

An Efficient Stereo Vision Algorithm with Configurable Image Rectification and Disparity Refinement

Submitted in partial fulfilment of the requirements for the award of Bachelor of
Engineering degree in Electronics and Communication Engineering

By

V. Gangaraju (Reg.No - 40130271)

B. Vishwa Teja (Reg.No - 40130272)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119

April - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

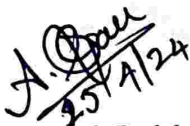
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of V. GANGARAJU (40130271) and B. VISHWA TEJA (40130272) who carried out the project entitled "AN EFFICIENT STEREO VISION ALGORITHM WITH CONFIGURABLE IMAGE RECTIFICATION AND DISPARITY REFINEMENT" under my supervision from November 2023 to April 2024.


25/4/24
Internal Guide

DR. A. SAHAYA ANSELIN NISHA, M.E., Ph.D


Head of the Department
DR. T. RAVI, M.E., Ph.D

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I B. Vishwa Teja (Reg.No - 40130272) hereby declare that the Project Report entitled "An Efficient Stereo Vision Algorithm with Configurable Image Rectification and Disparity Refinement" done by me under the guidance of Dr. A. Sahaya Anselin Nisha M.E., Ph.D at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 15-04-2024

PLACE: Chennai

1. Vishwa Teja

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of Sathyabama Institute of Science and Technology for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D.**, prof. & Dean, School of Electrical and Electronics, **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to our Project Guide **Dr. A. SAHAYA ANSELIN NISHA, M.E., Ph.D.**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Stereo matching, the process of estimating dense disparity maps from stereo image pairs, is a fundamental task in computer vision with various applications such as 3D reconstruction, autonomous driving, and augmented reality. In this project, a patch-based stereo matching approach using Siamese neural networks is proposed.

The Siamese architecture enables learning of similarity metrics between patches, allowing for robust matching across stereo images. The Siamese NetStereo Matching architecture, which encodes image patches into feature vectors; the `similarity_score` function, which computes the similarity scores between patches using cosine similarity; and the `hinge_loss` function, which formulates the loss function for training the Siamese network. The KITTI dataset is utilized for training and validation, providing stereo image pairs along with ground truth disparity maps.

The Patch Provider class efficiently generates training batches by randomly selecting patches from stereo image pairs. During training, the Siamese network parameters are optimized using stochastic gradient descent with backpropagation. The performance of the trained model is evaluated using metrics such as training and validation loss, accuracy, and endpoint error (EPE) between predicted similarity scores and ground truth disparities. Experimental results demonstrate the effectiveness of the proposed approach in accurately estimating dense disparity maps from stereo image that been taken as a set of pairs.

CONTENTS

CHAPTER NO.	TITLE	PAGE.NO
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	1
	1.1 OVERVIEW.	1
2.	LITERATURE REVIEW	4
	2.1 INTRODUCTION	4
	2.2 SUMMARY	8
3.	REQUIREMENT ANALYSIS	9
	3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT.	9
	3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT.	10
4.	DESCRIPTION OF PROPOSED SYSTEM	13
	4.1 SELECTED METHODOLOGY OR PROCESS MODEL.	14
	4.1.1 WORKING PRINCIPLE	14
	4.2 ARCHITECTURE/OVERALL DESIGN OF PROPOSED SYSTEM.	15
	4.2.1 DATA COLLECTION	15
	4.2.2 MODEL TRAINING	17
	4.2.3 SIAMESE NETWORK	19
	4.2.4 MODEL TESTING	21
	4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING.	22
	4.3.1 COMPONENTS	22
	4.3.2 WORKFLOW	22
	4.3.3 IMPLEMENTATION PLAN	23

4.4	PROJECT MANAGEMENT PLAN.	24
4.5	FINANCIAL REPORT ON ESTIMATED COSTING.	25
4.5.1	BUDGET OVERVIEW	25
4.6	TRANSITION/SOFTWARE TO OPERATIONS PLAN.	26
4.6.1	TRANSITION PLAN	26
4.6.2	OPERATIONS PLAN	26
4.7	STANDARDS	27
4.8	CONSTRAINTS	28
4.9	TRADE OFF	28
5.	RESULT AND ANALYSIS	29
6.	CONCLUSION	35
7.	REFERENCES	36
8.	APPENDIX	38

LIST OF FIGURES

FIG.NO.	TITLE	PAGE NO.
1.1	The 3D scene projected into a plane.	2
1.2	A 2D projection of a scene. “Teddy” image (Scharstein, 2010).	3
4.1	A multi-layer perceptron model of neural network.	14
4.2	Stereo left and right images	16
4.3	Architecture Diagram	18
4.4	Siamese network architecture	20
5.1	Training loss and accuracy	30
5.2	Validation loss and accuracy	31
5.3	Disparity	33

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Depth estimation or extraction refers to the set of techniques and algorithms aiming to obtain a representation of the spatial structure of a scene. In other terms, to obtain a measure of the distance of, ideally, each point of the seen scene. This project will also present about 3D vision.

This chapter will review the main topics, problems and proposals about depth estimation, as an introduction to the Stereo Vision research field. This work will deal with some essential and structural aspects of the image processing field, as well as with the depth perception capabilities and conditions of both computer and human based systems.

This chapter is organized as follows:

- This introduction section will present some basic concepts and problems of the depth estimation.
- The depth estimation strategies section will detail, analyze and present results of the main families of algorithms which solve the depth estimation problem, among them, the stereo vision based approaches.
- Finally, a conclusions section will summarize the pros and contras of the main paradigms seen in the chapter.

This project will call the “3D scene” to the set of objects placed in a three dimensional space. A scene, however, is always seen from a specific point. The distorted image that is perceived in that point is the so-called projection of the scene. This projection is formed by the set of rays crossing a limited aperture arriving to the so-called projection plane which is shown in figure 1.1.

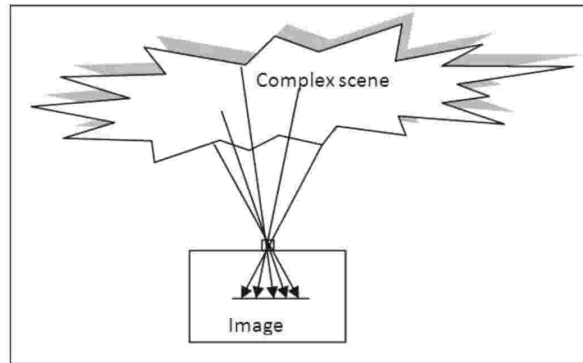


Fig:1.1 : The 3D scene projected into a plane.

