KLE Society's
KLE Technological University

A Mini Project Report

On

Image Depth Estimation

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

In

Computer Science and Engineering

Submitted By

Shreya B Devagiri	01FE20BCS019
Rohit Devaranavadagi	01FE20BCS246
Sneha K Pamali	01FE20BCS035
Nishant R Negali	01FE20BCS204

Under the guidance of
Dr.Meena S

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
HUBLI–580 031 (India).

Academic year 2022-23
KLE Society's

KLE Technological University
2022 - 2023

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Mini Project entitled “Image Depth Estimation” is a bonafide work carried out by the student team Ms. Shreya B Devagiri 01FE20BCS019, Mr. Rohit Devaranavadagi 01FE20BCS0246, Ms. Sneha K Pamali 01FE20BCS035, Mr. Nishanth R Negali 01FE20BCS204, in partial fulfillment of completion of Fifth semester B. E. in Computer Science and Engineering during the year 2022 – 2023. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said programme.

Guide	**Head, SoCSE**
Dr. Meena S. M	**Dr. Meena S. M**

External Viva:

Name of the Examiners	**Signature with date**
1.	
2.	

ABSTRACT

Each element in the depth map correlates to the depth of the object at a specific pixel in the image, and it is a 2D representation of the depth of objects in a scene or environment. Researchers have created methods for picture depth estimation that use various modalities, including stereo images or LiDAR point clouds, to increase accuracy. These methods can benefit from the supplementary data offered by various modalities to more accurately estimate the depths of objects in the picture. In this study, we used transformers to create a new state of the art in picture depth map estimation. We first identify the object on the input image then apply an instance to the object identified. The instance segmented output is then fed into the depth map-generating transformers. In order to account for the variations in how objects look in photos due to changes in their position and orientation in relation to the camera, as well as for occlusions, reflections, transparent objects, low light images, and image aspect ratios, depth maps are created. The generation of the depth map was flawed, but we were able to fix it by using transformers, instance segmentation, and object detection. Our method has the advantage of accurately generating a depth map even when there are numerous items in a single image and objects that are far away from the camera. Even with poor image quality, severe camera angles or distortion, and a variety of depth and scene kinds, the production of depth maps is successful. We used the 85.67% accurate KITTI Dataset to train our model.

The emergence of new applications and technologies that rely on depth information, as well as continual efforts to improve the accuracy and efficiency of algorithms, make picture depth estimation an active area of research overall. Image depth estimation is anticipated to remain a significant issue in the field of computer vision in the future, having the potential to have an impact on a variety of applications and technologies.

In the upcoming years, picture depth prediction's application space is anticipated to expand and change as researchers and developers work to take advantage of this technology's power to address a variety of issues in industries like robotics, augmented reality, and computer vision.

ACKNOWLEDGEMENTS

The happiness and elation that accompany the successful completion of any task would be insufficient if we did not acknowledge those persons whose professional counsel and encouragement assisted us in successfully completing this report work.

We take this opportunity to thank Dr. Ashok Shettar (Vice-Chancellor, KLE Technological University, Hubli) and Dr. Prakash Tewari (Dean of Academic Affairs, KLE Technological University, Hubli).

We also take this opportunity to thank Dr.Meena S. M who is our guide and Uday Kulkarni for having provided us with an academic environment that nurtured our practical skills contributing to the success of our project.

We sincerely thank Mr.Mahesh Patil, Mini Project Coordinator for their support, inspiration and wholehearted cooperation during the course of completion.

Shreya B Devagiri	01FE20BCS019
Rohit Devaranavadagi	01FE20BCS246
Sneha K Pamali	01FE20BCS035
Nishanth R Negali	01FE20BCS204

1. INTRODUCTION

1.1 Preamble

Determining the distance between objects in a scene or image and a camera or observer is a process known as depth estimation. Numerous industries, including robotics, augmented and virtual reality, and autonomous vehicles can benefit from this vital computer vision task.

Numerous techniques, such as stereoscopic vision, structured light, and laser range, can be used to estimate depth. All of these techniques triangulate the distances of objects in the scene by taking several photographs or measurements from various angles. The principles and methods used to determine depth differ.

The quality of the pictures or measurements, the size and distance of the objects, and the characteristics of the surroundings are only a few of the variables that affect how accurate and reliable depth estimation is. Occlusions, reflections, and the presence of surfaces without textures or distinguishing features are some of the additional difficulties and restrictions that depth estimation must overcome.

Overall, depth estimation is an essential computer vision task with a wide range of real-world applications and is still a prominent topic for research in both academia and industry.

1.2 Motivation

The method of estimating the distance or depth of objects in a scene from a certain point of reference, like a camera, is termed as depth estimation. There are a number of reasons to determine the depth of the a scene, including:

3D reconstruction: Depth estimate can be used to recreate a scene's three-dimensional (3D) representation, which is beneficial for a variety of applications, namely virtual reality, robotics, and computer-aided design.

Augmented reality: Depth estimation can be used to develop augmented reality (AR) applications, which overlay virtual objects on the actual world in a realistic manner. To ensure that the virtual objects are precisely aligned with the scene in reality, this calls for precise depth estimation.

Object recognition: Depth information allows the system to differentiate between things that are entirely or partially covered up by other objects, thus facilitating object detection and segmentation in a scene.

Scene understanding: Since depth estimation enables the machine to recognise the relative positions and separations of items in the scene, it can also be used to optimize the overall scene interpretation.

Robotics: As depth estimation enables the robot to maneuver around and interface with elements in its vicinity, it might be helpful in robotics applications.

1.3 Objectives of the project

- To implement and compare among various methods and algorithms for Object detection, Segmentation and Generation of Depth Map.
- To perform object detection followed by instance segmentation on a given image for depth estimation.
- To generate a depth map for the segmented image for depth prediction.
- To produce better results than already existing state of the art algorithms and models.

