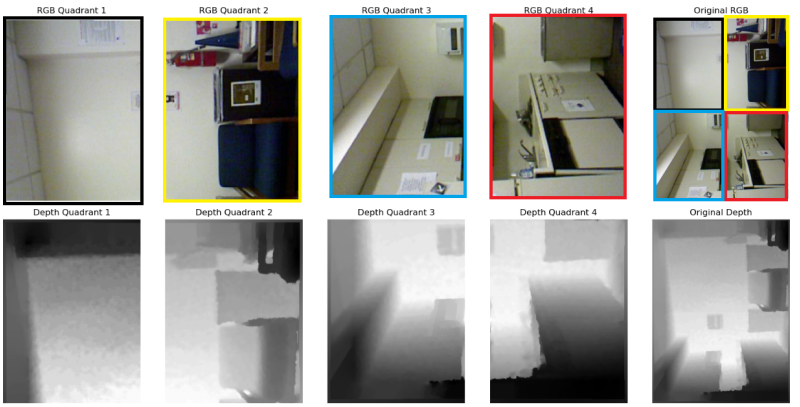

Figure 11: Initial Split (640x480) into 4 (320x240) images

```
RGB Quadrant 1 shape: (320, 240, 3)
RGB Quadrant 2 shape: (320, 240, 3)
RGB Quadrant 3 shape: (320, 240, 3)
RGB Quadrant 4 shape: (320, 240, 3)
Depth Quadrant 1 shape: (320, 240)
Depth Quadrant 2 shape: (320, 240)
Depth Quadrant 3 shape: (320, 240)
Depth Quadrant 4 shape: (320, 240)
Original Image Resolution :  (640, 480, 3)
Original Depth Map Resolution :  (640, 480)
```

Figure 12: Resolution output

to concatenate a new image. A sample of programmatically synthesized images is presented below We see that by deconstructing a single image, we created atleast

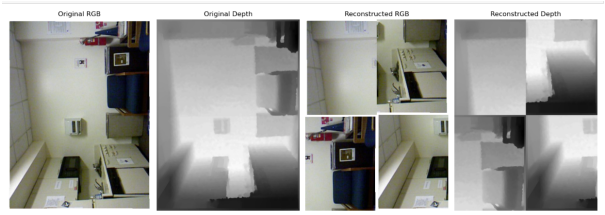

Figure 13: Shuffle Quadrants

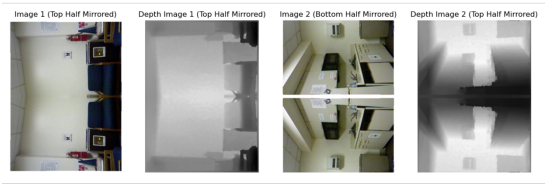

Figure 14: CutFlips Vertical

3 more images, first my shuffling the quadrants, second and third by mirroring and combining top half with top half and bottom half with bottom half. The more we play around with it the more dataset that can be generated.

5.2.3 Data Synthesizing by Data Dropout

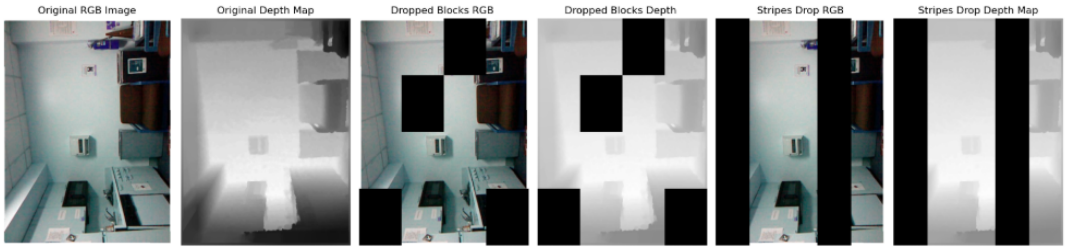

Figure 15: Data Dropout Techniques

Data Dropout techniques are used to dropping out or blacking out specific blocks of within the original image and it's corresponding depth maps. By Dividing the image into X,Y matrix and then setting certain numpy blocks to zero makes the dropout easy. By

Dropping out we get the above results. This technique is specifically used when we have shiny or reflective object which cause distortions in machine learning. By dropping those objects (or in some cases reducing brightness of those blocks) we increase our chances of getting more accurate depth maps.

In order to save the generated images into the local machine, we use the PIL libraries for the save function and *os* library for generating the relative path to indicate the store location.