This projection presents some relevant characteristics:

- The most evident consequence of a projection is the loss of one dimension. Since in each pixel only one point of the real scene is projected, the depth information is mathematically erased during the projection process into the image plane. However, some algorithms can retrieve this information from the 2D image.
- On the contrary, the projection of a scene presents important advantages, such as simple sampling by already well developed devices (the so-called image sensors). Moreover, dealing with 2D images is, by obvious reasons, much simpler than managing 3D sets of data, reducing computational load.

Thus, the scene is transformed into a 2D set of points, which can be described in a Cartesian plane:

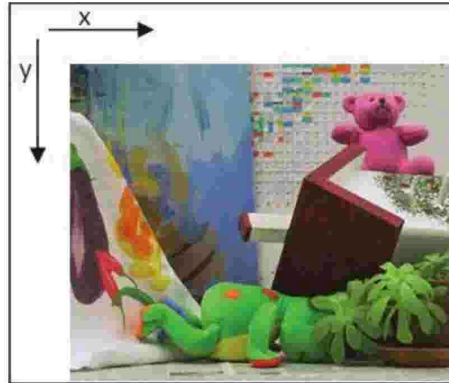


Fig:1.2 : A 2D projection of a scene. “Teddy” image (Scharstein, 2010).

The 3D vision processes have as goal the reconstruction of this lost information, and, thus, the distances from each projected point to the image plane. An example of such reconstruction is shown in figure 1.2.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This literature survey explores recent advancements in stereo vision algorithms and hardware accelerators, focusing on their applications in real-time stereo imaging and depth estimation. Through a detailed examination of journal articles published in IEEE, this survey sheds light on innovative methodologies, such as model-based design, binary neural networks.

The journal article by Akshay Jain and Pulkit Goel, titled "Symmetric k-Means for Deep Neural Network Compression and Hardware Acceleration on FPGAs," published in IEEE in 2020 [1], introduces symmetric k-means, an innovative approach for deep neural network compression and hardware acceleration on FPGAs. This technique focuses on reducing inference-time latency, hardware-specific optimization, efficient model compression, and scalability. Notably, it ensures no loss in accuracy while providing cost-effective hardware acceleration. This research showcases the versatility and efficiency of symmetric k-means in accelerating deep learning tasks on FPGA platforms.

The journal article by CHUNBO CHENG and HONG LI, titled "Two-Branch Convolutional Sparse Representation for Stereo Matching," published in IEEE in 2021 [2], introduces a novel two-branch convolutional sparse representation technique for stereo matching. This technique offers an efficient algorithm for unsupervised stereo matching cost estimation (USMC), resulting in high accuracy and generalization. By combining convolutional neural networks with sparse representation, this approach demonstrates promising results in stereo matching tasks, contributing to the advancement of stereo vision algorithms.

The journal article by Chiang-Heng Chien and Chen-Chien James Hsu, titled "Multiple Master-Slave FPGA Architecture of a Stereo Visual Odometry," published in IEEE in 2021 [3], presents a multiple master-slave FPGA architecture tailored for stereo visual

odometry. This architecture prioritizes efficiency, real-time processing, low power consumption, and cost-effectiveness. Notable features include scalability, reconfigurability, hardware-implemented image pyramid, local linear exhausted search (LES) matching, and pose estimation enhancement. This research offers a versatile solution for stereo visual odometry tasks with potential for broader applications.

The journal article by D. S. Kaputa and K. A. Derhak, titled "Model Based Design of a Real Time FPGA-Based Lens Undistortion and Image Rectification Algorithm for Stereo Imaging," published in IEEE in 2023 [4], serves as a foundational piece for the literature survey . It explores the application of model-based design in developing a real-time FPGA-based lens undistortion and image rectification algorithm for stereo imaging. The features highlighted in the journal align with the benefits of employing model-based design, including reduced development time, improved quality, better communication, reduced complexity, improved performance, and increased flexibility. This study provides essential insights for researchers and engineers seeking efficient and effective methodologies for stereo imaging system development.

The journal article by Gang Chen and Yehua Ling, titled "StereoEngine: An FPGA-Based Accelerator for Real-Time High-Quality Stereo Estimation With Binary Neural Network," published in IEEE in 2020 [5], presents StereoEngine, a specialized FPGA-based accelerator designed for real-time, high-quality stereo estimation using binary neural networks (BNNs). It focuses on real-time performance, energy efficiency, and hardware-friendly implementation. Notable features include standalone hardware architecture, significant speedup, energy efficiency improvement, and leading accuracy. This research showcases the potential of FPGA-based acceleration for stereo vision tasks.

The journal article by Jung-Gyun Kim and Donghwan Seo, titled "Spiking Cooperative Network Implemented on FPGA for Real-Time Event-Based Stereo System," published in IEEE in 2022 [6], introduces a spiking cooperative network architecture implemented on FPGA for real-time event-based stereo vision. This approach prioritizes optimal hardware resource utilization, parallel processing, and low hardware footprint. It achieves comparable accuracy to traditional methods while enabling real-time event-

based processing. Additionally, it demonstrates scalability and offers potential for further improvement, showcasing a unique approach in stereo vision system design.

The journal article by Kai Yit Kok and Parvathy Rajendran, titled "A Review on Stereo Vision Algorithms: Challenges and Solutions," published in IEEE in 2019 [7], provides a comprehensive overview of stereo vision algorithms, focusing on challenges and solutions in the field. This review serves as a valuable resource for researchers and practitioners, offering insights into algorithm design, time-saving strategies, and the potential of hardware acceleration. By addressing both local and global approaches, this review facilitates a deeper understanding of stereo vision techniques.

The journal article by Pingcheng Dong and Zhuoyu Chen, titled "Configurable Image Rectification and Disparity Refinement for Stereo Vision," published in IEEE in 2022 [8], proposes configurable techniques for image rectification and disparity refinement tailored for stereo vision systems. It emphasizes buffer size reduction, high-precision rectification, parallel processing, and resource efficiency. Additionally, it highlights high throughput, configurability, FPGA implementation, and comprehensive stereo matching as key features. This research contributes to enhancing the performance and flexibility of stereo vision systems.