1.4 Literature Survey

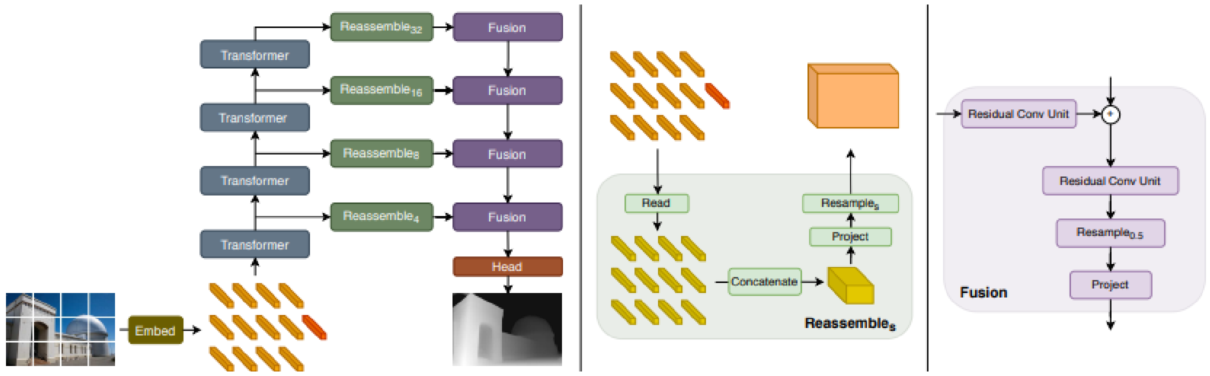
In this section we will be discussing the various algorithms and architectures that are currently used in our problem domain. The study is done on a common dataset to better understand the impact of the algorithms on our domain. The different approaches that we came across have been discussed here.

1.4.1 Using Vision Transformers for dense prediction

It is an architecture where the foundation for dense prediction tasks is provided by vision transformers as opposed to convolutional neural networks. We basically create image-like representations of tokens from different stages of the vision transformer at varying resolutions, then finally integrate them into full-resolution predictions using a convolutional decoder.

The transformer backbone has a global receptive field at each and every stage and it generates representations at a constant, comparatively high resolution. When compared to fully convolutional networks, the dense prediction transformer can offer predictions that are more precisely granular and globally coherent.

This architecture yields better results compared to the traditional results of the convolutional neural networks. For the task of monocular image depth estimation where we are given abundant training data (KITTI dataset), the DPT increases its performance by almost 28% when compared to the traditional top-performing fully-conventional network.

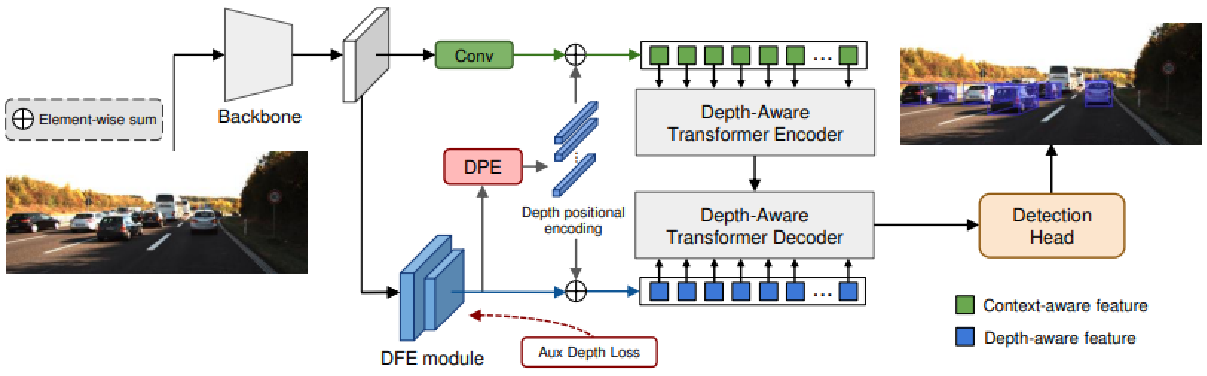


The basic architecture of the DPT is shown above. The input image is embedded into encoders, reassembled and then fused back to get the depth map (image).

1.4.2 3D Object Detection with Depth-aware Transformer

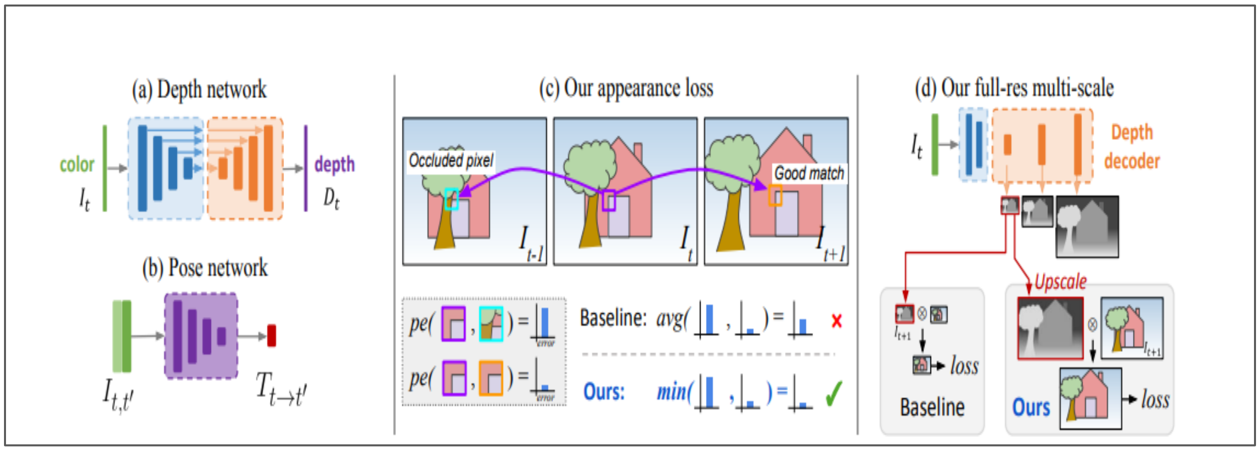
It's critical to remember that 3D object detection is a challenging operation that calls for particular methods and procedures in order to interpret and evaluate the 3D data. Although depth-aware transformers and other NLP-based methods could be able to improve the performance of a 3D object recognition system, they are not likely to serve as the main method of identifying and categorizing items in the scene. Instead, they could be utilized as a component of a bigger pipeline that integrates a variety of methods and strategies to provide the greatest results.

The author of this paper introduces MonoDTR which is an end-to-end depth-aware transformer.Depth-aware Feature Enhancement (DFE) and Depth-aware Transformer Module make up the majority of its two components (DTR). These proposed depth-aware modules can be plugged into existing 3D object detectors to improve the performance.



Encoder and Decoder layers are used in this architecture as well. DFE and DTR modules are embedded at various layers in the architecture.

1.4.3 Digging into Self-Supervised Monocular Depth Estimation



In this paper, the author shows how a surprisingly simple model and related design decisions result in better predictions. The author has proved this by introducing a better model with a minimal reprojection loss with effective occlusion handling, a reduced full-resolution multi-scale sampling technique and also an auto-masking loss to ignore training pixels that goes against the presumption of camera motion. This model can be trained using monocular video data, stereo data, or mixed monocular as well as stereo data.