This way, we introduce unstructured geometry into our dataset to add diversity for better generalization on indoor settings. From the evidence presented above, we observed that a single RGB image with it's corresponding depth map can generate atleast 15 new images. With 1449 RBG-D images from NYU-Depth-v2, we can easily have over 20,000 images with their generated respective depth maps. We are currently not deploying Cut-Mix techniques where-in we apply the above mentioned operations with 2 different images. We slice the images into halves and combine two different halves.

5.3 Experimentation

5.3.1 Without Pre-processing and Without Data Augmentation Techniques

Our Initial approach was to implement the teacher-student model as-is using NYU-Depth V2 dataset without any data augmentation, or any data pre-processing. The teacher model was pre-trained on multiple data points for generating outdoor depth maps (Kumar et al.; 2024).

Figure 16: Teacher-Student Model baseline(without pre-processing) results

Now we exposed the teacher model to learn from current 1449 base images and ex-

pected it to produce the pseudo results. This results were supposed to be replicated by the student model. The first experiment was done without any pre-process as we need to present a baseline of depth maps that were generated out of the box. This will also be used to compare side by side the improvement we observed with more pre-processing we did with the images, as well as the impact of increasing data points by data augmentation and data dropout techniques would have on the depth estimation scenarios. The illustrated images in 16 indicates that pre-processing was of the essence. The student model was barely able to predict depth maps. It displayed only 2 regions of interest, the closest with bright red light, and black (dark blue) as the furthest points. The regions identified with current but were not smooth. Thus we started training our models with augmented dataset and data pre-processing.

5.3.2 With Pre-processing and Data Augmentation Techniques

We did not pre-process the NYU-Depth V2 dataset on the fly. Due to the project being processing intensive and us lacking the resources to train the entire model in one go, we pre-processed, created a new dataset and stored it into the local for future usage. This way, we reduced the burden of single click execution.

6 Evaluation & Discussion

The Knowledge Distillation by teacher student framework succeeded in concluding our study on how to address the generalization problems from outdoor pre-trained model to help estimate depths in indoor environments. This following section gives valuable insights on the model performance for the experiments and implementations.

6.1 Experiment 1

We implemented the teacher student framework with pre-trained ResNet model on outdoor environment and knowledge distillation onto student model with NYU-DepthV2 dataset with no pre-processing and no augmentation techniques. We observed that with limited dataset, the model lacks capabilities to completely generalize our DenseDepth model as 1449 images have repeated structures and limited diversity (Ref 17). This showcased that we need to pre-process the data before it is ready for depth estimation. Thus we turned to experiment 2.

6.2 Experiment 2

After realizing the need for pre-processing, and keeping the input count the same, we normalized the data, implemented Gaussian filter, reduced the brightness and adjusted the contrast according to images before feeding the dataset to pre-trained as well as our DepthDense model. We noticed that we have a slightly better results when it came to generalization and to the overfitting situation (Ref 18). The problem for lack of diversity and structural geometry still posed a problem and we could not improve the model beyond a certain limit.