The journal article by Pingcheng Dong and Zhuoyu Chen, titled "A 4.29nJ/pixel Stereo Depth Coprocessor With Pixel Level Pipeline and Region Optimized Semi-Global Matching for IoT Application," published in IEEE in 2022 [9], presents a stereo depth coprocessor optimized for IoT applications. It features dense and robust disparity maps, addressing challenges specific to IoT environments. This coprocessor utilizes a pixel-level pipeline and region-optimized semi-global matching to achieve high processing speed and energy efficiency while maintaining accuracy. The hardware demonstration underscores its practical applicability in resource-constrained scenarios

The study presents "Lite-Stereo," an innovative hardware accelerator tailored for real-time stereo estimation via binary neural networks [10]. Highlighting its core strengths, the research underscores its exceptional hardware efficiency, catering to low-resource

settings without compromising performance. Moreover, the accelerator prioritizes portability, ensuring its adaptability across various platforms and environments. With a focus on optimizing computational resources, Lite-Stereo stands out as a promising solution for stereo estimation tasks, offering a balance between high-quality results and resource constraints, thus contributing significantly to the field of computer vision and hardware acceleration.

The journal article by Zhikai Li and Liping Ma, titled "Hardware-Oriented Algorithm for High-Speed Laser Centerline Extraction Based on Hessian Matrix," published in IEEE in 2021 [11], presents a hardware-oriented algorithm tailored for high-speed laser centerline extraction using the Hessian matrix. This algorithm prioritizes fast and efficient processing, parallel processing, robustness, subpixel precision, noise reduction, and versatility. By leveraging hardware implementation, it addresses the challenges of real-time laser centerline extraction, showcasing its potential for various applications requiring precise and rapid processing of laser data.

The journal article by Zhimin Lu and Jue Wang, titled "A Resource-Efficient Pipelined Architecture for Real-Time Semi-Global Stereo Matching," published in IEEE in 2022 [12], introduces a novel architecture optimized for real-time semi-global stereo matching. This architecture emphasizes efficiency through a pipelined design, weighted path cost aggregation, and resource savings. Additionally, it highlights FPGA implementation, high throughput, adaptability, impressive processing speed, and resource efficiency as key features. This work contributes significantly to advancing stereo matching techniques, particularly in terms of accuracy, performance, and resource utilization

2.2 SUMMARY

These studies provide valuable insights into the evolving landscape of stereo vision techniques, offering valuable guidance for researchers and engineers seeking to develop robust and resource-efficient solutions for stereo imaging tasks. Understanding the resource-efficient approaches and high-quality results achieved in real-time stereo

estimation informed the decision to leverage Siamese neural networks for stereo vision code development, aiming to optimize computational efficiency while maintaining accuracy. This survey serves as a roadmap for integrating cutting-edge techniques into this stereo vision codebase, ensuring its competitiveness and relevance in the evolving landscape of computer vision.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

Technical Feasibility has

- a. Hardware Requirements: Evaluate the hardware requirements for stereo vision systems, including cameras, processors, and memory.
- b. Software Capabilities: Assess the availability and compatibility of software tools and libraries for stereo vision algorithms and image processing.
- c. Dataset Acquisition: Ensure access to diverse and comprehensive datasets for stereo vision training and testing, considering factors like image quality and depth accuracy.

Operational Feasibility has

- a. Calibration Process: Determine the feasibility of calibrating stereo vision cameras for accurate depth perception and 3D reconstruction.
- b. Real-time Processing: Evaluate the system's capability for real-time image processing and depth map generation to meet operational requirements.
- c. User Interface: Design an intuitive interface for users to interact with stereo vision systems, enabling tasks such as object detection and scene analysis.

Economic Feasibility has

- a. Cost Analysis: Conduct a cost-benefit analysis considering expenses associated with hardware, software, dataset acquisition, and system maintenance.
- b. Return on Investment: Assess the potential return on investment from implementing stereo vision systems in terms of improved efficiency, accuracy, and cost savings.
- c. Alternative Solutions: Explore alternative approaches or technologies if the cost of implementing stereo vision proves prohibitive.

Data Quality and Quantity:

- a. Image Quality: Address issues related to image quality, resolution, and noise that may affect the accuracy of depth perception and object detection.
- b. Dataset Size: Ensure an adequate dataset size to train stereo vision algorithms effectively, considering variations in lighting conditions, object shapes, and environments.

Algorithm Training and Validation:

- a. Overfitting/Underfitting: Implement techniques to prevent overfitting or underfitting during algorithm training, such as regularization and

cross-validation. b. Hyperparameter Optimization: Experiment with different hyperparameters to optimize algorithm performance, considering factors like stereo matching parameters and neural network architecture.

Real-time Processing: a. Computational Resources: Ensure that the system has sufficient computational resources for real-time stereo vision processing, minimizing latency and delays. b. Performance Monitoring: Implement mechanisms to monitor system performance and address any issues related to processing speed, resource utilization, or memory constraints.

User Acceptance: a. Accuracy Expectations: Manage user expectations regarding the accuracy and limitations of stereo vision systems for tasks like object detection and depth estimation. b. User Interface Design: Conduct user testing to assess the usability and effectiveness of the interface, incorporating feedback to improve user experience and acceptance.

Legal and Ethical Considerations: a. Privacy Protection: Implement measures to protect the privacy of individuals captured by stereo vision systems, adhering to regulations and best practices for data handling and storage. b. Compliance: Ensure compliance with relevant laws and regulations governing data privacy, surveillance, and consent in the deployment of stereo vision technology.

Deployment and Maintenance: a. Scalability: Plan for the scalability of stereo vision systems to accommodate future growth and expansion in terms of data volume, processing demands, and user requirements. b. Maintenance Strategy: Develop a maintenance plan to address software updates, hardware upgrades, and algorithm refinements to keep the system operational and effective over time.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant Whitespace. Its language constructs and object-oriented approach aim to help programmers write

clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

PIP is the package management system used to install and manage software packages written in Python.

NumPy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones: A powerful N-dimensional array object

- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python. We can analyze data in pandas with

1. Series

2. Data frames

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and macOS. Anaconda distribution comes with 1,500 packages selected from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command-line interface (CLI).

Tensor flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Keras is an open-source neural-network library written in Python. it focuses on being user-friendly, modular, and extensible. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems enabling it to even mimic human behaviour. Its applications lie in fields of Computer Vision, Natural Language Processing, Robotics, Speech Recognition, etc. Advantages of using AI are improved customer experience, accelerate speed to market, develop sophisticated products, enable cost optimisation, enhance employee productivity and improve operational efficiency. Machine Learning (ML) is a subset of AI which is programmed to think on its own, perform social interaction, learn new information from the provided data and adapt as well as improve with experience. Although training time via Deep Learning (DL) methods is more than Machine Learning methods, it is compensated by higher accuracy in the former case. Also, DL being automatic, large domain knowledge is not required for obtaining desired results unlike in ML.

Neural Networks (NN) form the base of deep learning, a subfield of machine learning where the algorithms are inspired by the structure of the human brain. NN take in data, train themselves to recognize the patterns in this data and then predict the outputs for a new set of similar data. NN are made up of layers of neurons. These neurons are the core processing units of the network. First this thesis will have the input layer which receives the input; the output layer predicts our final output. In between, exist the hidden layers which perform most of the computations required by the project's network.