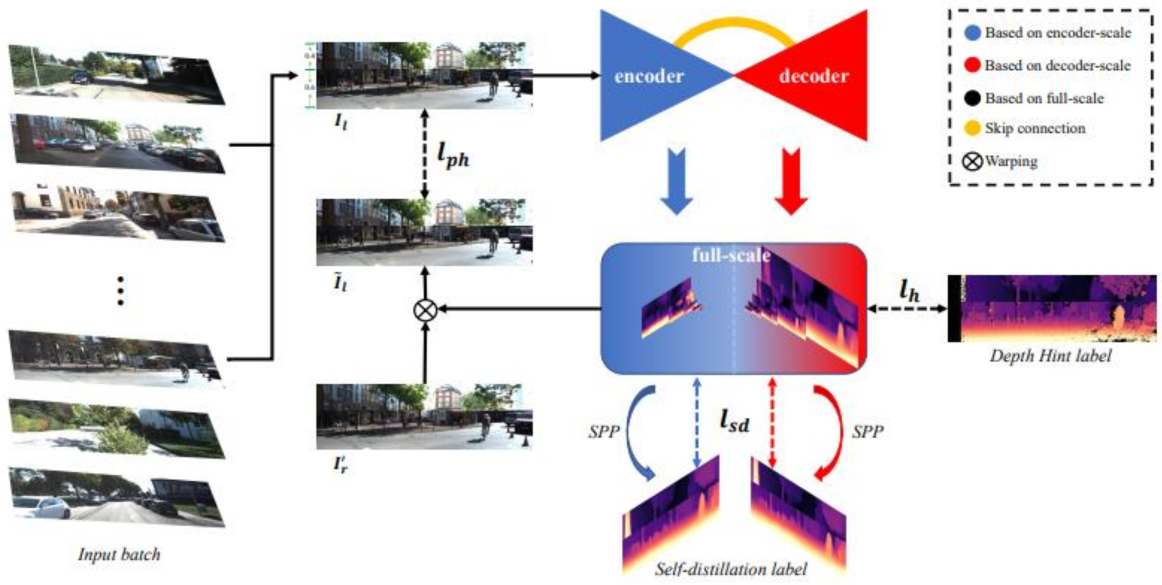
This model solved the challenge of acquiring per-pixel ground truth, all together this work suggested a lot of improvements which, when combined, produce depth maps that are superior to competing self-supervised approaches in terms of both quality and quantity.

1.4.4 Self-supervised Monocular Depth Estimation

In order to compete with many supervised methods for monocular depth estimation, self-supervised methods play a vital role. Here, in this work, to extract the full capacity of self-supervised monocular depth estimation method, an EPCDepth model is proposed by the author which helps to to close the gap with other state of the art supervised methods.

Data grafting is being introduced here, an innovative method of enhancing the data that challenges the model to look beyond the vertical image position for depth clues.

And also, a self-distillation loss that is exploratory in nature and is suggested by the self-distillation label produced by the novel post-processing technique, selective post-processing.



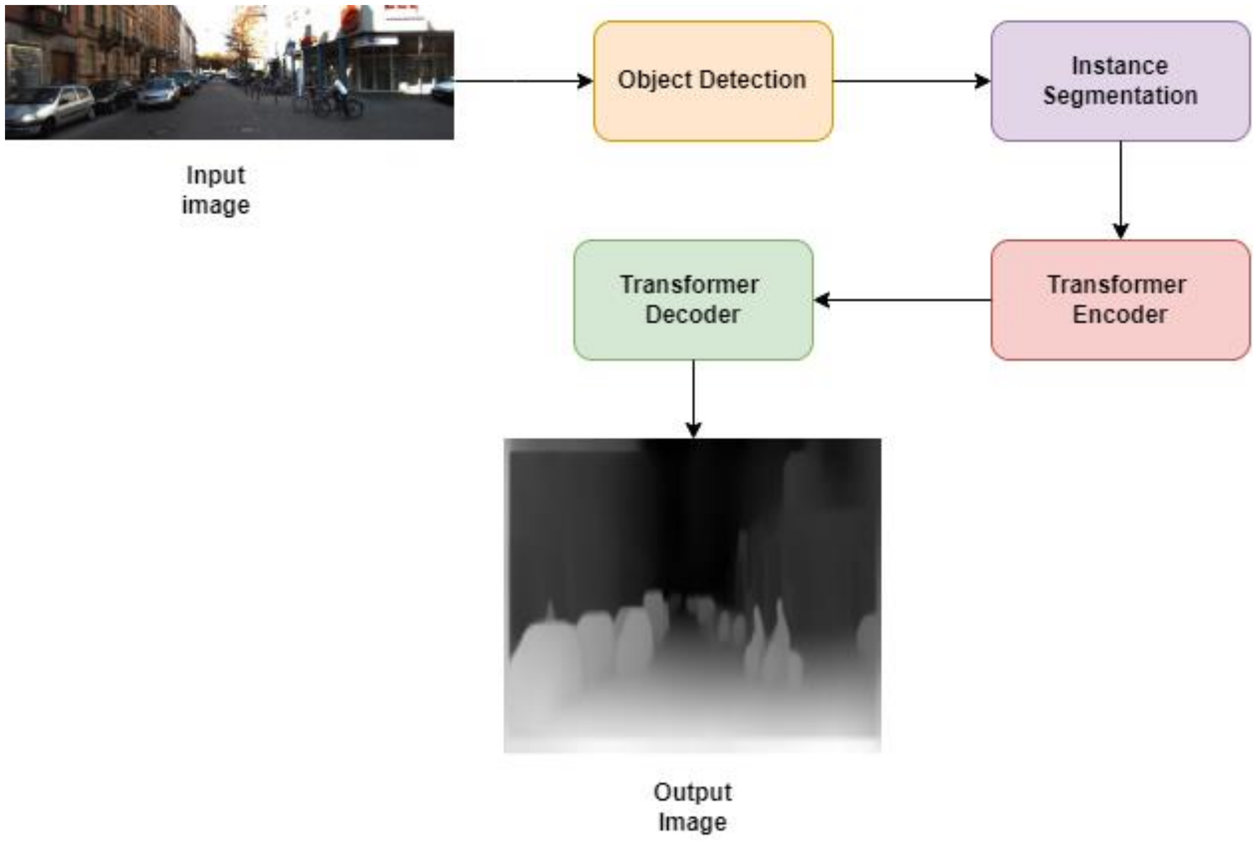
By doing this, the full-scale network, created to give the encoder an emphasis on the depth estimation task and strengthen the model's ability to represent data.

