**A**

**Report**

**on**

**"Monocular depth estimation"**

**Prepared by**

Lakshminaryan Shrivas (19EC071),

Bhavesh Kumar(19EC001),

Utsav Bhojani(19EC008)

**Under the guidance of** Prof. Hardik Modi

**Submitted to**

Charotar University of Science & Technology for Partial Fulfillment of the Requirements for the

Degree of Bachelor of Technology in Electronics & Communication EC456 – Summer Internship - II

of 7th Semester of B.Tech

**Submitted at**



DEPARTMENT OF ELECTRONICS & COMMUNICATION

Faculty of Technology & Engineering, CHARUSAT Chandubhai S. Patel Institute of Technology

At: Changa, Dist: Anand – 388421 July 2022





This is to certify that the report entitled “**Monocular depth estimation**” is a bonafide work carried out by **Lakshminarayan Shrivas (19EC071), Bhavesh Kumar (19EC001), Utsav Bhojani (19EC008)** under the guidance and supervision of  **Prof. Hardik Modi** for the subject **Summer Internship - II (EC456)** of 7th Semester of Bachelor of Technology in Electronics & Communication at Faculty of Technology & Engineering (C.S.P.I.T.) – CHARUSAT, Gujarat.

Under the supervision of,

|  |  |
| --- | --- |
| Prof. Hardik Modi |  |
| Designation |  |
| Department of EC Engineering, |  |
| C.S.P.I.T., CHARUSAT-Changa. |  |

Dr. Trushit K. Upadhyaya Head of Department,

Department of Electronics & Communication C.S.P.I.T., CHARUSAT- Changa, Gujarat.



**Chandubhai S Patel Institute of Technology (C.S.P.I.T.) Faculty of Technology & Engineering, CHARUSAT**

At: Changa, Ta. Petlad, Dist. Anand, Gujarat - 388421

**ABSTRACT**

Monocular depth measurement from red-green-blue images (RGB) is a well read problem with computerized detection that has been extensively investigated over the past decade using Deep Learning Methods (DL). The latest methods of measuring monocular depth are generally dependent on Convolutional Neural Networks (CNN). Measuring depth from two-dimensional images plays an important role in a variety of programs that include scene reconstruction, 3D visualization, obstacle avoidance, autonomous vehicles and private driving. This study provides a comprehensive overview of this research topic including problem representation and a brief description of traditional methods of measuring depth. Appropriate data sets and 13 advanced methods based on in-depth learning of the depth of one language ratings are reviewed, evaluated and discussed. We conclude this paper with a view of future research work that requires further investigation into the challenges of measuring monocular depth.

# ACKNOWLEDGEMENT

I would like to express my sincere thanks to my internal guide **Prof. Hardik Modi** for his valuable guidance, enthusiastic attitude and encouragement throughout the period of Summer Internship - II and Report work. His guidance, suggestion and very constructive appreciation have contributed immensely to the evolution of the Internship.

I would also like to express my sincere gratitude and sincere thanks to **Dr. Trushit Upadhyaya** HoD, Department of Electronics and Communication Engineering, Charotar University of Science and Technology, Changa for providing me the opportunity to undertake such a challenging and intellectual work.

**Lakshminarayan Shrivas (19EC071)**

**Bhavesh Kumar (19EC001)**

**Utsav Bhojani (19EC008)**

**Department of EC Engineering,**

**CSPIT, CHARUSAT**

**Table of Contents**

**Abstract i**

Contents

[ACKNOWLEDGEMENT 4](#_Toc110402016)

[Monocular depth estimation 6](#_Toc110402017)

[1.1 INTRODUCTION 6](#_Toc110402018)

[1.2 Objective 7](#_Toc110402019)

[1.3 Motivation 8](#_Toc110402020)

[Literature Review 9](#_Toc110402021)

[1.1 Datasets for depth estimation 9](#_Toc110402022)

[1.2 Deep learning methods and monocular depth estimation 12](#_Toc110402023)

[1.2.1 supervised methods 12](#_Toc110402024)

[1.2.2 Self-supervised methods 14](#_Toc110402025)

[1.2.3 Semi-supervised/ unsupervised methods 16](#_Toc110402026)

[1.3 Evaluation matrices for measuring performance of methods 17](#_Toc110402027)

[Software flow diagram 19](#_Toc110402028)

[1.1 Architecture 20](#_Toc110402029)

[Test Results and discussion 23](#_Toc110402030)

[CONCLUSION 24](#_Toc110402031)

[REFERENCES 26](#_Toc110402032)

# Monocular depth estimation

## 1.1 INTRODUCTION

* Monocular depth estimation is a widely discussed topic in the field of computer vision due to its potential strength in a wide range of applications, including [1] autonomous driving, [2] 3D reconstruction, [3] robotics, and [4,5] medical imaging.
* Due to the nature of 2d images, it is difficult to measure the depth of a scene since they do not retain temporal or spatial information. While there are systems utilizing multiple vision geometry, such as [6] stereo vision cameras, which produce stereo images, and provide fairly accurate results in general, they require proper and precise installation, calibration, and alignment [7].
* While multi-view geometry methods can produce accurate results, the downsides include their processing speed and large memory requirements; monocular depth estimation may address all of these problems.
* However, there is a lack of information regarding the necessary depth features, so it may prove to be difficult to effectively estimate depth on a monocular plane.
* In addition, there are specialized sensors that can generate multi-dimensional depth maps with high accuracy, such as [8] LIDAR based sensors, but this is an expensive and time-consuming approach because these sensors generally require high computational power and, thus, quite inadequate memory, making them less suitable.
* In recent years, the methods that are used for monocular depth estimation have improved, and the publicly accessible dataset has also enhanced their performance.
* Convolutional neural networks have shown very effective results through advancement and development.

## 1.2 Objective

* An important and frequently debated issue in computer vision is monocular depth estimation, which includes maintaining 3D information in 2D images, which is a difficult process.
* Our goal is to demonstrate one such scenario where monocular depth estimation is used in the real world, using a mono vision camera like one that may be found on a computer or laptop, known as the webcam.
* Recent developments have seen some good and accurate findings. Pretrained weights will be used in our model because, in general, training this system takes very high computing power, which may be obtained through expensive high end graphics processing units.
* As this system or model will be running on a central processing unit, it does not allow the incoming graphics to process as it should, so the decrease in frame rate is unavoidable.
* Our goal is to produce a respectable grade depth map from 2d images with a frame rate of 2-7 fps.
* We also aim to estimate the distance between the camera and the human face and to detect faces in incoming video, we also aim to measure the distance between the camera and the human face. To do this, we will provide a refrence image that shows the measured distance between the camera and the human face.
* By taking the use of this reference distance we will determine the variable distance between our face and the camera in real time.