The predicted output is compared against the actual output to realize the error in prediction. The magnitude of the error gives an indication of the direction and magnitude of change to reduce the error. This information is then transferred backward through the described network. This is known as back propagation. Now based on this information the weights are adjusted. This cycle of forward propagation and back propagation is iteratively performed with multiple inputs. Neural Networks may take hours or even months to train but time is a reasonable trade-off when compared to its scope. Several experiments show that after pre-processing, neural network

classification algorithm was the best more specifically CNN(Convolutional Neural Network) as compared to Support Vector Machine(SVM),Random Forest Field.

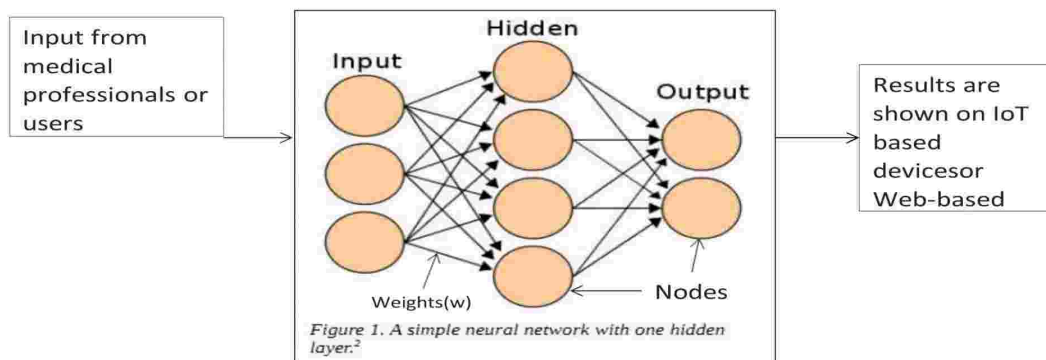


Fig:4.1 : A multi-layer perceptron model of neural network

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

4.1.1 Working principle

Data Acquisition and Preprocessing Obtain stereo image pairs along with ground truth disparity maps from datasets such as KITTI. Preprocess the images by converting them to grayscale and resizing if necessary. Normalize the pixel values to a suitable range, typically between 0 and 1.

Siamese Network Architecture Design Design the architecture of the Siamese neural network for patch-based stereo matching. Define the convolutional layers, activation functions, and normalization layers. Implement the forward pass method to encode image patches into feature vectors.

Similarity Score Calculation Implement the similarity_score function to compute the similarity scores between pairs of patches.

Utilize cosine similarity or other suitable similarity metrics to measure the similarity between feature vectors.

Loss Function Definition Define the `hinge_loss` function to formulate the loss function for training the Siamese network. Design the loss function to encourage the model to assign higher similarity scores to positive patch pairs than to negative patch pairs.

Training Setup Initialize hyperparameters such as batch size, learning rate, and margin for the hinge loss. Choose an optimizer such as SGD or Adam to optimize the network parameters. Split the dataset into training and validation sets for model evaluation.

Training Procedure Iterate over training batches generated by the Patch Provider class. Compute similarity scores for positive and negative patch pairs using the Siamese network. Calculate the hinge loss and update the network parameters through backpropagation. Periodically evaluate the model on the validation set to monitor performance.

Model Evaluation Assess the performance of the trained model using metrics such as training and validation loss, accuracy, and endpoint error. Visualize the disparity maps generated by the model and compare them with ground truth maps.

Results Analysis the results to understand the strengths and limitations of the proposed approach. Discuss potential areas for improvement and future directions for research.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

4.2.1. DATA COLLECTION

- The project begins with collecting stereo image datasets suitable for stereo matching tasks. datasets include KITTI 2012.
- Stereo image pairs are captured using stereo camera setups or synthesized using computer graphics techniques.
- *Description:* Captures input images from two synchronized cameras.
- *Functions:* Acquires raw image data for processing.



Fig:4.2 : Stereo left and right images

Data preprocessing

Image Loading : Load stereo image pairs and corresponding ground truth disparity maps.

Color Space Conversion : Convert color images to grayscale if necessary, as stereo matching often operates on grayscale images to reduce computational complexity.

Downsampling : Optionally, downsample images to reduce computational load while maintaining essential information. This downsampling might be done consistently across the dataset.

Normalization: Normalize pixel values to a common scale, often ranging from 0 to 1, to ensure consistent data ranges.

Data Augmentation: Augment data by applying transformations such as rotation, flipping, and brightness adjustments to increase dataset diversity and improve model generalization.

Dataset Splitting: Divide the dataset into training, validation, and test sets to facilitate model training, hyperparameter tuning, and performance evaluation.

Patch Extraction: Extract patches from stereo image pairs along with their corresponding ground truth disparities. Each patch consists of a reference patch from the left image and positive and negative patches from the right image.

Batch Generation: Implement a data loader or batch generator to efficiently provide batches of training samples during model training. These batches typically include batches of reference, positive, and negative patches.

4.2.2. MODEL TRAINING

- **Model Initialization:** Instantiate the Siamese neural network model, defining its architecture and parameters.
- **Loss Function Selection:** Choose a suitable loss function for training the model. In this project, hinge loss is commonly used for training Siamese networks.
- **Optimizer Configuration:** Select an optimizer (e.g., SGD, Adam) and configure its parameters, such as learning rate and momentum.
- **Training Loop:** Iterate over batches of training samples, forward pass through network, compute loss, perform back propagation and update model parameters iteratively.
- **Validation:** Periodically evaluate the model's performance on the validation set to monitor training progress and prevent overfitting.