1.5 Problem Definition

To estimate the depth of each pixel in an image, given a single 2D image as input this is referred to as image depth estimation. The approach to this problem is to use machine learning techniques to learn a mapping from the 2D image to the depth map generation.

2. PROPOSED SYSTEM

2.1 Description of proposed system



2.2 Description of target users

The target users of image depth estimation are individuals and organizations working in fields where accurate and detailed information about the distance and location of objects in an image are important. In our project, our main target users are the autonomous vehicle(car) manufacturers.

2.3 Applications of our proposed system

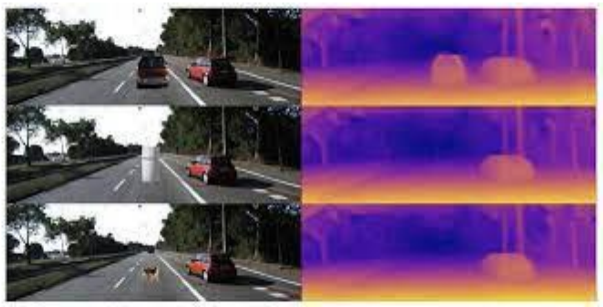
Image depth prediction has a wide range of applications, including robotics, augmented reality, and computer vision. It can be used to improve the accuracy and efficiency of object recognition and tracking, and to enable more realistic rendering of 3D scenes. It can also be used to improve the performance of autonomous vehicles and other robots by providing a more accurate understanding of their environment.

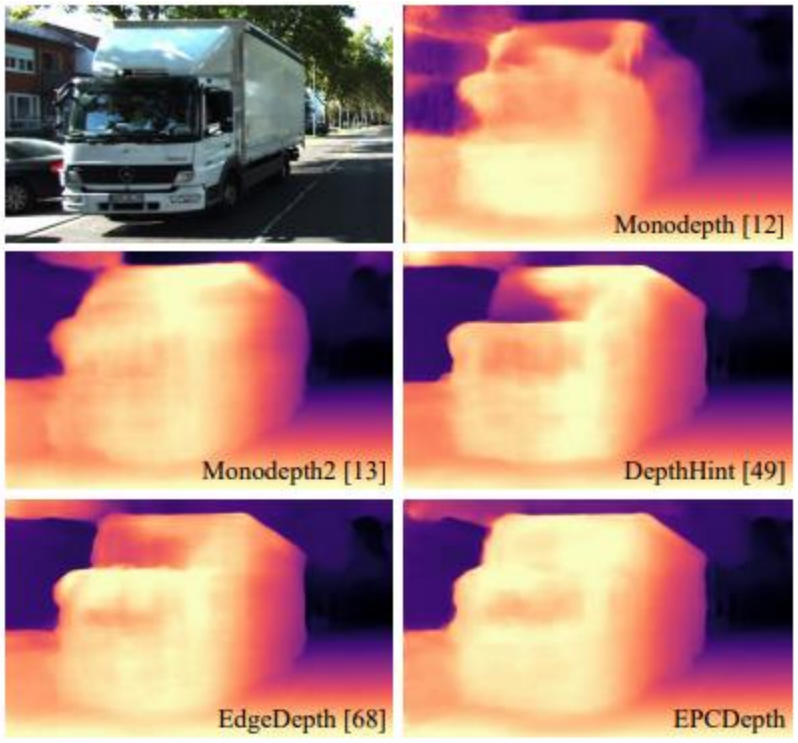
2.4 Scope of proposed system

The scope of image depth estimation is broad, as it has many potential applications in a wide range of fields and industries. The proposed system will mainly help in getting the sharp and accurate depth images of the given input images.

Our proposed system, as seen in the results section, performs better in getting a better depth map of an image compared to the traditional methods.

An accurate depth map can be used in the above mentioned applications more efficiently.





3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Overview of SRS

Development teams may develop a high-quality result with the aid of clear, simple, and actionable requirements. To organize and present these requirements Software system Requirements comes in. A software requirements specification (SRS) is a document that outlines the functions and system performance for the software. It consists of the specifications, benchmarks, and expectations for a future prototype design. It could include word docs, spreadsheets, plans, examples, and other materials that make the company's point clear. It also outlines the functionality the solution has to have in order to meet the demands of both industry and user groups.An SRS provides you with a comprehensive overview of your entire project. It offers a single point of reference that each development team will rely on. It serves as your strategy and keeps every team you work with, from development to maintenance, informed.