## 1.3 Motivation

The advancement in computing power and processing power has led to the development of novel technologies which are making breakthroughs in the field of computer intelligence. Monocular depth estimation is one of those applications. Nowadays, computer vision is providing extremely helpful and accurate solutions for autonomous robots, vehicles, and aerial vehicles. This project is motivated by our desire to understand and implement one of these applications in the real world, given the speed at which research is conducted today, it seems very likely that computer vision will be the main focus of autonomous vehicles in the future. A computer vision system can be used to conduct research in areas where humans cannot reach and where human intelligence fails. Modern computer vision will have a major impact on society in the upcoming decades. Due to the fact that cost is a very critical limiting factor for developing applications that are required for a certain domain, monocular depth estimation aims to solve that problem as monovision cameras are the cheapest. Other alternatives can give reasonably accurate results like lidar-based 3D mapping but they are very expensive and require a great deal of processing power, which makes them unsuitable for applications that require a limited budget. These applications astonished us, so we decided to take on this project and understand the domain better.

# CHAPTER 1

## Image formation in camera

## The human eye will focus on the sharpest visual scene to gather information and objects that stand out in an image. Moreover, different cells inside retina will react to absorbing light in a different way, which helps our brain construct images. Image formation in camera is an interesting topic because it can make us think about how our brain perceives images as well as causes a fascinating illusion Images can be formed by the sensors in the camera or by optical devices such as mirrors, lenses, or prisms. A number of factors affect how objects are imaged. Some of these factors affect the image in two ways (from two angles). First, an image is created because light rays come into perpendicular contact with an imaginary surface inside the lens and form a real image on a screen inside or outside of the lens. Second, some cameras have components that require multiple images to be processed simultaneously.

## The process of how our eye reacts to light is called "reflection-absorption theory." The three major processes involved in this theory are

## 1) The initiation or release of light

## 2) The entry of the stimulus into the eye

## 3) The conversion of optical stimuli response.

## Reflection-absorption theory can be present in cameras, too. In other words, if your camera has these three parts, it, too is capable of implementing reflection-absorption theory. We can use a camera to measure an area and create a three dimensional model which looks like a magic trick! It’s so easy to do that some have even made their own rough versions from cardboard. The best of the best.

## Today, camera lens are merely evident and recognizable tools for capturing clear photographs in a variety of shapes and sizes. Another important operation for these lenses is to convert the collection of light waves that travel through the lens aperture and hit the camera's film or digital sensor into an image. This translates to me, as a visual English major, as rendering or bringing something into a form that is palpable; in this case, an image. Conversely, without a lens we will see nothing but black and white blobs scattered across a big canvas. While reference to any fancy dining table might be associated with flickering candles consumed by cream and wax patterns upon dark oak furniture both visible up to 3ft away.

## Focusing

## An example of this is when light enters the lens, it refracts or changes direction. These changes in direction occur because of the density and consistency of media this light has passed through.

## Exposure: Darkness or obstruction of image are captured because they decrease the amount light that enter your camera and form on your film plane.

## Squinting: occurs when you focus on an object where objects in front or behind become blurred to appear close to each other for human viewing.

## Changing focal length: when you move back from an object/ subject distance increases so that object’s size decreases, a greater depth of field is achieved. The opposite happens as one moves closer to what’s being photographed.

## Positioning Object In Camera Frame: achieved by looking at the background first. An average camera relies on optics, software and settings to form the final image we see. The first step in forming an image is light entering the camera and falling onto its sensor. This introduction gives a linear overview of how camera settings and internal filters are important for getting a good photo. It's ignored what an ISO is, but the article will touch on that eventually.

## Working

CCD is short for charge-coupled device. It is an image sensor and, like other sensors, it captures values ​​and converts them into an electrical signal. In the case of a CCD, it captures the image and converts it into an electrical signal. This CCD is actually shaped like an array or rectangular grating. It's like a matrix where each cell in the matrix contains a censor that senses the intensity of the photon. As with analog cameras, with digital cameras, when light hits an object, the light bounces back after hitting the object and is allowed to enter the camera. Each CCD array sensor is itself an analog sensor. When photons of light hit the chip, it is held as a small electrical charge in each photo sensor. The response of each sensor is directly equal to the amount of light or (photon) energy hitting the sensor surface.

|  |  |
| --- | --- |
| C:\Users\CHARUSAT\Desktop\temp\autonomous-flight\Image formation in cmaera\ccd.jpg Figure 1-ccd sensor | C:\Users\CHARUSAT\Desktop\temp\autonomous-flight\Image formation in cmaera\ccd_array.jpg Figure 2-ccd sensor |

It has a limited number of sensors, which means it can capture limited detail. Also, each sensor can only have one value against each photon particle that hits it so the number of incident photons (current) is calculated and stored.

In order to accurately measure these values, external CMOS sensors with a CCD array are also connected. In a digital camera, it's the exact opposite. The light from the thing you are photographing is brought into the camera lens. This incoming "picture" hits the image sensor chip, which breaks it up into millions of pixels.

|  |
| --- |
| C:\Users\CHARUSAT\Desktop\temp\autonomous-flight\Image formation in cmaera\ccd_sensor_array.jpg  Figure 3-ccd array |

The sensor measures the color and brightness of each pixel and stores them as a number. Your digital photo is actually an extremely long string of numbers describing the exact details of every pixel it contains.

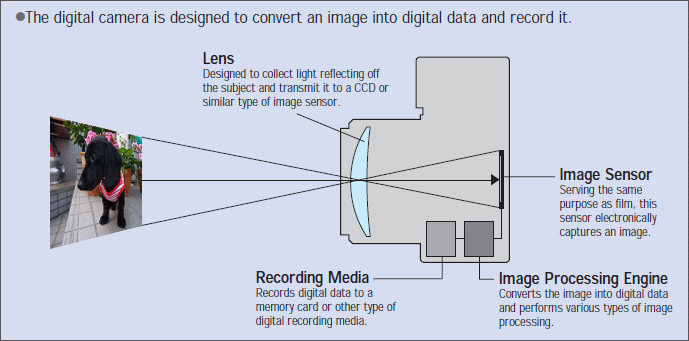


Figure 4-camera working

# Chapter 2

## Deep neural networks

Deep neural networks (DNNs) are a class of massively parameterized artificial intelligence models built on the basis of connectionist or neural theories such as neocognitron. It can create a vast number of modelling architectures for deep learning at various layers by connecting large numbers of sequential non-linear processing blocks, which may be written as feed-forward artificial neural network implementations.

One implementation is multilayer perceptron (MLP) architecture which consists of one or more fully connected layers and allows very efficient updating without redundant computations but also suffers from vanishing or exploding gradients. Mathematically, weights are applied via linear transformations in a layer's local space using matrix multiplication. The resulting activations at the input corresponding to a particular filter are computed by summing the multiply products passed upward.

Deep learning is a subfield of artificial intelligence is concerned with algorithms inspired by neuroscience.

Deep neural networks:

-were limited in that they are fully connected and not composed of multiple layers

-were slow to be updated as neurons being only capable of cell-wise updates

- operated in a very low space, requiring many layers to learn complex tasks

Deep neural network is one of the popular models used in AI for natural language processing. They particularly excel in generating meaning from language that references abstract ideas.

We introduce a deep transformer architecture and present methodologies for construction of these architectures that emphasize development of abstract features suitable to identifying semantics and meaning. Deep neural networks are a very big subject, there is not really one answer in one sentence; they're kind of complicated.