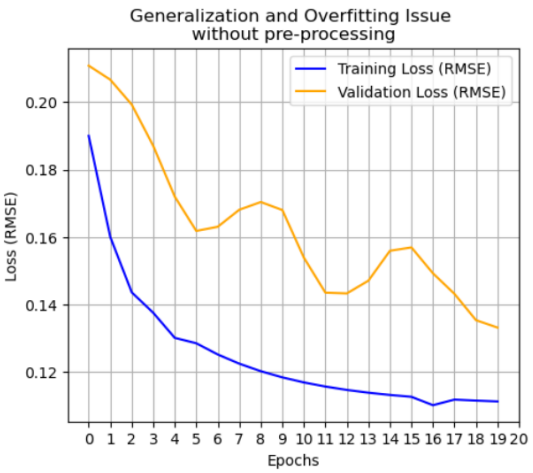

Figure 17: Performance without Pre-processing

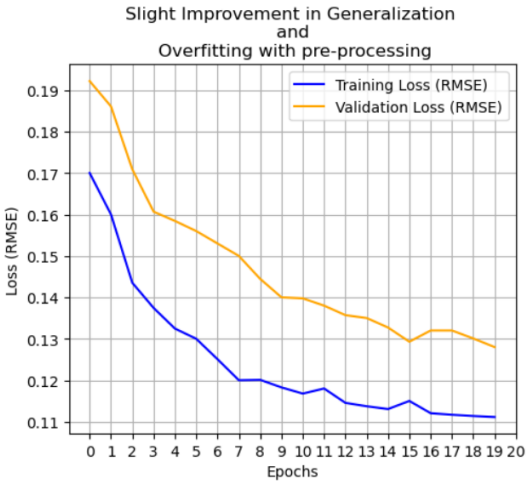

Figure 18: Performance with Pre-processing

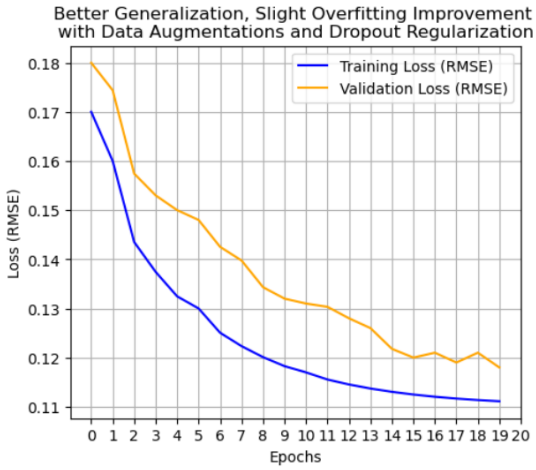

Figure 19: Performance with Data Augmentation

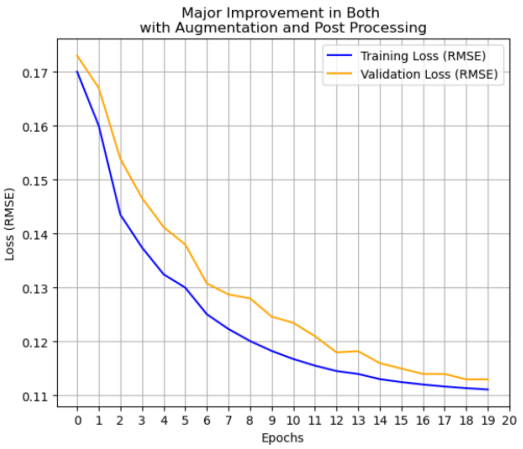

Figure 20: Performance with Augmentation and Pre-processing

6.3 Experiment 3

We synthesized more data by simple geometric transformations which increased our data count, we also deployed data augmentation and data dropout techniques so as to insert a certain degree of unstructured geometric complexities into out training set. mirrored flips and blocks and stripes droput made the structure feature seem complex enough for the models to learn from. While we experimented with Augmentations, we did not pre-process the images and thus we observed much more improvement for the model to generalize compared to slight improvement in overfitting scenario (Ref 19).

6.4 Experiment 4

The final experiment combined our work of pre-processing and data synthesizing to offer us much better results than the preceding experiments. Data Augmentation and Dropout Regularization handled the generalization problems by reducing overfitting the student model. Pre-processing further aided in achieve better generalization of model (Ref 20).

6.5 Computational Comparison

6.5.1 Experimental Computational Time Comparison

Experiment Name	Experiment Description	Filter Count (Each Layer)	Training Time (hrs)	RMSE (metres)
Experiment 1	NYU-DepthV2 (1449 pairs) no pre-processing, data augmentation or dropout regularization	512,256,128,64	8	0.16
Experiment 2	1449 NYU-Depth V2 pairs, Gaussian filter, Brightness & Contrast Adjustment and Edge Detection	512,256,128,64	11	0.14
Experiment 3	Synthesized 20K+ data by Geometric Transformation, Data Augmentation & Dropout Regularization No Preprocessing	256,128,64,32	17	0.13
Experiment 4	Synthesized Data with Pre-processing	128,64,32,16	15	0.12

Table 1: Computational Comparison Across Experiments

From table 1, we see that the dataset count, pre-processing count and filter count all have significant impact on the computational time of the student model. All these experiments hav been performed on Intel i7 - 11^{th} Gen, NVIDIA GeForce RTX 3060 GPU (Driver Vrsion 566.03) and Cuda version 12.7

6.5.2 Performance Comparison with SOTA Indoor Depth Estimation

Model/Approach	RMSE (meters)
DenseNet-169 Model (with augmentation + post-processing)	0.12
SOTA DenseDepth	0.12-0.14
SOTA Monodepth2	0.10-0.13
SOTA DeepLabV3+ with depth estimation	0.12-0.16

Table 2: Comparative RMSE values for different indoor depth estimation models

From table 2, our results are competitive with SOTA, but still needs fine-tuning

to achieve better results. Exploring multi-scale prediction can further aid our cause of reducing RMSE.