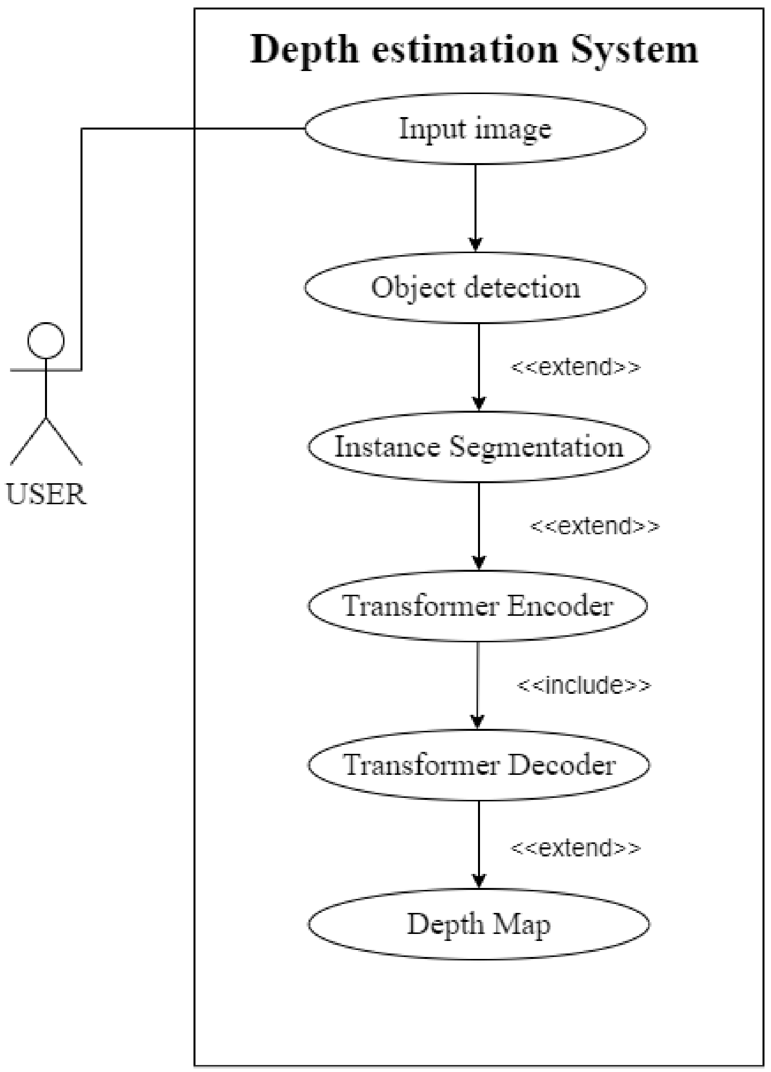
3.2 Requirement specifications

A functional requirement (FR) outlines the functionality that the system must provide. It describes a piece of software or a software system. The software system's inputs, actions, and outputs are all together referred to as a function. A system's likely purpose can be determined by a computation, data manipulation, business process, user interaction, or any other relevant feature. In software engineering, functional specifications are another name for functional requirements.The following should be part of a system's functional requirements; Information about the procedures performed on each window. The system has to have data processing logic inserted into it, along with descriptions of system outputs like reports. Full disclosure of the workflows that the system performs. It should expressly state who will be permitted to add, edit, and delete data in the system. The functional document has to provide information on how the system will satisfy any applicable regulatory and compliance requirements.

3.2.1 Functional requirements

- The system shall be able to handle a large number of images and generate the depth map for the given image.
- The system shall be able to handle with larger number of objects in a image and for that image an appropriate depth map will be generated.
- The system shall be able to handle different types of input data, such as images with different resolutions or aspect ratios.
- The system shall be able to predict the depth map which is close to the ground truth of the image.

3.2.2 Use Case Diagram



3.2.3 Use case descriptions using scenario

Use case: Detect the objects for given input image

Primary actors: User

Goal in contexts: To know the objects present in the image for further processing.

Preconditions: Given input image must have those classes which are in trained dataset.

Triggers: User wants to get an object detected for annotation.

Scenario:
1) Open the model
2) Choose the input image from testing dataset
3) Change the path in model and insert the testing image relative path
4) User run the model
5) The system displays the output with objects identified for given image
6)Identified objects have a boundary box which gives a clear idea about detected objects.

Exceptions:
1)Choosed image is not in testing dataset
2) Relative path of the testing image is not found hence it couldn't process.
3)Model fails to run for a given testing image.
4) The system could not identify all the objects.
5) Few objects in the image may not be included in training dataset.

Priority: Moderate priority to be implemented after basic functions

When available : Third increment

Frequency of use : Moderate Frequency

Channel to Actor : Via PC based Ubuntu platform.

Secondary Actor: Proposed Model

Channel to Secondary Actor: PC Ubuntu operating system

Open Issues:
1) What if all objects are not identified despite their classes being included?
2) What if the model crashes while running for new test images?

3) Will the system be able to annotate all objects identified?
4) Is a trained model enough to identify the object for new images?

3.2.4 Non-Functional requirements

Non-functional requirements (NFRs) specify restrictions on how the system should operate.The restrictions or demands placed on the system are referred to as non-functional requirements. They outline the software's quality features. Scalability, maintainability, performance, portability, security, dependability, and many more challenges are covered by non-functional requirements. Non-functional requirements focus on critical quality problems for software systems.

The non-functional requirements are as follows:

- The system should generate the depth map in a reasonable amount of time, as it may be used in real time applications.
- The system should be reliable and consistently produce accurate depth maps.
- The system should have good performance, with fast processing time and low latency.
- The system should be easy to use, with a clear and intuitive interface.

3.2.4.1 Performance requirements

- Model precision needs to be higher than 75%
- The system should generate the depth map in a reasonable amount of time, as it may be used in real time applications.
- The system should be reliable and consistently produce accurate depth maps.
- The system should have good performance, with fast processing time and low latency.

3.2.4.2 Usability

- The system should be easy to use, with a clear and intuitive interface.
- Users should be able to use an interface when they first see it with enough ease to accomplish goals without needing assistance from other sources or experts.

3.3 Software and Hardware requirement specifications

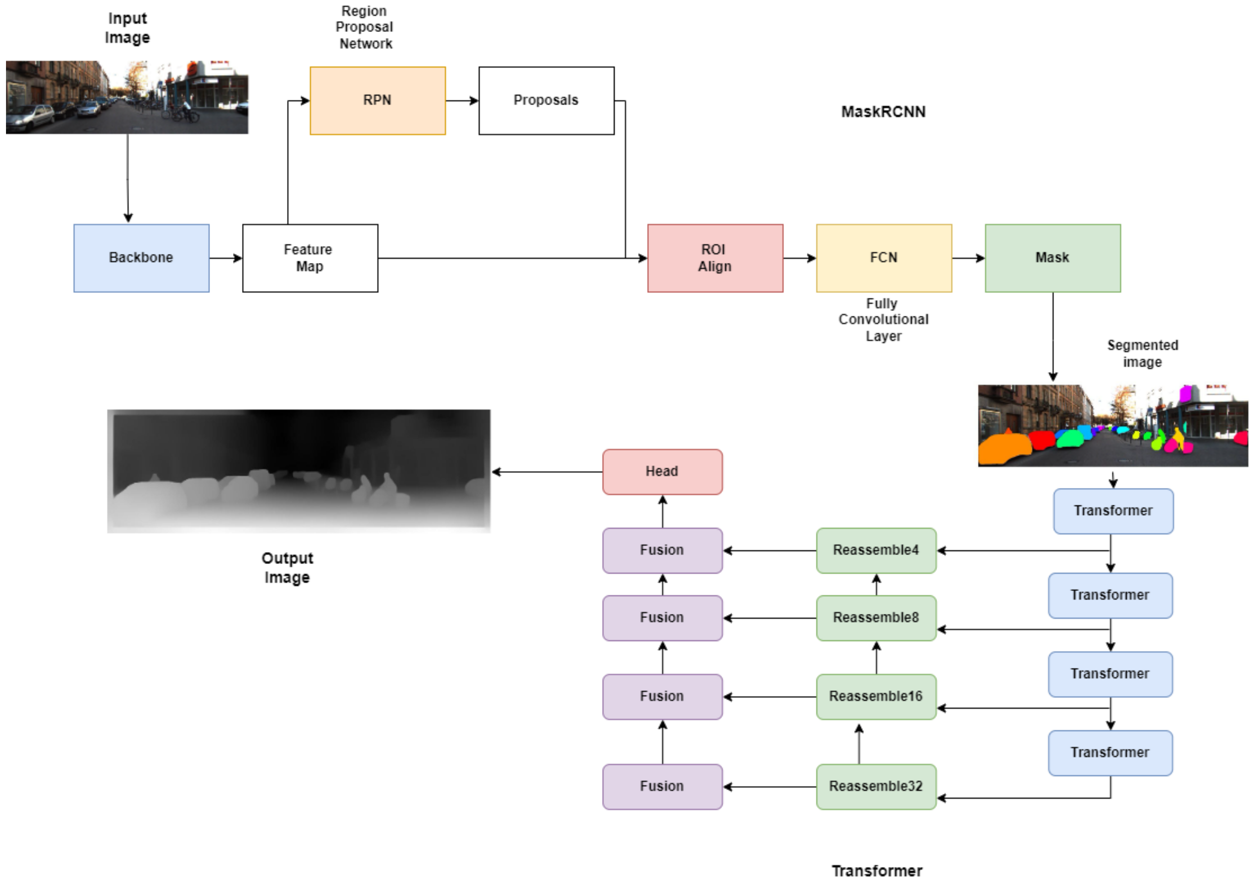
Hardware: Processor, Motherboard, GPU, Hard Disk .