In conclusion, deep neural network consists of three parts:

A neural network is a solution to raise the machine’s recognition proficiency, learn from data and to use known information, such as statistics or certain correlations, for prediction.

These commonly are called artificial neural networks (ANN) or parallel distributed processing (PDP) networks. The term ANN was coined in 1957 and the most typical type of ANN’s architecture is called multilayer perceptron’s (MLP). It does not implement fuzzy logic, genetic algorithms, graphical models or particle swarm optimization but it does not rule them out either. The neural networks of Deep Learners are inspired by the brain and consist of many artificial neurons.

The relative times of input signals to the neurons affect their weights. The relative strength of connections influences the feelings, sensations, and other quantifiable phenomena in human beings. Neurons reproduce and transmit parameter values to others over the synapses depending on their probabilities of firing.

Deep neural networks (NNs) are a set of algorithms that allows neural networks to work together to process data. An example of this would be an input layer, two hidden layers, and an output layer. NNs can read specific complex information off images by identifying patterns that the humans cannot see.

For example, if you were getting your hair cut, the NN could detect your beard in the dim background and know you would need less trimming on that side. That is one use case of machine vision technology. While we can take advantage of this limitless processing power for many tasks, there are some negative disadvantages we must consider. First off, it has been said that once a NN is formed, it will not change its actual weight unless provided new data for it to learn from.

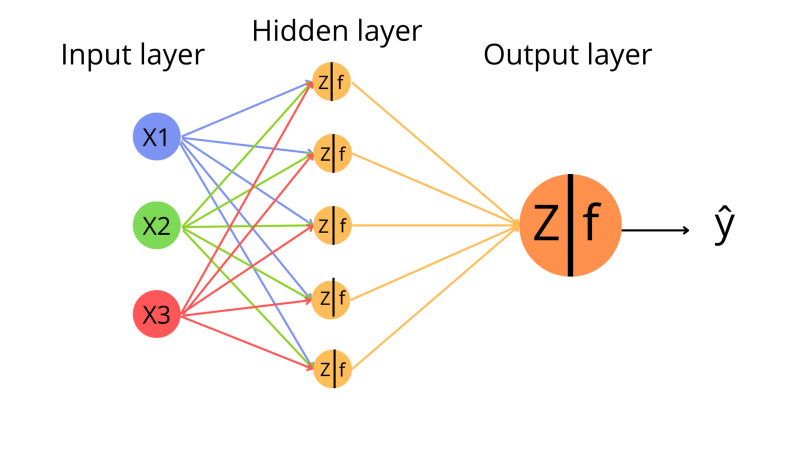
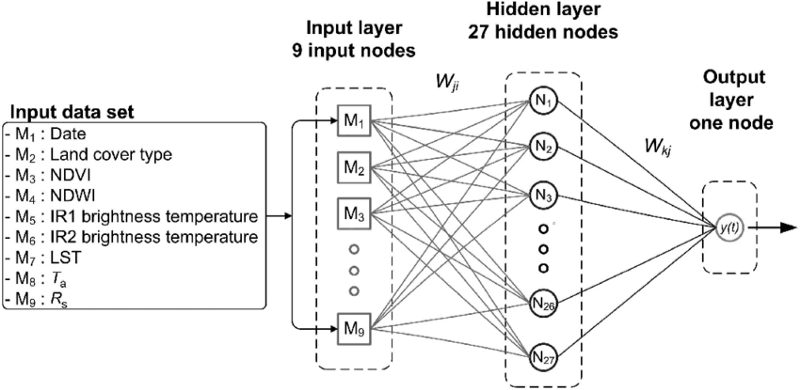


Figure 5-neural network

In the image above, you can see a single layer neural network being fed input by the input layers. Once the inputs have been passed on to a hidden layer of nodes, they are then passed on to an output layer which generates our final output ŷ. Let’s talk about each of these layers.

**Input layer:** The input layer, which includes passive neurons, is the primary layer of a neural network. They are responsible for feeding preliminary data into the system for processing. They are considered passive because they do not execute any mathematical calculations on our data. The input layer is a vector of our data with numerical values X1, X2, X3. This vector is then input into our hidden layers, which we’ll discuss next.



As seen in the preceding image, we have our input data set, which will be fed to the input layer as a vector of all numerical values from m1 to m9. Let's talk about the hidden layer.

**Hidden layer:** This is the core layer of our neural network, where all of the critical mathematical stuff takes place. I’ll attempt to explain as simply as possible. A neural network can have numerous hidden layers, but in our situation, we are only considering one. Every neuron or node in these layers runs a mathematical operation, assigning weights and biases to our input values. Let us first understand weight and bias.

**Weight:** Weight allows a neuron to focus on or prioritise an input feature. Let’s look at an example to help you understand. Assume you wish to identify a fish species; the input characteristics could include fish size, colour, wing size, fin shape, tail fin shape, and so on. Remember that the value assigned to weight is only the initial value, it will be modified as our neural network learns. Weights will be allocated to all input features, with one we wish to emphasise more will be receiving a higher weight value. This assists us in maintaining the focus or relevance of the most important input elements in our data, which will eventually have a greater influence on the neural network’s output. Let’s first understand the activation function before moving on to bias, as it will make more sense once we do.

**Bias:** Bias is basically a numerical value in general it is 1 which will be added to the sum of weighted input that is our weight multiplied to input values and then a bias which is value 1 is added to this weighted input values. Generally, it is seen that when we pass our weighted input sum to the activation function which we’ll discuss next, the activation is not flexible as we want to have it.

# Literature Review

## 1.1 Datasets for depth estimation

1. KITTI: The KITTI data set [9] is the most commonly used reference for exterior scenes taken from moving vehicles.There are two main subdivisions used to evaluate monocular depth in outdoor conditions. One of them is a training/test set with 23488 pairs of training images and 697 test images. The other are formal collisions with 42,949 training image pairs, 1000 test images and 500 test images. For the official department, the actual depth maps for the test images are maintained with test reviewers to validate the model against new data. Figure below shows some examples of the outer part of the KITTI data set.

|  |  |
| --- | --- |
| Figure 6-KITTI Dataset | Figure 7- Configuration used for KITTI data collecion |

1. Make3D: The Make3D [10] dataset is another open source dataset for deep learning-based monocular depth estimation. The Make3D dataset includes a collection of depth maps generated by a laser scanner, along with city and natural landscapes during the day. The dataset contains a total of 53 pairs of RGBD images, of which 400 pairs are used for training and 13 pairs for testing. The native resolution of the RGB image is 2272 x 170 and the resolution of the depth map is 55 x 305 pixels. FIG 3 gives you an idea for the Make3D dataset.

|  |
| --- |
| Figure 8-Samples from Make3D dataset |

1. NYUD V2: The NYU-Depth V2 [11] data set is composed of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. It contains, 1449 densely labeled pairs of aligned RGB and depth images 464 new scenes taken from 3 cities 407,024 new unlabeled frames Each object is labeled with a class and an instance number (cup1, cup2, cup3, etc) The dataset has several components: Labeled: A subset of the video data accompanied by dense multi-class labels. This data has also been preprocessed to fill in missing depth labels.

|  |  |
| --- | --- |
| Figure 9-NYUD-v2 Dataset samples with respected depth maps | Figure 10-NYUD-v2 Dataset samples with ground truth |

1. Pandora: The Pandora [12] dataset is used for head pose estimation with head localization and pose estimation on depth images. It contains nearly 250K high resolution RGB and depth images with annotation.

|  |
| --- |
| Figure 11-Pandora dataset samples |

1. SceneFlow: The SceneFlow [13] dataset introduce first large-scale datasets for training and validation of scene flow models. Their dataset contains nearly 39k stereo images with mapped disparity, depth information, optical flow and segmentation masks.