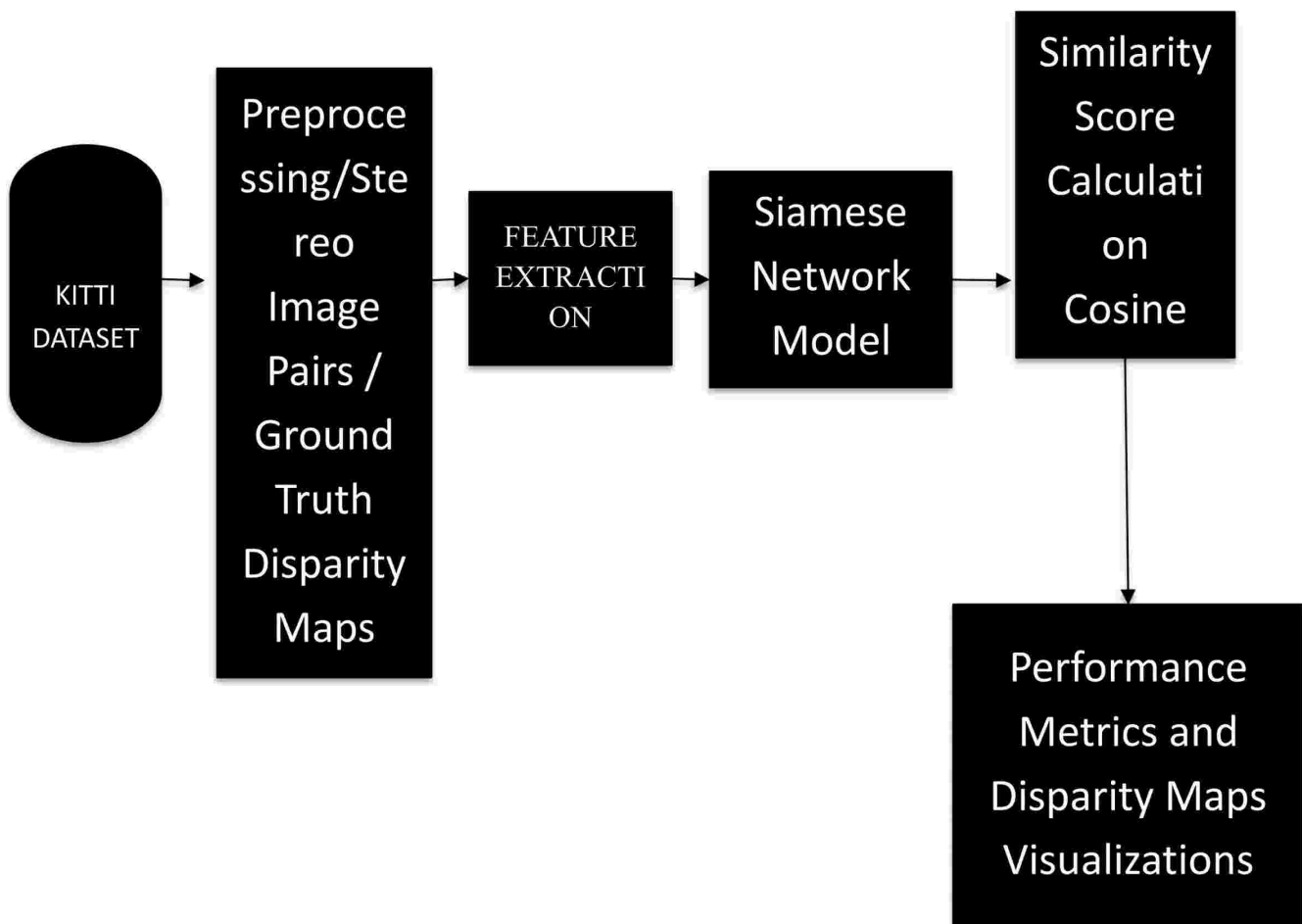


Fig:4.3 : Architecture Diagram

This is architecture diagram describing the flow of functions that will be set in motion in order for the algorithm to work and produce a disparity map as output. This Siamese network model plays a crucial role in increasing the accuracy with the feedback provided by it.

4.2.3 SIAMESE NETWORK

Convolutional layers are the primary building blocks of deep neural networks used for processing image data.

Each convolutional layer consists of a set of learnable filters (also known

as kernels) that slide across the input image, computing dot products between the filter weights and local input regions.

The output of a convolutional layer is a feature map that captures hierarchical representations of the input image, extracting features such as edges, textures, and shapes.

In the Siamese network, multiple convolutional layers are stacked to learn hierarchical representations of input patches.

Activation functions introduce non-linearity into the network, enabling it to learn complex relationships between input and output.

Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh.

ReLU is often used in convolutional neural networks (CNNs) due to its simplicity and effectiveness in mitigating the vanishing gradient problem.

In the Siamese network, ReLU activation functions are typically applied after each convolutional layer to introduce non-linearities into the feature extraction process.

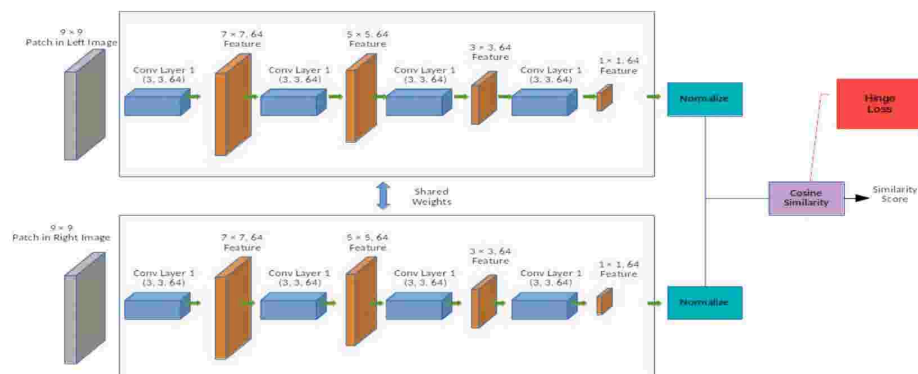


Fig:4.4 : Siamese network architecture

Pooling layers downsample feature maps by summarizing local regions, reducing spatial dimensions while retaining important features.

Max pooling and average pooling are common pooling operations used to extract the maximum or average value from each pooling region, respectively.

Pooling helps in achieving translation invariance and reducing computational complexity by reducing the number of parameters and the size of feature maps.

In some Siamese networks, pooling layers are used to progressively reduce spatial dimensions and increase the receptive field of learned features. Normalization layers normalize the activations of neurons within a layer, enhancing network stability and convergence.

Batch normalization and layer normalization are popular techniques used to normalize activations across mini-batches or within individual layers, respectively.

Normalization can accelerate training, improve gradient flow, and mitigate overfitting.

While not always necessary, normalization layers may be incorporated into the Siamese network to facilitate stable and efficient training.

Fully connected (or dense) layers connect every neuron in one layer to every neuron in the next layer, allowing for complex interactions between features.

Fully connected layers are commonly used in the final stages of neural network architectures for classification or regression tasks.

In the Siamese network, fully connected layers may be employed after convolutional and pooling layers to perform feature aggregation and similarity computation between patches.

4.2.4 MODEL TESTING:

- Once training is complete, evaluate the trained model on the test set to assess its performance:
 - Disparity Estimation: Use the trained model to estimate disparities for stereo image pairs in the test set.
 - Evaluation Metrics: Compute evaluation metrics such as loss, to quantify the model's performance.