Software: Ubuntu, Google Collab.

Frameworks: TensorFlow and PyTorch.

4. SYSTEM DESIGN

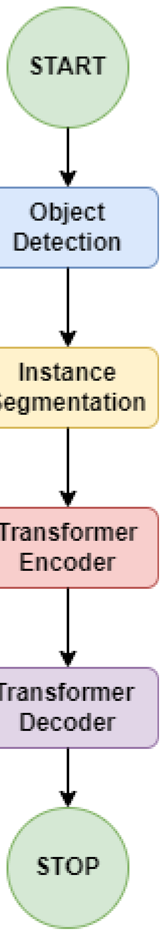
4.1 Architecture of the system



4.2 State transition diagram

A state machine diagram, state diagram, or state transition diagram is a graphical representation of a system's potential states and transitions between them. It is frequently used to simulate the behavior of a process or system, such as a business process, a software application, or a computer programme.

The states of the system are represented by circles or rectangles in a state transition diagram, and the transitions between those states are symbolized by arrows. The events or circumstances that cause the transition from one state to another are listed next to the arrows.



4.3 Data set description

The KITTI dataset is a large dataset of images and 3D point clouds captured by a vehicle-mounted LiDAR and camera system. The dataset was collected by the Karlsruhe Institute of Technology and Toyota Technological Institute and is widely used for evaluating algorithms in the fields of autonomous driving, robotics, and computer vision.

The KITTI dataset consists of a number of different data modalities, including stereo images, monocular images, LiDAR point clouds, and GPS/INS data. The dataset also includes ground truth data for a variety of tasks, including object detection, object tracking, 3D object reconstruction, and depth estimation.

One of the key features of the KITTI dataset is the wide range of scenarios it covers, including urban, suburban, and rural environments, as well as different weather conditions and lighting conditions. This makes it a useful benchmark for evaluating the performance of algorithms on a wide range of different types of data.

Overall, the KITTI dataset is a valuable resource for researchers working on problems related to autonomous driving, robotics, and computer vision, and has been widely used in a variety of research studies and applications.

Some images from the KITTI dataset.

5. IMPLEMENTATION

5.1 Proposed Methodology

This section describes how we designed a multi-picture depth prediction network using transformers. We first identify the object in the input image and then apply instance segmentation to the object before passing it to the transformers to generate a depth map without the assistance of ground truth depth.

1. **Object Detection and Instance Segmentation**

   Mask RCNN is a deep classification algorithm for object detection and instance segmentation. It is a two-stage approach that first creates potential object areas in a picture using a region proposal network, and then uses a different network to categorize and segment the regions. Mask RCNN's ability to produce excellent object masks, which allow it to correctly separate individual objects in an image, gives it a significant edge over other object recognition techniques. It is an advancement of the Faster R-CNN architecture, which introduced the use of a region proposal network (RPN) to create possible regions of interest (ROIs) in an image. Mask R-CNN extends the Faster R-CNN architecture by adding a third branch that generates a binary mask for each identified object.

2. **Transformers for Depth prediction task:**

   The output of object detection and instance segmentation is then processed as input for transformers . Transformers are made up of encoder and decoder. An encoder-decoder model for depth prediction is to use a convolutional neural network (CNN) as the encoder and a fully connected network as the decoder. The CNN can be trained to extract features from the input image, and the fully connected network can be trained to generate the depth map based on those features.

5.2 Description of Modules

1. **Object Detection and Instance Segmentation**

   Mask R-CNN detects objects in two stages. The first stage includes a feature extractor and a region proposal network. The feature extractor is often a convolutional neural network (CNN) which has been pre-trained on a large dataset, such as ImageNet, to extract important features from an input image. The region proposal network generates a set of ROIs using these features. It contains a detection network and a mask generation network in the second stage. The ROIs are used by the detection network to classify and locate each object in the image. Finally, the mask generation network uses the ROIs to build a binary mask for each observed object.

2. **Encoder Transformer :**

   Output of object detection with instance segmentation will highlight the important objects on the roads like cars, buses, human beings, traffic lights, bicycles etc. This highlighted image then will be given more attention in the encoder part. The objects in the image are tokens. To create tokens from the input image, either non-overlapping patches are extracted, their flattened representations are linearly projected, or a ResNet-50 feature extractor is used. A patch independent readout token and a positional embedding are added to the image embedding. The tokens are processed by numerous steps of transformers.

3. **Decoder Transformer:**

   The tokens from different stages are arranged in order to create an image-like representation at various resolutions. To produce a fine-grained prediction, the representations are incrementally fused and up-sampled by fusion modules. overview of the operation of reassembling. With the spatial resolution of the input image, tokens are pieced together into feature maps. The Fusion blocks upsample the feature maps and integrate features using residual convolutional units.