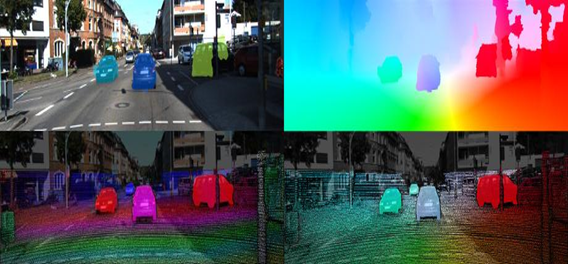


Figure 12-SceneFlow dataset samples with stereo images and respective depth information.

## 1.2 Deep learning methods and monocular depth estimation

As we have discussed earlier monocular depth estimation has became an important aspect for various applications and also there has been significant advancement in learning-based monocular depth estimation approaches and methods in some previous years like show in [1,2,3,4,5,6] papers, as we have observed most of this methods utilized convolutional neural networks which requires training with pre made datasets that contains RGB images with additional depth information. We have reviewed and categorized these methods as supervised, semi-supervised and self-supervised. Supervised learning approaches require some good quality of data with pre calculated depth information with the image to train and it can output the depth information instantly, Although the dataset required is huge and it becomes really hard and expensive to collect such dataset with accuracy.

To solve this problem of supervised learning there has been many semi-supervised learning methods developed in past years which has come to be really helpful as they only require a limited number of training images with depth information and a large number of unlabeled data for training [7,8,9]. As we know about the bias here it is a problem as semi-supervised methods are unable to update its own bias value and requires some extra information to correct it like sensory data and camera distance focal lengths.

### 1.2.1 supervised methods

Many supervised methods take in use of some sensory data such as Roas et al.[10] proposed a supervised framework which can estimate constant depth maps from Lidar points, they have used the Hilber Maps methodology [11] for generating depth maps from the Lidar scanner taking about the architecture this framework takes in use of fully convolutional residual network (FRCN) which is propose by Laina et al[56] for estimating the depth in the scene. The training for this network is done by augmenting depth images by flipping and applying color distortion, but it also limits the generation of depth maps upto 128 X 160 PX resolution where compared it to the other state of the art methods also if we look into the network it seems to be biased in the output because of the Hilbert maps densification process which may hinder in the depth.

Yuru et al [12] proposes attention based context aggregation network abbreviated as ACAN looking at the architecture this proposed algorithm take in use of the deep residual network [13] architecture and self-attention architecture as described in [14,15,16] for monitoring and varying the spatial scale and constant depth estimation to the level of pixels. Also the use of attention algorithm module it is able to create the relation between every pixel which in results in learning contextual information which is eventually lead to production and estimation of much more accurate results, In addition they used image-pooling methods for depth estimation along with soft ordinal inference translation which transforms the predicted results into constant depth information for the respective pixel which has helped in estimating more realistic depth maps. This network consumes the dataset NYU-v2[14] and KITTI[15] . They have processed this dataset by cropping or resizing the images present inside this dataset; this network has been able to produce sharp boundaries in the estimated output depth maps.

We have also learned about transfer learning and many such methods which are nowadays being implemented in many research happening worldwide one such method is proposed by Ibraheem et al[16] proposed a method which take in use of transfer learning which is an supervised method, this method is taking use of convolutional neural networks for estimating the depth information in an image they are using the typical encode-decode network architecture which is based on DenseNet-169[17] and ImageNet[18] for extracting the features out of the respective images the data from this architecture is carried to the decoder layers for estimating the respective final depth maps with sampling layers[19]. They have trained their network on densified depth images and they have augmented them by applying horizontal flipping and color distortion along with interchanging the rgb channels, the good thing about this architecture is that it produces depth maps with 320 X 240 px and is not bias unlike the one above we have discussed previously although the bilinear upsampling layer hinder in the information for all regions but it has also overcome.

Another unique approach for supervised methods for depth estimation is proposed by Fu et al.[20] which is using Spacing Increasing Discretization approach, which is utilizing the deep feature extractor multi scale feature learner with encoder and ordinal regression optimizer for high quality output depth maps this network avoids needless subsampling and captures multiple scale information in order to consume less computational resources which interns save cost and time. They have removed the subsampling layer from the pooling layers and have added dilated convolutional for getting more accurate depth information on the scene, the training of the networks is done using four datasets Make3D[21], NYU-v2[14], KITTI[15] and ScanNet[22] which in turn makes this network more versatile and accurate.

There are also some simple architecture by just taking the use of geometry like Yin et al [23] which is using simple type of geometry constraint which is defined as virtual norm and it is determined by taking any randomly sampled three points in any 3d space and this can be used to obtain a good quality depth information of the scene and the commendable thing is that this network can produce depth maps with 384 X 512 px resolution which seems to be more accurate and have strong global constraints.

Jin et al[24] proposes a new Local planar guidance layers LGPL which is inserted into the phase where we decode of the network this architecture is using decoding stage with spatial resolutions of 1/8,1/4,1/2 they are placing a layer that is guiding all the input features to the respected depth also they are using a Dense Feature Extractor, Contextual Information Extractor and the dense features for predicting the final depth of respected pixels. This architecture is using a dense Astrous Spatial Pyramid Pooling Layer[25] for estimating the depth for training; they have used the KITTI[15] and NYU-v2[14] with some cropping configuration.

A hybrid network have highlighted the issue of monocular depth estimation in videos, Zachary et al.[26] proposed a method called DeepV2D which is combining two classical algorithm which is an end to end architecture this network is containing of two modules, which is depth estimation and camera motion here the depth map module is used to take input from the camera motion and outputs the respected depth maps for training they have utilized Make3D[21], NYU-v2[14], KITTI[15] and ScanNet[22].

### 1.2.2 Self-supervised methods

Self-supervised has methods have show great improvements in monocular depth estimation with advancement in their architecture in previous years Matan et al[27] proposes a self supervised methods which uses the Siamese networks [28] they have consumed the siamese DispNet[29], ResNet[30], VGG[31] architectures using this methods they predict multi scale disparity maps which are then added with the output of previous decoder layers and the respective encoder output with using the skip connections. For training they have used the ground truth images with a resolution of 1242 X 375 pixels also it has to be noticed that this proposed network is having a way to share weights which in turn reduce on the computational operations by cutting the network size into half which can be a potential for consumer level market.