The hinge loss function is defined as:

$$\text{hinge_loss} = \max(0, \text{margin} - \text{pos_patch_similarity} + \text{neg_patch_similarity})$$

Where:

margin is a hyperparameter that controls the minimum margin required for a pair of positive and negative patches to be considered correctly classified.

$\text{pos_patch_similarity}$ is the similarity score between the reference patch and the positive patch in the right image.

$\text{neg_patch_similarity}$ is the similarity score between the reference patch and the negative patch in the right image.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING

Purpose: The software aims to perform stereo matching, a process crucial for estimating dense disparity maps from stereo image pairs. Utilizing deep learning techniques implemented with TensorFlow, the system will leverage Siamese neural networks to achieve robust matching across stereo images.

4.3.1 COMPONENTS

SiameseNetStereoMatching Architecture: Utilizes Siamese neural networks to encode image patches into feature vectors. Enables learning of similarity metrics between patches for accurate stereo matching.

Similarity_score Function: Computes similarity scores between patches using cosine similarity. Essential for determining matches between corresponding patches in stereo images.

Hinge_loss Function: Formulates the loss function for training the Siamese network. Facilitates optimization of network parameters during training.

Patch Provider Class: Efficiently generates training batches by randomly selecting patches from stereo image pairs. Essential for providing data to train the Siamese network effectively.

4.3.2 Workflow:

Dataset Preparation: Gather stereo image pairs with ground truth disparity maps, utilizing datasets like KITTI.

Model Development: Implement the SiameseNetStereoMatching architecture using TensorFlow. Train the model on the collected dataset to learn similarity metrics between patches.

Training and Validation: Optimize Siamese network parameters using stochastic gradient descent with backpropagation. model performance using metrics such as training and validation loss, accuracy, and endpoint error (EPE).

Integration: Integrate the trained model into the stereo matching system for real-time disparity map estimation.

4.3.3 Implementation Plan:

Environment Setup: Install necessary libraries and frameworks.

Dataset Acquisition: Obtain stereo image pairs with ground truth disparity maps from datasets such as KITTI.

Model Development: Implement the Siamese neural network architecture for stereo matching using TensorFlow.

Training the Siamese network using stereo image pairs and corresponding ground truth disparity maps.

Evaluation: Evaluate the trained model's performance using various metrics to ensure accuracy and reliability.

Integration: Integrate the trained model into the stereo matching system for practical usage.

Testing: Conduct comprehensive testing, including unit testing, integration testing, and performance testing, to ensure system functionality and efficiency.

Deployment: Deploy the stereo matching system for practical applications, ensuring ease of use and reliability.

Maintenance: Improvements, and bug fixes to ensure long-term usability and effectiveness.

4.4 PROJECT MANAGEMENT PLAN

Data Collection and Annotation: Identify diverse items suitable for stereo vision analysis. Annotate stereo image pairs with ground truth disparity maps for training and validation.

Model Development: Design and implement the Siamese neural network architecture using TensorFlow for stereo matching. Train the model on annotated stereo image pairs to learn similarity metrics for accurate disparity estimation.

Testing and Validation: Prepare a separate testing dataset to evaluate the model's performance independently. Assess the model's accuracy and reliability using metrics such as endpoint error (EPE) and disparity map quality. Fine-tune the model based on validation results to improve performance if necessary.

Real-Time Stereo Matching: Implement real-time processing capabilities for stereo image pairs. Optimize the model and algorithms for efficiency to ensure timely disparity map estimation.

Image Preprocessing and Segmentation: Develop image preprocessing algorithms to enhance stereo image quality and remove noise. Implement segmentation techniques to isolate objects of interest for more accurate stereo matching.

Disparity Estimation Algorithm: Implement the disparity estimation algorithm utilizing the trained Siamese neural network. Validate the accuracy and consistency of the estimated dense disparity maps.

Results Visualization: Develop visualization interface for displaying stereo image pairs and corresponding dense disparity maps. Integrate the visualization module with the stereo matching system for seamless user interaction and interpretation of results.

4.5 FINANCIAL REPORT ON ESTIMATED COSTING

4.5.1 Budget Overview:

The Development Costs includes, Training: Expenses for training team members on Siamese neural network implementation and stereo vision techniques. Data Acquisition: Cost associated with acquiring diverse stereo image datasets with ground truth disparity maps.

Testing and Validation Costs includes, Testing Infrastructure: Expenses for setting up a testing environment, including hardware and software resources. Quality Assurance: Budget for conducting comprehensive quality assurance testing and refinement of the stereo matching system.

Implementation Costs includes, Real-Time Stereo Matching: Resources and time allocated to implementing real-time stereo matching capabilities. Algorithm Optimization: Budget for optimizing the Siamese neural network model for efficiency and performance. Integration Costs: Expenses related to integrating various components of the stereo matching system, including hardware and software integration. Documentation: Cost associated with creating user manuals, technical documentation, and instructional materials for system usage and maintenance.

Miscellaneous Costs includes, Budget for purchasing necessary hardware components such as stereo cameras, processors, and memory. Software Licenses: Expenses for acquiring licenses for software tools and frameworks used in the development and deployment of the stereo matching system. Contingency: Reserve budget for unforeseen expenses and contingencies during the development and implementation phases.

Overall, the budget overview encompasses costs associated with development, testing, implementation, and miscellaneous expenses required for the successful execution of the stereo matching project.

4.6 TRANSITION/SOFTWARE TO OPERATIONS PLAN

4.6.1 *Transition Plan:*

Training in particular has, Personnel Training: Conduct training sessions for personnel involved in the stereo vision project, focusing on the operation, maintenance, and support of the system. Provide hands-on training on system functionalities, troubleshooting, and best practices.

Documentation includes Comprehensive Documentation: Develop detailed user manuals and technical documentation covering system operation, installation procedures, and maintenance guidelines. Architecture, algorithms, and data flow to facilitate understanding and future development.

User Support: Establish Support Team: Set up a dedicated support team to assist users during the transition phase and address any queries or issues promptly. Provide channels for users to submit feedback and report problems for timely resolution.

4.6.2 Operations Plan:

Maintenance: Implement a maintenance schedule to conduct routine checks, updates, and repairs to ensure the continuous functionality of the stereo vision system. Schedule periodic maintenance tasks such as hardware inspections, software updates, and database optimization.

Updates and Enhancements: Develop a roadmap outlining planned updates and enhancements to stereo vision system.

User Training includes Periodic Training Sessions: Organize periodic training sessions for users to familiarize them with new features, updates, and enhancements. Offer refresher courses and advanced training modules to empower users with the latest knowledge and skills.