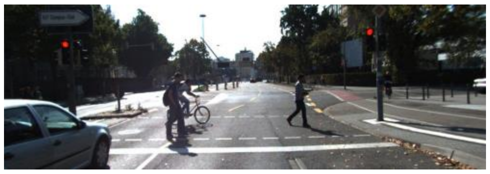
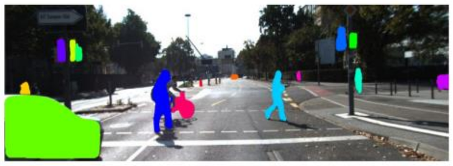
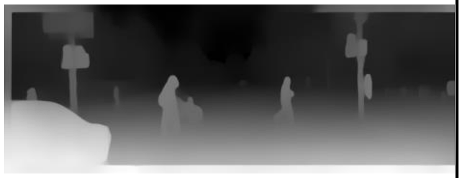
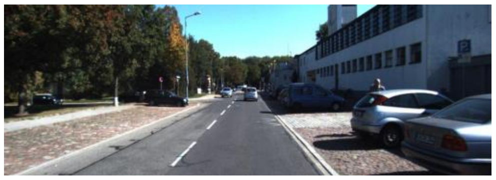
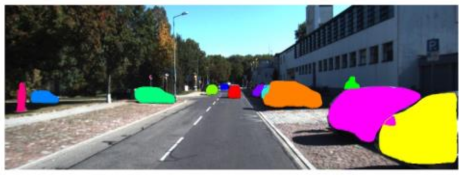
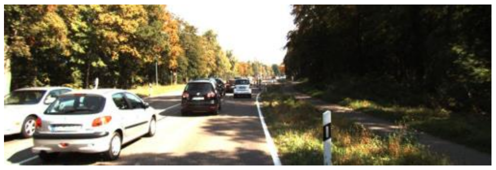
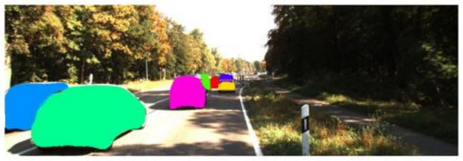
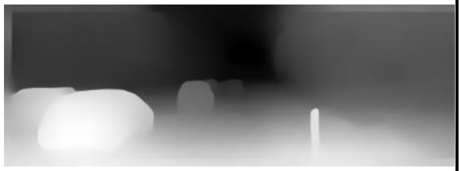
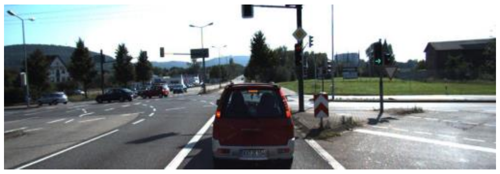
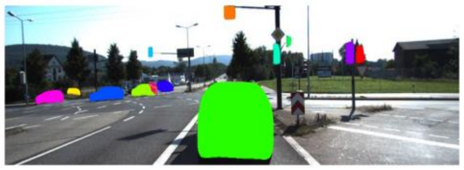
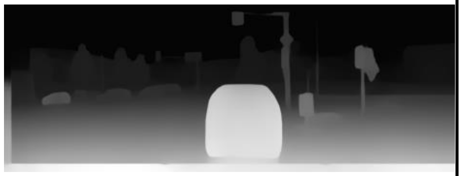
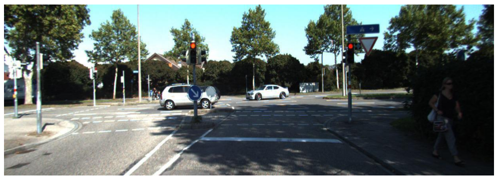
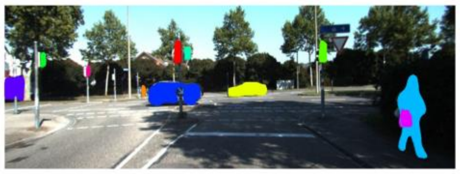
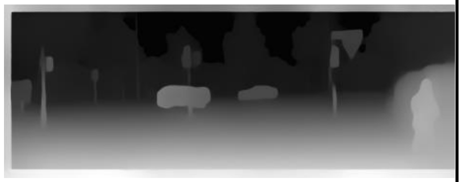
6. TESTING

6.1 Test Plan and Test Cases

Test Plan and cases

Requirement ID	**Test Case ID**	**Test Case Description**	**Test Steps**	**Input Data**	**Expected Results**	**Actual Results**	**Pass/Fail**
RE01	TC01	To generate the depth of an image with a single,clear object in the background	1.Load an image with a single car on the road 2. Generate the depth mao	Image from the KITTI Dataset	Depth map generation.	Depth map generated	Pass
RE02	TC02	To generate the depth map for images with different resolutions and aspect ratios.	1, Load 227 X 227 image. 2. Generate the depth mao	Image resized to 227 X 227 from KITTI dataset	Depth map generated properly with all the details covered in it.	Depth map generated with all the appropriate input images	Pass
RE03	TC03	To generate the depth map from images in different angles.	1.load an image with different angles captured. 2. Generate the depth map	Image clicked with different angles.	Depth map is generated.	Depth map is generated.	Pass
RE04	TC04	To generate the depth map for images with more objects in it.	1. Load an image with more vehicles in it. 2.Generate the depth map.	Image clicked from traffic scenario.	Depth map captures all the objects.	Depth map is generated with all the objects covered in it.	Pass
RE05	TC05	To generate the depth map with images with low contrast and lighting	1.load an image which has low contrast and lighting. 2. Generate the depth map	Image captured when it was raining	Depth map captures all the objects present in the image	Depth map is generated which has captured all the objects.	Pass
RE06	TC06	To generate the depth map for an image which is very far from the camera	1.Load an image where the car is very far away from the camera. 2. Generate the depth map	Load an image from KITTI dataset which matches with test case description	Depth map captures the image	Depth map is generated with the car present on the depth map.	Pass
RE07	TC07	To generate depth map where there multiple, overlapping of objects in the foreground	1.Load an image from the accident scenario. 2. Generate the depth map	Image from accident scenario.	Depth map captures all the overlapping objects	Depth map is generated with vehicles collided	Pass

7. RESULTS AND DISCUSSIONS

Input Image	Instance Segmentation	Depth Map
		
		
		
		
		

8. CONCLUSION AND FUTURE SCOPE

We proposed a novel methodology to estimate the depth of an image by designing a multi-picture depth prediction network using transformers. To create a depth map without the aid of ground truth depth, the input image's objects were identified, and after instance segmenting them, the objects were passed to the transformers.

Looking ahead, it is anticipated that research on image depth prediction will remain busy, with continued efforts to increase algorithm efficiency and accuracy as well as the creation of new tools and technologies that use depth data. Overall, the issue of producing depth maps is an active area of research with continual efforts to increase algorithm precision and effectiveness as well as the creation of new technologies and applications that rely on depth data. With the potential to have an impact on a wide range of applications and technologies, it is expected that depth maps will continue to be a crucial tool for comprehending the 3D structure of scenes and the relative placements of objects.

9. REFERENCES

[1] Zhaowei Cai Nuno Vasconcelos.Cascader-cnn:Delving into high quality object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[2] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics Automation Magazine*, 22(3):36–52, 2019.

[4] Nicolas Carrion,FranciscoMassa,GabrielSynnaeve,Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End- to-end object detection with transformers. In *The European Conference on Computer Vision (ECCV)*, 2020.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei- Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox. Self-supervised 6d object pose estimation for robot manipulation. In *The IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[8] M. Ferguson and K. Law. A 2d-3d object detection sys- tem for updating building information models with mobile robots. In *IEEE Winter Conference on Applications of Com- puter Vision (WACV)*, 2019.