Aleotti et al[32] have proposed a network which uses end to end monocular residual matching knows as monoResMatch, this framework is also taking the use of stereo matching approach the input rgb images is being mapped with the feature space and then it is used to obtain the features which are aligned with virtual right images. The network further finds multi scale inverse depth maps which is being aligned with the input image by taking the high dimensional features into consideration, this architecture is begin constructed using the hourglass structure with skip connections this network estimate the residual corrections to the initial disparity in the final stage with using a disparity refinement module, the training of this network is done using the structural similarity reconstruction loss, disparity smoothness loss with an edge aware term and reverse Huber loss[33] they have utilized the Cityscape [34] and KITTI [15] dataset.

Guizilini et al [34] proposes a methods which predicts the depth maps by combining the geometry form the PackNet they are using the symmetrical packing and unpacking blocks for assembling the encoded and decoded information which is being done by 3d convolutions, they have consumed the architecture from [35] which contains an encode and decoder layer with skip connections which is having the geometry information. In this paper they have also introduced a new method for packing and unpacking blocks which is in turn having a visual information for much accurate depth maps predictions the training of this mode is done using the ground truth images with resolution of 640 X 192 pixels and with unlabelled data.

Andraghetti et al [36] proposes a state of the art visual odometry method for obtaining 3d points and sparse depth maps also they feed the sparse data into a auto encoder for generating more denser dept maps after getting the output form this stage they feed it to a convolutional neural network for getting the final densified depth map. The training of this network is done using ground truth depth images which consumed the KITTI[15] dataset with varied resolutions.

Clement et al[37] proposes a approach for estimating depth which utilizes a combination of three architecture and their loss function, this architecture is built based on a fully connected U-Net[38] for predicting depth and for estimating the pose between pairs of images it uses a pose network, for encoding technique they have used ResNet[30] encoder and the pre-trained ImageNet[18] as to also initialize the initial weights. This network is also utilizing appearance based loss and they have also introduced modified per pixel minimum reprojection loss, the training of this network is done using the KITTI[15] dataset with eigen split and with a resolution of 640 X 192 pixels.

### 1.2.3 Semi-supervised/ unsupervised methods

Shashan et al[39] proposes a GASDA which is a semi supervised based method for estimating the depth which takes in use of geometry aware symmetric domain adaptation, this method is targeting for training the model with synthetic data from natural images so they use symmetric style image translational by utilizing CycleGan[40] GASAD involves form real and unreal images to unreal to real images together with translations from epipolar geometry of the real stereo images they have trained this network image style translation and symmetric depth estimation with pixel resolution of 192 X 640.

[41] propose a dual CNN based model for generating disparity maps with 6 losses, they also extend the model with 12 losses by utilizing the cross disparities. This models DNM6 and DNM12 are experimented over the KITTI driving and cityscapes dataset.The DNM6 model demonstrated two cnn layers one for each left and right images in stereo vision, this model is based on auto-encoder algorithm which is taken from [2] Gordan et al. In DNM12 architecture left and right disparity is independently predicted, four bilinear samplers is being used for constructing two images from left cnn and two images from right cnn the DNM12 model outperforms the [2]left-right consistency method.

[42] Proposes unsupervised learning framework for monocular depth and camera motion estimation from unlabeled video sequences. Along with recent work [43, 44, 45], an end-to-end learning approach with view synthesis as the supervisory signal is used. In contrast to the previous work, this method is completely unsupervised, requiring only monocular video sequences for training. This method uses single-view depth and multiview pose networks, with a loss. The loss is based on warping nearby views with the target using computed depth and pose.

The networks are then merged with the loss during training, and it can be applied independently during test time.System is trained on the split provided by [46], all the frames from the testing scenes and static sequences are excluded with MOF (mean optical flow) magnitude less than 1 pixel for training.The length of image sequences are fixed to be 3 frames, the central frame is treated as the target view and the ±1 frames as the source views.

The length of input image sequences to their system is fixed to 5 frames.

The length of input image sequences to their system is fixed to 5 frames. To resolve scale ambiguity during evaluation, first the scaling factor for the predictions made by ORB-SLAM [47] (full), ORB-SLAM (short) and dataset mean of car motion to best align with the ground truth is optimized, and then the Absolute Trajectory Error (ATE) [48] is measured as the metric. ATE is computed on 5-frame snippets and average is taken over the full sequence.

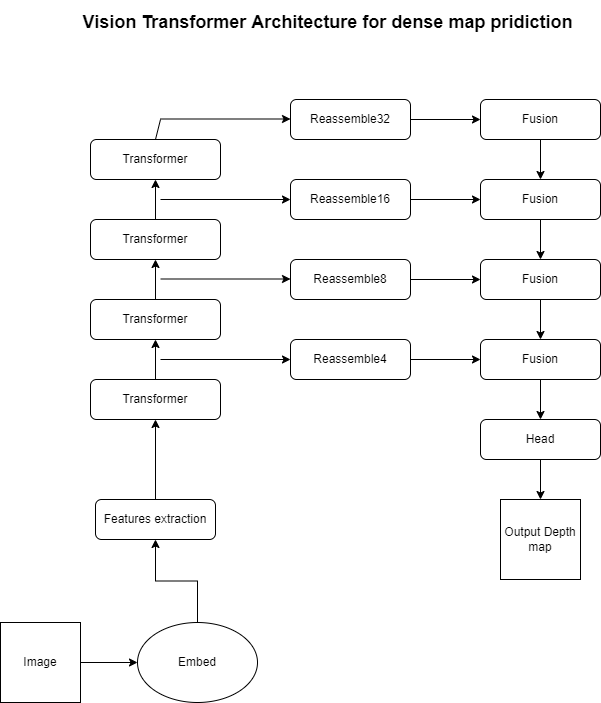
[49] proposes a training framework that trains a neural network without ground truth images, they take in use of epipolar geometry constraints. By image reconstruction loss they generate disparity images, and contributes a end to end unsupervised monocular depth estimation with a novel training loss.Their fully connected architecture is based on dispnet[50] with several modifications, their defined losses are Appearance Matching Loss, Disparity Smoothness Loss.

## 1.3 Evaluation matrices for measuring performance of methods

Here, we are listing some of the common matrices which were used for evaluating the performance fo the above methods and architecture which is also useful in comparing the models with each other to rank the performance.

We can describe this matrices as Absolute Relative Difference, Root Mean Square Error, Root Mean Square Error log and Square Relative Error or SqRel, for comparing all the methods there are publicaly available pre-trained models available which we can take in use now by using the pre-trained dataset we can actually measure a more generalised performance of our architectures on different testing sets.

# Software flow diagram



## 1.1 Architecture

As seen in the architecture of the vision transformer above, we use the embedding layer for creating the images patches or the tokens which our transformer maintain throughout the computation process. As we know in ViT this tokens maintain one to one relationship with other image patches so it means it can also maintain the spatial resolution in the embedding layer throughout the transformer steps. This also aims to maintaining the initial receptive field in embedding, this lets to increase overall receptive field as features pass from consecutive sampling and convolutional layers.