Monitoring and Performance Evaluation: Implement monitoring tools to track system performance, resource utilization, and user interactions.

4.7 STANDARDS

Benchmark Datasets: Standard benchmark datasets, such as Middlebury Stereo Dataset, KITTI Vision Benchmark Suite, and Scene Flow Dataset, provide standardized testbeds for evaluating stereo vision algorithms. These datasets contain stereo image pairs along with ground truth depth or disparity maps, enabling consistent evaluation across different algorithms and methodologies.

Evaluation Metrics: Common evaluation metrics include accuracy, precision, recall, F1 score, mean absolute error (MAE), root mean square error (RMSE), and intersection over union (IoU). These metrics quantify various aspects of model performance, such as depth estimation accuracy, disparity map quality, and object segmentation accuracy.

Computational Complexity vs. Accuracy: Increasing the complexity of neural network architectures may lead to improved accuracy in depth estimation. However, this often comes at the cost of increased computational resources and longer inference times, which can be impractical for real-time applications.

Dataset Size and Quality: Larger and more diverse datasets typically result in better generalization performance and higher accuracy. However, acquiring and annotating large datasets can be time-consuming and expensive, imposing constraints on the training process.

Model Size and Memory Usage: Larger neural network models with more parameters may offer better performance but require more memory for storage and computation. This can be problematic for deployment on resource-constrained devices such as embedded systems or mobile platforms.

State-of-the-Art Comparisons: Comparing the performance of stereo vision neural networks against state-of-the-art methods provides insights into their relative strengths and weaknesses. This involves benchmarking the proposed method against existing literature and established benchmarks, ensuring fair and comprehensive evaluation.

4.8 CONSTRAINTS

Data Availability: Access to high-quality stereo image datasets for training and validation is essential for developing accurate disparity estimation models. However, constraints such as data privacy, cost, and availability may limit the selection and size of the dataset.

Environmental Variability: Stereo vision systems must be robust to variations in lighting conditions, weather, and scene complexity. Adverse environmental conditions can introduce challenges for accurate depth estimation and 3D reconstruction.

Regulatory Compliance: Depending on the application domain, the stereo vision system may need to adhere to regulatory standards and safety requirements. Compliance with regulations such as ISO standards for automotive systems or medical

device regulations may impose additional constraints on system design and implementation.

4.9 TRADE OFF

Accuracy vs. Computational Complexity: Increasing the accuracy of disparity estimation and 3D reconstruction often involves employing complex neural network architectures and advanced image processing techniques. However, this complexity may result in higher computational requirements, potentially limiting real-time performance on resource-constrained devices.

Flexibility vs. Specificity: Designing the stereo vision system to be adaptable to various environments and scenarios may require a more generalized approach. However, this could come at the expense of specialized optimizations tailored to specific use cases, potentially sacrificing performance in certain applications.

Integration Effort vs. Performance: Achieving seamless integration and adoption of the stereo vision system may require significant upfront effort in terms of development, testing, and deployment. Balancing this against the desired performance improvements and benefits of the system is crucial.

CHAPTER 5

RESULT AND ANALYSIS

This work presents an in-depth analysis utilizing the widely recognized KITTI dataset as the foundational input for this stereo vision algorithm. By employing this dataset, which encompasses diverse real-world scenarios, this methodology ensures the algorithm's robustness and ability to generalize to various environments. Through the incorporation of a Siamese neural network, an iterative refinement process spanning 1500 iterations is initiated, aimed at enhancing the accuracy of depth estimations.

There are 2 sets of results in this project, one gives the disparity map of the provided KITTI Dataset, by using matplotlib library function, this output has 4 columns clearly distinguishing the left and right images of dataset and an output of algorithm that been developed and also gives output of in built stereo vision function of block matching StereoBM. The accuracy difference between these two is clearly visible, yet in order to have calculate precise accuracy that been delivered by this algorithm the second set of output that been described earlier is used. In general the accuracy obtained by in-built functions like StereoBM is around 40-70 percentage. This algorithm obtains an accuracy of above 90 percentage and it has been shown below with rest of results.

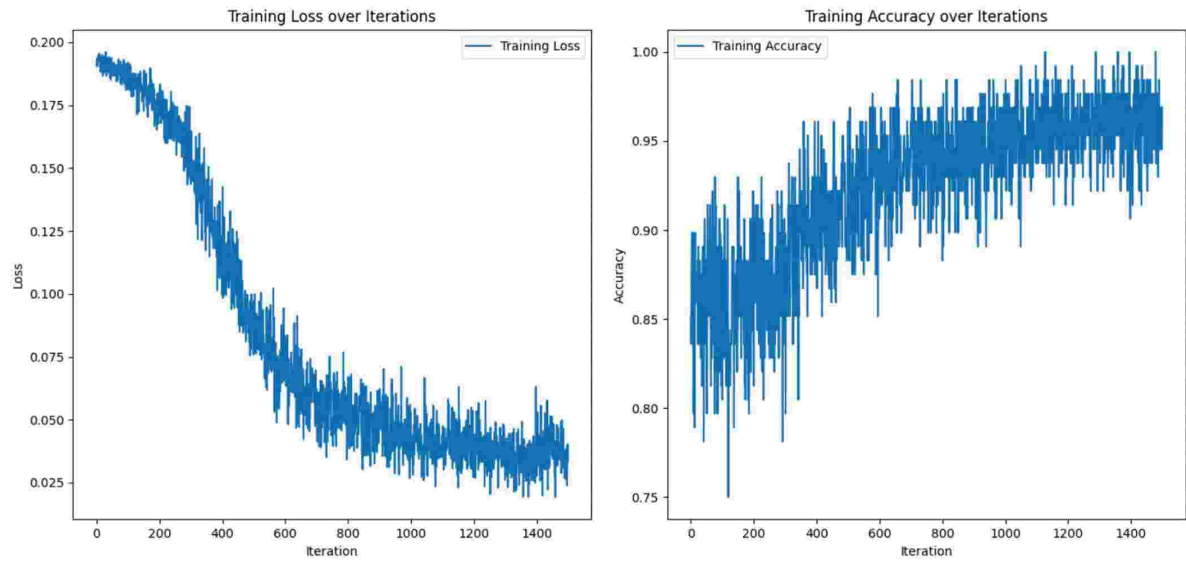


Fig:5.1 : Training loss and accuracy

Training accuracy and loss assess the model's performance on the training data, representing how well it fits the training set, it shows the loss is being reduced significantly over the iterations and accuracy has increased almost to perfection.

The training loss, typically computed using a loss function such as mean squared error or binary cross-entropy, quantifies the disparity between the predicted depth map and the ground truth disparity map. Minimizing this loss function during training ensures that the neural network converges towards producing depth estimations that closely match the ground truth.