[9] Steven H. Ferris. Motion parallax and absolute distance. *Journal of Experimental Psychology*, 95(2):258–263, 1972.

[10] L.Gan,R.Zhang,J.W.Grizzle,R.M.Eustice,andM.Ghaffari. Bayesian spatial kernel smoothing for scalable dense semantic mapping. *IEEE Robotics and Automation Letters (RA-L)*, 5(2), 2020.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] F. A. Gers and J. Schmidhuber. Recurrent nets that time and count. In *The IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN)*, 2000.

[13] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *The IEEE/CVF International Conference on Com- puter Vision (ICCV)*, 2019.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[15] Cle ́ment Godard, Oisin Mac Aodha, and Gabriel J Bros- tow. Unsupervised monocular depth estimation with left- right consistency. In *CVPR*, 2017.

[16] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

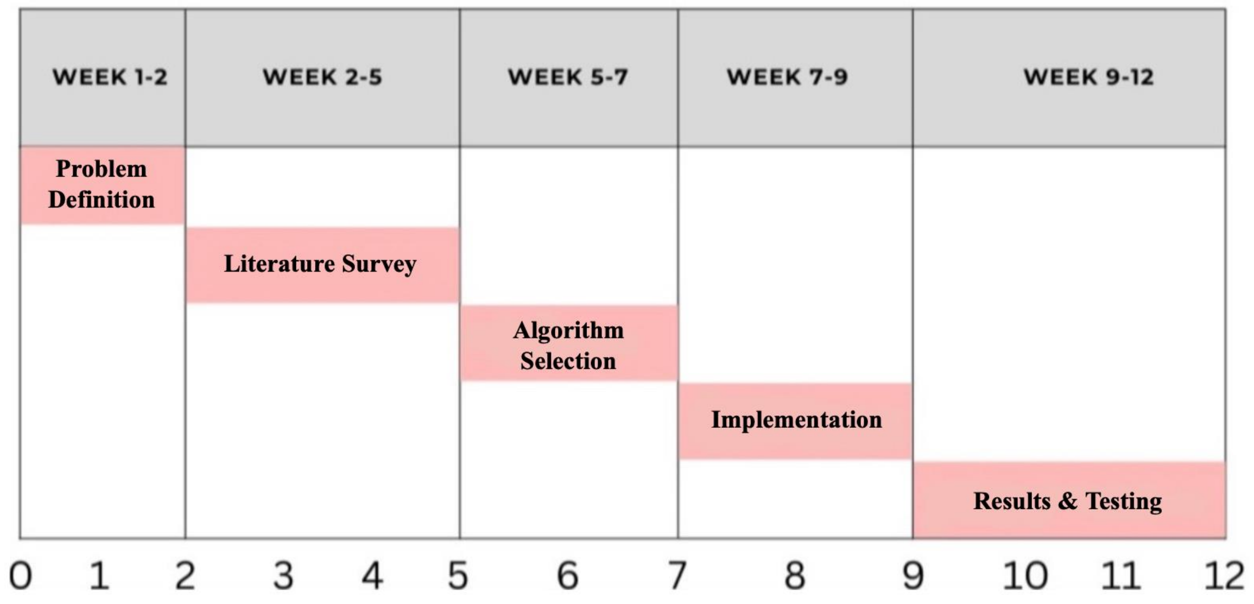
[18] Carol Barnes Hochberg and Julian E Hochberg. Familiar size and the perception of depth. *The Journal of Psychology*, 1952.

[19] Derek Hoiem, Alexei A Efros, and Martial Hebert. Auto- matic photo pop-up. *TOG*, 2005.

[20] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet2: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

10. APPENDIX

10.1 Gantt Chart



10.2 Description of tools and Technology used

Ubuntu: Based on the Linux kernel, Ubuntu is a free and open-source operating system. It is a well-liked option for consumers who seek an intuitive operating system that is dependable, safe, and simple to use. Ubuntu may be installed on a wide range of hardware, including desktop PCs, servers, and mobile phones. It is supported by a large selection of programmes and tools that can be found in the Ubuntu Software Center, and it is created and maintained by a global community of volunteers. Ubuntu is popular in both private and professional contexts and is renowned for its simplicity, usability, and adaptability.

Google Collab: Google Colab is a free online tool that gives users access to an environment similar to a Jupyter notebook where they may run and execute code. It is a cloud-based platform that gives users access to numerous potent computing resources, such as GPUs and TPUs, making it the perfect option for processes involving machine learning and data science. With Colab, users may import and display data, develop and execute code in Python and other programming languages, and work in real-time collaboration with others. For academics, data scientists, and students who don't want to install and configure their own gear and software but still require access to high-performance computing resources, Colab is a handy tool.It is also a smart option for users who wish to test out machine learning methods without committing to a particular infrastructure or environment.

TensorFlow: An open-source software library for artificial intelligence and machine learning is called TensorFlow. It was made by Google and is employed in the creation, construction, and training of machine learning models. TensorFlow is used for a variety of applications, including image and audio recognition, natural language processing, and predictive analytics. It is versatile and effective by design.

Tensors, which are multi-dimensional data arrays, form the foundation of the TensorFlow algorithm. TensorFlow allows users to create sophisticated models by combining low-level and high-level operations, as it represents and executes machine learning algorithms using a graph-based method. Python is the language used to implement TensorFlow, although it also offers APIs for C++, Java, and Swift.

TensorFlow is a well-liked option for creating deep learning models and is widely utilized in both research and production scenarios. It is a versatile and effective tool for a number of machine learning applications due to its great scalability and ability to run on several platforms, including CPUs, GPUs, and TPUs.

PyTorch: Based on the Torch library, PyTorch is an open-source machine learning library for Python. It was created by Facebook's AI Research unit and is employed in many different fields, such as machine learning, computer vision, and natural language processing.

Tensors are multi-dimensional data arrays, and PyTorch and TensorFlow both use this idea as their foundation. Because PyTorch uses a dynamic computational graph, users can alter the graph as they go, giving them more freedom when creating and refining models. A library for creating and training neural networks is among the high-level libraries and tools that PyTorch offers to make it simpler to create and use machine learning models.

Due to its adaptability and simplicity of use, PyTorch is well-liked among researchers and practitioners in the machine learning community. It is a good option for developing and implementing machine learning models in real-world settings and is widely used in industry.

10.3 Blueprint

Project information:

Team Number	I8
Guide	Dr.Meena S. M
Project title	Image Depth Estimation
Industry name	Bosch
Department vertical(either of Data/ Network/System)	Data Analytics/ AI and ML
University/ Department Research group	