As the architecture shows the input image is converted into tokens, either by using a ResNet-50 feature extractor or by extracting patches that do not overlap which is then been flattened. In the embedding process the image is augmented with positional embedding and patch-independent readout token then added to it.

After the initial embedding and tokenization the image is then passed through couple of transformer stages, as we pass the tokens from each transformer stages we also input them to the respective reassemble. This ressembler read the tokens from the transformers and reassemble the image in multiple different resolution it forms and image like representation of each tokens received from the transformers.

The fusion model as seen from the software flow diagram above takes input from the reassembler then they gradually sample and fuse the image like representation received from reassembler to create an accurate prediction considering all the features fine-grained.

as the embedding of the images proceeds they are then flattened into vectors with linear projection, as we have discussed above we can use ResNet50 for the image and with pixel features of the resulting features map as token for the transformer network. we know as transformer works as set-to-set functions, they do not conserve the spatial position information in individual tokens. So we concatenated the embedded images with a position embedding for preserving this information, further as nlp the ViT add a token that is different from input image and that token we take in use for classification and we call this token as readout token.

This network is basically inspired by the work presented in [1] of which they represent several variants this network consumes the ViT-Base in this they use patch-based embedding technique and features with 12 transformer subsequent layers, similarly their other work ViT-large which uses the same configuration as above but it has 24 transformer network layer and a wider feature size and one more configuration which use ResNet50 architecture for computing embedding of image with same 12 transformer configuration. Mainly if I talk about this architecture it use patch size of 16.For ViT base and large network the flattened patches upto the dimensitons 768 and 1024 notice the dimensions of this features are larger than the pixels in an input patch, this way the embedding technique can learn to preserve if it's important for the consideration.

As decoder are equally important this architecture's decoder combines the set of tokens and make a image-like representations with different resolutions. This features are then fed into the final dense or the fusion model prediction. This proposed reassembler is a three stage operation for restoring image like representation from the tokens received by the layers of transformers. we can show this relations as an equation,

Reassembeler(t) = (Resample concatenate Read)(t) ….. eq.1

In reassembler we mainly consider the output size ration for the representation which we have gathered with respect to the input image and the dimensions of the output features, as we have discussed above the tokes lets say N the set of tokens we have to map them with N + 1 this has the ability to accept spatial concatenation into the representation. This process is equally important from the view of readout token as discussed earlier, as we see the the readout token quiet don't make sense or states the purpose for the prediction task but still i can be useful to accumulate and disperse global information.

specific to this architecture it uses three types of mapping for the read out token which is stated below this simply ignores the readout token,

Read(ignore)(t) = {t1,...,tn}…eq.2

Read(add)(t) = {t1 + t0,..., tnp+ t0}

this configuration carry the information from readout token upto all available tokens in the representation, and the last one,

Read(proj)(t) = {mlp(cat(t1,t0)),...,mlp(cat(tnp,to))}

this configuration carries information by concatenating out readout token to all the other tokens before projecting the representation to the original features dimensions.As we have understood about the readout tokens now after the read block the resulting Np tokens will be reassembled and reshaped into the image like representation by considering the initial patch position in the image, basically spatial concatenation operating is applied which results into a feature map of size (H X W)/p2. As we receive otu representation which we will pass to a spatial resampling layer that transforms the representation to size (H X W)/s2, so basically this is first implemented using 1 x 1 convolutional to project the representation of the input with feature map which is then followed by the strided 3 X 3 convolution.

We talked about the resolution earlier so we reassemble their features in four different stages which results into four different resolution of our image representation, the features received from the transformer which are at deep layers are assembled at relatively lower resolutions and the one from the earlier layer are assembled at higher resolution, as we know and have talked about the architecture of this three different ViT transformers specifically this configuration with ViT-large with 24 transformer layers reassemble at layer {5,12,18,24} while with the ViT-Base architecture with 12 transformer assembles at layers {3,6,9,12}.

After all of the above processes it combines the extracted feature mapors form the respective stages with taking use of RefineNet based feature architecture and upsampling the image representation with some factor in each fusion layer. So we get our final representation size which is almost half in the size fo the resolution of the initial input image then we connect a task oriented head in the output for producing the final prediction.

