## 

## 

**University of Toronto**

**Faculty of Applied Science & Engineering**

**APS360: Artificial Intelligence Fundamentals**

**Project Proposal**

Monocular Depth Estimation

Word Count: 1369

**Team 1**

Akram Khan

Brandon Wu

Darie Roman

Nada Al Aker

[**1.0 Introduction:**](#_hurpcuk09p5g) **3**

[**2.0 Background & Related Work:**](#_3kygzv2nzbt1) **3**

[**4.0 Architecture:**](#_61t5y2uzto6p) **4**

[4.1 Illustration](#_ngkbuzs4s12e) 4

[4.2 Architecture Description](#_tskcgwvb8aru) 4

[**5.0 Baseline Model:**](#_rrimlhcl948x) **5**

[**6.0 Ethical Considerations:**](#_oa6gyoz87ylz) **5**

[**7.0 Project Plan:**](#_vqtki67dlyh) **5**

[**8.0 Risk Register:**](#_t7zmvruuogt4) **7**

[**9.0 Colab Notebook:**](#_ok4zxrphk40e) **7**

[**10.0 References:**](#_juz43mcl2d8w) **7**

## 

## 

## 

## 

## 

## 

## 

## 1.0 Introduction:

Our project idea is to train a neural network that will generate a depth map from an RGB image. Estimating the depth from RGB images can be significantly useful in determining the geographic representations of objects in their environments [1]. Depth estimation has widespread potential in applications including robotic systems, augmented reality (AR) and autonomous vehicles due to its low-cost advantage over cost-prohibitive solutions such as LIDAR [2]. Machine learning is a reasonable approach as variations on this project have been explored in the past, and there tends to be global and local cues such as object locations and alignments that a neural network could use to generate the depth map.

## 2.0 Background & Related Work:

Based on the analysis of existing papers in this field, several proposed solutions make use of a convolutional architecture to address the problem of depth generation given an RGB image. To obtain the dataset for depth mapping on RGB images, the NYU-Depth data set is widely utilized. The NYU depth dataset consists of video sequences from several indoor scenes recorded by both RGB and depth cameras from the Microsoft Kinect [3]. One such implementation utilizes DenseNets or deeply connected convolutional layers with pretrained weights to power the encoder. In this implementation, the input RGB image is encoded into a feature vector and is successively fed into several up-sampling layers to construct the final depth map at half the input resolution. For training dataset, one implementation utilizes 50,000 images from the NYU dataset [4]. The loss function for our model will consider the difference between the ground truth map and the predicted depth map. Since the loss function for depth generation can have significant variations, one paper defined the loss as a weighted sum of three loss functions; the first loss function described the loss between the actual and predicted depth, the second loss term was defined over the gradient of the actual and predicted depth and the last term used structural similarity (SSIM) to define the loss term [5].

3.0 Data Processing:

Our training dataset consists of the NYU-Depth V2 data set, which contains images and depth maps for different indoor scenes captured at a resolution of 640 X 480 [5]. The NYU data set contains over 400,000 images with their corresponding depth maps. The labelled dataset also contains 1449 images that are accompanied with dense multi-class labels that utilize the raw depth image data and preprocess it to fill in the missing labels. Since the scope of this project only involves raw depth generation, we will clean the NYU dataset to exclude the processed image dataset. Additionally, while the NYU dataset contains images of different environments including basements, bedrooms, etc. We will modify the NYU dataset to ensure that the training dataset includes only one environment setting, such as a home office.

## 4.0 Architecture:

### 4.1 Illustration

The figure below demonstrates the architecture that corresponds to the neural network described in [section 4.2](#_tskcgwvb8aru)

### 4.2 Architecture Description

The general architecture of our neural network is a CNN, but many improvements will be made to increase performance on depth estimation. One strategy that can help create a depth map is implementing an autoencoder structure. The 3-channel input image’s height and width are decreased in the encoding portion through convolution and downsampling, and then the decoder applies deconvolution and upsampling in order to generate a 1-channel depth map with the same height and width.

An approach for convolution that has seen great success in recent years is the use of densely connected convolutional layers (DenseNet). This structure connects each convolutional layer to every layer in front of it in a feed-forward fashion, as the coloured lines in section 4.1 depict. The benefit of this method is that the network can learn which connections are important and which are not, so the architecture itself can be learned. However, the number of parameters increases rapidly with the addition of new layers, which is why transfer learning is very common with DenseNets.

One technique not illustrated above that can differentiate our architecture from state of the art models is processing of the input images with predetermined computer vision techniques such as Sobel edge detection, SIFT, and adaptive thresholding. These modifications can be passed into separate convolutional layers, and their feature maps concatenated and fed into the autoencoder. This approach takes advantage of existing patterns in the data to improve performance.

## 5.0 Baseline Model:

The simplest machine learning model that can realistically perform depth estimation would be a multi-layer CNN with consistent layer sizes. Such a network would take the RGB image as input and process it in convolutional layers that increase the number of channels but do not change the height or width. This can be achieved with kernel size 3 and padding 1, kernel size 5 and padding 2, etc. The output layer would then compress all channels to produce a single-channel depth map of the same dimensions as the input.