7 Conclusion

While we worked on reducing the overfitting situation while working with pre-trained model on outdoor images, the pre-trained weights were used to create pseudo depth maps as numpy files for the student model to learn from. We established that accuracy of teacher model caps the limits of student model. Teacher has to deliver better predictions for the student to mimic those results in this kind of transfer learning. We also need huge sample set to address the overfitting situation and to proceed with generalization of the DenseDepth model. Data Augmentation and Dropout plays a crucial part in synthesizing couplet of RGB and depth maps which introduces unstructured geometry. It is observed that neglecting the object clutter that are smaller in size also reduces the burden on the model for complex depth maps. The filters within the Student model should be kept similar to what the pre-trained teacher model has. The filter count needs to be changed according to the goal of the estimation. If we need to have deeper more complex features, if we are working with high resolution images and have the necessary computing environments, it is beneficial to increase the filter count. If wee need to prevent overfitting and have smoother spatial generalization in the predictions, we choose to decrease the filter count. It is also noteworthy, when working with augmented data, when we wish to reduce the memory usage and promote efficiency, decreasing the filter count is the way to go. Adjusting the filter counts in the student model smoothened the maps. The Squeeze and Extract blocks proved to be critical in address the loss issue. The study does acknowledge the computational restraints and the dataset biases limit the performance. Pre-processing in terms of edge detection, varying contrast are also vital steps to smoothen the prediction maps. The pre-processing remains domain specific and is difficult to generalize for all environments. Implementing Post-processing as an when required also helps smoothen the predictions. Reducing the noise, smoothening the map while preserving the edges is achieved by Bilateral Filtering.

8 Future Work

The Research has still a long way to go, which is currently limited by computational power possessed by us. Here are the list of future scope for this study.

- The work on identifying domain adaptive techniques like Domain-Adversarial Neural Networks can help the model classify the domains which then can learn more about domain-independent features which can work towards unrestricted generalization of the model.
- Integrating More data augmentation techniques like cut-mix where we cut and combine two different images to synthesize even diverse data can help us better generalize the model. This step however still put burden on the computation.
- Another drawback that was observed during this study was the lack of benchmark evaluation metrics. We need to develop some unified evaluation model that can assess the performance across the board for multiple datasets and environments. This will measure the generalization capabilities better.

References

Bhat, S.F., Birkl, R., Wofk, D., Wonka, P. and Müller, M. (2023). Zoedepth: Zero-shot transfer by combining relative and metric depth.

Jo, Seong-Uk, Lee, Du Yeol and Rhee, Chae Eun (2024). Occlusion-aware amodal depth estimation for enhancing 3d reconstruction from a single image, *IEEE Access* .

Jun, W., Yoo, J. and Lee, S. (2024). Synthetic data enhancement and network compression technology of monocular depth estimation for real-time autonomous driving system, *Sensors* **24**(13): 4205.

Khanal, N. and Sheshappanavar, S. V. (2024). Edadepth: Enhanced data augmentation for monocular depth estimation, *arXiv preprint arXiv:2409.06183* .

Kumar, T., Brennan, R., Mileo, A. and Bendechache, M. (2024). Image data augmentation approaches: A comprehensive survey and future directions, *IEEE Access* .

Li, Z., Bhat, S.F. and Wonka, P., (2024). Patchfusion: an end-to-end tile-based framework for high-resolution monocular metric depth estimation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10016-10025)* .

Martinho, L. A., Calvalcanti, J. M., Pio, J. L. and Oliveira, F. G. (2024). Diving into clarity: Restoring underwater images using deep learning, *Journal of Intelligent & Robotic Systems* **110**(1): 32.

Yoo, J., Jun, W. and Lee, S. (2024). Performance enhancement using data augmentation of depth estimation for autonomous driving, *2024 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, pp. 1–7.

Zhang, Juzheng, Fu, Xiao Da Terrence and Srigrarom, Sutthiphong (2024). Depth estimation in static monocular vision with stereo vision assisted deep learning approach, *2024 4th International Conference on Computer, Control and Robotics (ICCCR)*, IEEE, pp. 101–107.

Zhao, L., Chen, J., Shahzad, M., Xia, M. and Lin, H. (2024). Mfpanet: Multi-scale feature perception and aggregation network for high-resolution snow depth estimation, *Remote Sensing* **16**(12): 2087.

Zhou, Keyu, Chen, Jin, Gui, Shuangchun and Wang, Zhenkun (2024). Towards lightweight underwater depth estimation, *2024 IEEE Conference on Artificial Intelligence (CAI)*, IEEE, pp. 1442–1445.