# Test Results and discussion

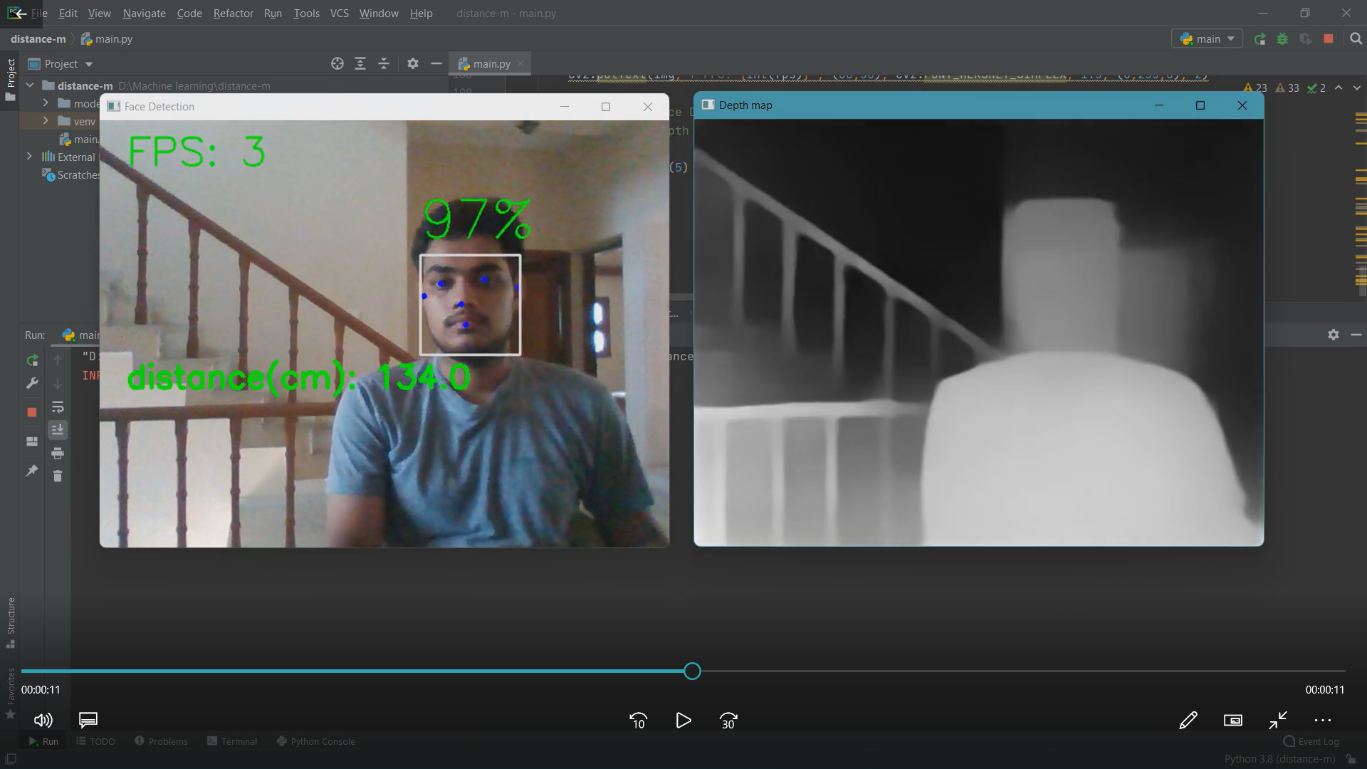
### 

We have aimed to predict depth man form a single frame or the sequence of frame which is a video from a monovision camera as our goal and we have been successfully executed the program for that we have tested the model indoor and outdoor but it seems that indoor results seems to be more accurate and understandable from a viewpoint of a laymen so we will be attaching the image of indoor view also we have implemented another method where we are also calculating the distance of my or a human face from the camera by taking in the use of a simple geometry formula which then quite accurately predicts the distance from the camera and the face as we have measured using a measuring tape in real time and we can say that there is a some fluctuation but overall we are satisfied with the test results. Our model uses pre-trained weights for estimating the depth from single image as training a deep neural network which is in research domain right now with all that complexity it becomes quite hard to train and implement also for training purpose we require to download hundreds of gigabyte of data which is not in out capacity and then we require a high end graphics processor for training out neural network which then can predict accurately so taking all this things in account we have decided to use pre-trained weights which is more easy to embed and work with and we have also gone through out the network to understand each and every aspect of it. We had concerns about the weights which we have received for running the model on our machine but it went well and to be precise we have used as smaller version of the architecture for running on our machine as the larger one was causing problems in the system, although the larger one may give more precise depth maps but we were limited by the computing power so we have chosen the small version. They have updated their models so now new models are also available which can also be used the model which is being used in this project works on convolutional nns with some embedding of the transformer networks which is making it more accurate but the latest they have transformed the whole network which is now based on transformers.

We are listing the comparison image side by side we have plotted while running our model

The image in the left windows show the real video that is being captured from the monovision camera or the web cam which is embedded on my laptop and also distance of my face from the camera can be seen in real time and it increase or decrease in real time based on the location of the human face, right side is the depth map which is being generated from the left side image we have aimed to get more then 8 to 10 frames per second but due to computational limitation we were able to only reach max 5 frames per second.

Now observing the depth image the pixels which are actually near the camera are lighted up with a light color and the picture which are actually for away from the camera are lighting up with a dark color so our model is trying to predict the distance of each pixel from the camera we could have used different colors for lighting up the pixel but as using white and dark color uses less storage and less computational power approved to use them.



# CONCLUSION

An Through out this project we have learned and understood that convolution neural networks have the ability to predict depth information from a single image which is really helping us in understanding the scene or the environment more accurately which is eventually gonna lead to development of certain breakthrough technologies which will be helping and advancing the human life for betterment, we have seen the implementation of this technologies in various sectors and analyzed the overall impact that will we created if applied in certain applications like autonomous driving vehicles or the military drone which can save lives by exploring the unknown area where humans cannot go or reach by giving the ability of depth to a computer machine we are making it more advanced like humans although the algorithms are not that advanced or may not be that fast and efficient but in future research work it will be more rich and faster in computing the depth.

Although the only issues with using complex deep neural network is their memory requirements which can a expensive problem while we deal with high-resolution images and when we aim to predict depth maps with high resolution, same goes with the computational power however if we can develop some lighter deep neural network architectures it will be much more efficient and less time consuming will be less expensive and can be developed at consumer level market.

Another challenge that remains is for achieving higher accuracy which can be affected by the type of environment like fog, occlusions, highly cluttered scenes and some material properties of some objects, also another problem is with the datasets as deep learning neural networks heavily rely on ground truth images which can be sometimes really expensive to collect in the real world. We aim to see the development of many more such datasets and also self-adoption methods which can adapt themselves in accordance with the scene conditions

# REFERENCES

1] F. Xue, G. Zhuo, Z. Huang, W. Fu, Z. Wu and M. H. Ang, "Toward Hierarchical Self-Supervised Monocular Absolute Depth Estimation for Autonomous Driving Applications," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 2330-2337, doi: 10.1109/IROS45743.2020.9340802.

2]  Chen L., Tang W., John N.W., Wan T.R., Zhang J.J. Augmented Reality for Depth Cues in Monocular Minimally Invasive Surgery. arXiv Prepr.

3] Liu X., Sinha A., Ishii M., Hager G.D., Reiter A., Taylor R.H., Unberath M. Dense Depth Estimation in Monocular Endoscopy with Self-supervised Learning Methods. IEEE Trans. Med. Imaging. 2019

4] Laidlow T., Czarnowski J., Leutenegger S. DeepFusion: Real-Time Dense 3D Reconstruction for Monocular SLAM using Single-View Depth and Gradient Predictions; Proceedings of the 2019 International Conference on Robotics and Automation (ICRA); Montreal, QC, Canada

5] Palafox P.R., Betz J., Nobis F., Riedl K., Lienkamp M. SemanticDepth: Fusing Semantic Segmentation and Monocular Depth Estimation for Enabling Autonomous Driving in Roads without Lane Lines. Sensors

6] Schönberger J.L., Frahm J.M. Structure-from-Motion Revisited; Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Las Vegas, NV, USA. 27–30 June 2016

7] Yu F., Gallup D. 3D Reconstruction from Accidental Motion; Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition; Columbus, OH, USA. 23–28 June 2014

8] N. . -A. . -M. Mai, P. Duthon, L. Khoudour, A. Crouzil and S. A. Velastin, "Sparse LiDAR and Stereo Fusion (SLS-Fusion) for Depth Estimation and 3D Object Detection," 11th International Conference of Pattern Recognition Systems (ICPRS 2021), 2021, pp. 150-156, doi: 10.1049/icp.2021.1442.

[9] Geiger A., Lenz P., Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite; Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition; Providence, RI, USA. 16–21 June 2012; pp. 3354–3361.

[10]. Saxena A., Sun M., Ng A.Y. Make3d: Learning 3d scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. 2008;31:824–840.doi: 10.1109/TPAMI.2008.132.

[11]. Silberman N., Hoiem D., Kohli P., Fergus R. Indoor segmentation and support inference from rgbd images; Proceedings of the European Conference on Computer Vision; Florence, Italy. 7–13 October 2012; pp. 746–760.

12] Borghi G., Venturelli M., Vezzani R., Cucchiara R. Poseidon: Face-from-depth for driver pose estimation; Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, HI, USA. 21–26 July 2017;

13] Mayer N., Ilg E., Hausser P., Fischer P., Cremers D., Dosovitskiy A., Brox T. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation; Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Las Vegas, NV, USA. 27–30 June 2016;

Reference from Literature review,

1] Ji R., Li K., Wang Y., Sun X., Guo F., Guo X., Wu Y., Huang F., Luo J. Semi-Supervised Adversarial Monocular Depth Estimation. IEEE Trans. Pattern Anal. Mach. Intell. 2019;

2] Tosi F., Aleotti F., Poggi M., Mattoccia S. Learning monocular depth estimation infusing traditional stereo knowledge; Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition; Long Beach, CA, USA. 16–20 June 2019;

3] Riegler G., Liao Y., Donne S., Koltun V., Geiger A. Connecting the Dots: Learning Representations for Active Monocular Depth Estimation; Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition; Long Beach, CA, USA. 16–20 June 2019;

4] Ramamonjisoa M., Lepetit V. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation; Proceedings of the 2019 IEEE International Conference on Computer Vision; Seoul, Korea. 27–28 October 2019.