The hope of this structure is that each layer will learn to detect certain patterns such as wall edges, shadows, and other distinct objects, and eventually converge to an image resembling a depth map.

## 6.0 Ethical Considerations:

Solely relying on this model in high-risk situations such as driving should be avoided, as there is always a chance for error. Additionally, while the dataset we are using to train our model does not have ethical issues, the fact that the model is trained for indoor environments opens up the possibility of privacy concerns with unauthorized data collection or collection of personally identifying information.

## 7.0 Project Plan:

Team Communication:

1. Google Colab: We will split up the google colab notebook into sections that each person will be responsible for.
2. Communicate through calls and messages: We have set up a group chat specifically for this project to keep the communication limited to
3. Google shared folder with all relevant documents needed for the team to work on the project

Team Responsibilities:

1. *Project Managers (PM) and Team Leader (TL):* Nada & Akram

Ensures that the team meets internal and external deadlines, whilst managing and distributing the work amongst team members effectively. Responsible for taking down meeting minutes and setting future deadlines.

1. *First Author (FA):* Brandon

Write up a skeleton for documents that need to be submitted officially i.e initialize structure of the table of contents, references , editing completed documents, and coordinating documentation for our code.

1. *Communication Facilitator (CF):* Darie

Facilitate with resolving conflicts within the team and ensure project goals are clearly communicated to each team member. Additionally, communicate project ideas and problems encountered to the course instructors and TAs.

Table 1: Hard deadlines set by the instructors intended for our team to meet

| **Task** | **Hard Deadlines** | **Person(s) Responsible** | **Status** |
| --- | --- | --- | --- |
| Progress Report (5%) | July 12th | - |  |
| Presentation (10%) | Aug 5th | - |  |
| Report & Validation(20%) | Aug 9th | - |  |

Table 2: Internal deadlines set by the team over the course of the project

| **Task** | **Internal Deadlines** | **Person(s) Responsible** | **Status** |
| --- | --- | --- | --- |
| Collect Data | June 15th | Team |  |
| Clean Dataset | June 18th | Team |  |
| Baseline Model Design 1 | June 28th | Nada, Darie |  |
| Baseline Model Design 2 | June 28th | Brandon, Akram |  |
| Design 1 | July 7th | Nada |  |
| Design 2 | July 7th | Darie |  |
| Design 3 | July 7th | Akram |  |
| Design 4 | July 7th | Brandon |  |
| Initial draft of Progress Report | July 9th | (PM) |  |
| Final Revision of Progress Report | July 11th | Team |  |
| Initial Draft of Project Report | July 25th | (FA) |  |
| Prepare Initial Presentation | July 25th | (CF) |  |

## 8.0 Risk Register:

|  | **Risk** | **Level of Risk** | **Solution** |
| --- | --- | --- | --- |
| 1 | Teammate dropping the course | Low | Redistribute our former teammate’s work amongst ourselves evenly |
| 2 | Training takes a long time | High | Have more than one team member run the neural network with different hyperparameters simultaneously, to speed up the tuning process |
| 3 | Neural networks fail to produce accurate depth maps | Medium | We can instead focus on subproblems within the main problem, such as classifying objects in the image to determine their relative size and orientation. |
| 4 | The data contains too much noise that impedes learning | Medium | We can use a loss function that penalizes outliers less, or apply binary morphological operations to remove noise |

## 

## 9.0 Colab Notebook:

The following is the link to our group’s Google collaborate notebook that we will be using for the project: <https://colab.research.google.com/drive/1sKPJUISqbX6VuG0GkTHQkRz5yuscFLBr?usp=sharing>

## 10.0 References:

[1] D. Mwiti, “Research Guide for Depth Estimation with Deep Learning,” Medium, 11-Feb-2020. [Online]. Available: https://heartbeat.fritz.ai/research-guide-for-depth-estimation-with-deep-learning-1a02a439b834. [Accessed: 13-Jun-2020].

[2] “Depth Estimation from Monocular Image and Coarse Depth Points based on Conditional GAN.” [Online]. Available: https://www.matec-conferences.org/articles/matecconf/pdf/2018/34/matecconf\_ifcae-iot2018\_03055.pdf.

[3] “NYU Depth Dataset V2,” NYU Depth V2 " Nathan Silberman. [Online]. Available: https://cs.nyu.edu/~silberman/datasets/nyu\_depth\_v2.html. [Accessed: 15-Jun-2020].[l](https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html)

[4 ] Dwivedi, “Depth Estimation on Camera Images using DenseNets,” Medium, 05-Jun-2019. [Online]. Available: https://towardsdatascience.com/depth-estimation-on-camera-images-using-densenets-ac454caa893. [Accessed: 15-Jun-2020].

[5] “High Quality Monocular Depth Estimation via Transfer Learning.” [Online]. Available: https://arxiv.org/pdf/1812.11941v2.pdf.