5] Pillai S., Ambruş R., Gaidon A. Superdepth: Self-supervised, super-resolved monocular depth estimation; Proceedings of the 2019 International Conference on Robotics and Automation (ICRA); Montreal, QC, Canada. 20–24 May 2019; pp. 9250–9256.

6] Godard C., Mac Aodha O., Firman M., Brostow G.J. Digging into self-supervised monocular depth estimation; Proceedings of the 2019 IEEE International Conference on Computer Vision; Seoul, Korea. 27–28 October 2019; pp. 3828–3838.

7] Kuznietsov Y., Stückler J., Leibe B. Semi-Supervised Deep Learning for Monocular Depth Map Prediction; Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, HI, USA. 21–26 July 2017; pp. 2215–2223.

8] Chen Y., Zhao H., Hu Z. Attention-based context aggregation network for monocular depth estimation. arXiv Prepr.

9] Alhashim I., Wonka P. High quality monocular depth estimation via transfer learning. arXiv Prepr.

10] Dos Santos Rosa N., Guizilini V., Grassi V. Sparse-to-Continuous: Enhancing Monocular Depth Estimation using Occupancy Maps; Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR); Belo Horizonte, Brazil. 2–6 December 2019;

11] Ramos F., Ott L. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. Int. J. Rob.

12] Laina I., Rupprecht C., Belagiannis V., Tombari F., Navab N. Deeper Depth Prediction with Fully Convolutional Residual Networks; Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV); Stanford, CA, USA. 25–28 October 2016;

13] Lee J.H., Han M.-K., Ko D.W., Suh I.H. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv Prepr.

14] Silberman N., Hoiem D., Kohli P., Fergus R. Indoor segmentation and support inference from rgbd images; Proceedings of the European Conference on Computer Vision; Florence, Italy. 7–13 October 2012;

15] Geiger A., Lenz P., Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite; Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition; Providence, RI, USA. 16–21 June 2012;

16] Teed Z., Deng J. Deepv2d: Video to depth with differentiable structure from motion. arXiv Prepr.

17] Huang G., Liu Z., Van Der M.L., Weinberger K.Q. Densely Connected Convolutional Networks; Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, HI, USA. 21–26 July 2017;

18] Deng J., Dong W., Socher R., Li L.-J., Li K., Li F.-F. ImageNet: A large-scale hierarchical image database; Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009); Miami, FL, USA. 20–25 June 2009.

19] Lehtinen J., Munkberg J., Hasselgren J., Laine S., Karras T., Aittala M., Aila T. Noise2noise: Learning image restoration without clean data. arXiv Prepr.

20] Fu H., Gong M., Wang C., Batmanghelich K., Tao D. Deep ordinal regression network for monocular depth estimation; Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018; Salt Lake City, UT, USA. 18–22 June 2018;

21] Saxena A., Sun M., Ng A.Y. Make3d: Learning 3d scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. 2008;31:824–840.

22] Dai A., Chang A.X., Savva M., Halber M., Funkhouser T., Nießner M. Scannet: Richly-annotated 3d reconstructions of indoor scenes; Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, HI, USA. 21–26 July 2017;

23] Goldman M., Hassner T., Avidan S. Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation; Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition; Long Beach, CA, USA. 16–20 June 2019.

24] Guizilini V., Ambrus R., Pillai S., Gaidon A. Packnet-sfm: 3d packing for self-supervised monocular depth estimation. arXiv Prepr.

25] Chen L.-C., Papandreou G., Schroff F., Adam H. Rethinking atrous convolution for semantic image segmentation. arXiv Prepr.

26] Andraghetti L., Myriokefalitakis P., Dovesi P.L., Luque B., Poggi M., Pieropan A., Mattoccia S. Enhancing self-supervised monocular depth estimation with traditional visual odometry; Proceedings of the 2019 International Conference on 3D Vision (3DV); Quebec City, QC, Canada. 16–19 September 2019;

27] Zhao S., Fu H., Gong M., Tao D. Geometry-aware symmetric domain adaptation for monocular depth estimation; Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition; Long Beach, CA, USA. 16–20 June 2019;

28] Bertinetto L., Valmadre J., Henriques J.F., Vedaldi A., Torr P.H.S. Fully-convolutional siamese networks for object tracking; Proceedings of the European Conference on Computer Vision; Amsterdam, The Netherlands. 11–14 October 2016;

29] Mayer N., Ilg E., Hausser P., Fischer P., Cremers D., Dosovitskiy A., Brox T. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation; Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Las Vegas, NV, USA. 27–30 June 2016;

30] He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition; Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016; Las Vegas, NV, USA. 27–30 June 2016;

31] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv Prepr.

32] Tosi F., Aleotti F., Poggi M., Mattoccia S. Learning monocular depth estimation infusing traditional stereo knowledge; Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition; Long Beach, CA, USA. 16–20 June 2019;

33] Huber P.J. Robust Estimation of a Location Parameter. In: Kotz S., Johnson N.L., editors. Breakthroughs in Statistics: Methodology and Distribution. Springer New York; New York, NY, USA: 1992.

34] Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B. The cityscapes dataset for semantic urban scene understanding; Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016; Las Vegas, NV, USA. 27–30 June 2016;

35] Chen W., Fu Z., Yang D., Deng J. Single-image depth perception in the wild; Proceedings of the Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016; Barcelona, Spain. 5–10 December 2016;

36] Zhou T., Brown M., Snavely N., Lowe D.G. Unsupervised learning of depth and ego-motion from video; Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, HI, USA. 21–26 July 2017;

37] Xie J., Girshick R., Farhadi A. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks; Proceedings of the European Conference on Computer Vision; Amsterdam, The Netherlands.

38] Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W.M., Frangi A.F., editors. Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III. Springer International Publishing; Cham, Switzerland: 2015.

39] Garg R., BG V.K., Carneiro G., Reid I. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In: Leibe B., Matas J., Sebe N., Welling M., editors. Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII. Springer International Publishing; Cham, Switzerland: 2016.

40] Zhu J.-Y., Park T., Isola P., Efros A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks; Proceedings of the 2017 IEEE International Conference on Computer Vision; Venice, Italy. 22–29 October 2017.

47] Zhou, Tinghui, et al. "Unsupervised learning of depth and ego-motion from video." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

48] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. DeepStereo: Learning to predict new views from the world’s imagery. In Computer Vision and Pattern Recognition, 2016.

49] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised

CNN for single view depth estimation: Geometry to the rescue. In European Conf. Computer Vision, 2016.

50] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with lef-right consistency. In Computer Vision and Pattern Recognition, 2017.