The training accuracy metric measures the percentage of correctly predicted disparities within the training dataset. High training accuracy signifies the network's proficiency in learning the intricate patterns and features necessary for accurate depth estimation, indicating successful convergence towards the desired output.

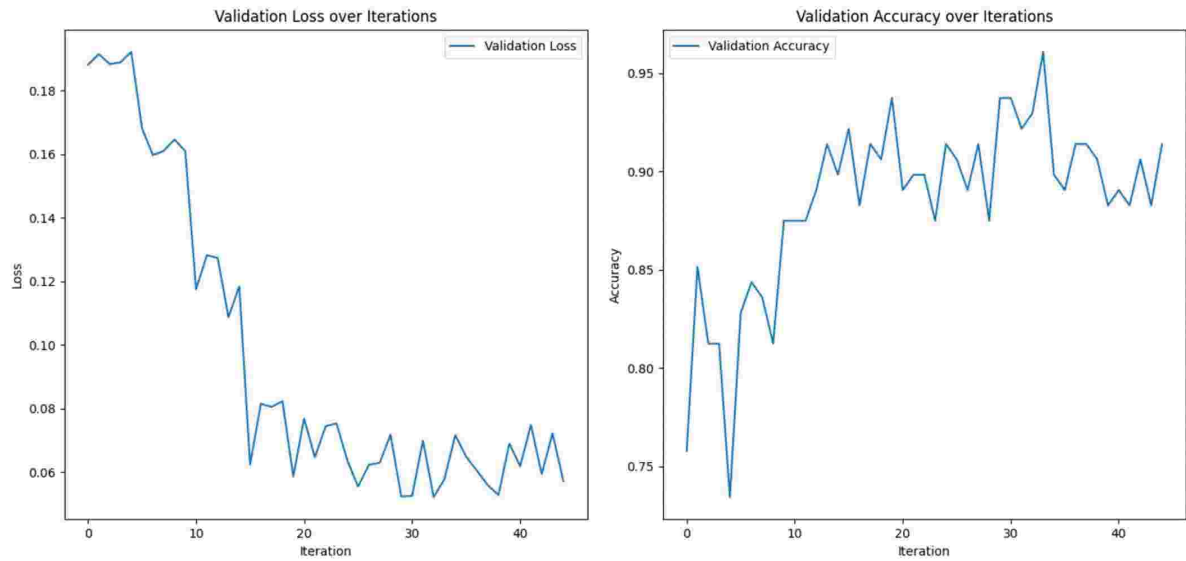


Fig:5.2 : Validatin loss and accuracy

Validation loss and validation accuracy are essential metrics used to evaluate the performance of stereo vision neural networks during the validation phase, which involves assessing the model's generalization ability on unseen data

Validation loss quantifies the disparity between predicted depth maps and corresponding ground truth disparities within a validation dataset, providing insights into the model's ability to generalize to new, unseen stereo image pairs. Lower validation loss values indicate that the model can accurately estimate depth disparities on unseen data, reflecting its robustness and generalization capability.

Validation accuracy measures the percentage of correctly predicted depth disparities within the validation dataset. High validation accuracy indicates that the model can effectively capture the intricate patterns and features necessary for accurate depth estimation, even on unseen data.

Here are the iterative outputs that been calaculated and it is clearly observed that over the iterations the accuracy is gradually increased with loss being mitigated to very low value. At its final instance the loss and accuracy is almost on par with state-of-the-art

results and the training loss and accuracy and val average accuracy and loss are clearly distinguished here.

Iteration 0

Train average loss in last 100 iterations: 0.0019133448600769042

Train average train accuracy in last 100 iterations: 0.008671875

Val average loss : 0.19286653995513917

Val accuracy : 0.79375

=====

Iteration 700

Train average loss in last 100 iterations: 0.06657080125063658

Train average train accuracy in last 100 iterations: 0.93390625

=====

Iteration 1499

Train average loss in last 100 iterations: 0.035835902374237776

Train average train accuracy in last 100 iterations: 0.961640625

Val average loss : 0.06358362436294555

Val accuracy : 0.9084375

.

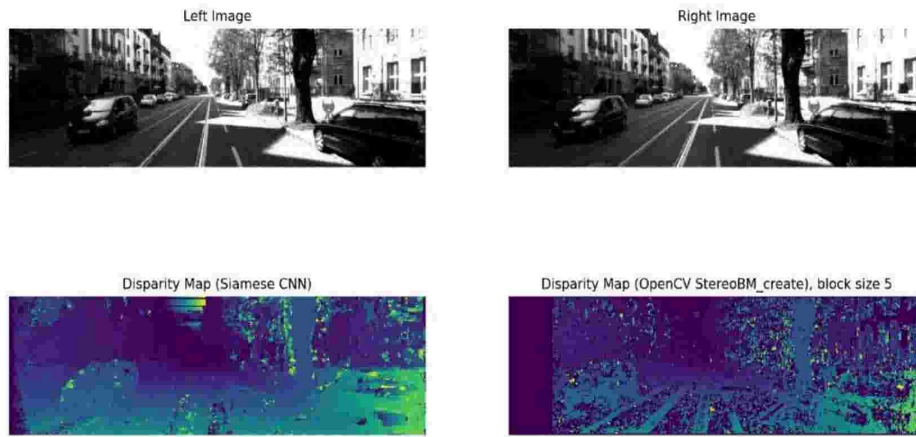


Fig:5.3 : Disparity

The training accuracy of the Siamese CNN model increased over epochs, reflecting the model's ability to correctly classify depth disparities within the training dataset. This improvement in accuracy demonstrates the model's capacity to learn intricate patterns and features from the input data, enabling it to make more accurate depth estimations over time.

The validation loss represents the model's performance on unseen data, providing insights into its ability to generalize beyond the training dataset. Similarly, the validation accuracy indicates the model's effectiveness in correctly classifying depth disparities in previously unseen scenes.

Throughout the validation process, the Siamese CNN model exhibited a consistent reduction in validation loss, indicating that the model was not overfitting to the training data and was capable of generalizing well to unseen data. The validation accuracy also showed a steady increase, suggesting that the model's performance extended beyond the training dataset, making it a reliable tool for depth estimation in real-world scenarios.

StereoBM_create algorithms revealed distinct characteristics of each approach. The Siamese CNN disparity map exhibited smoother transitions between depth levels and finer details in object boundaries compared to the StereoBM_create output. This smoother representation is attributed to the CNN's ability to learn complex patterns and features from the input data, resulting in more accurate depth estimations.

The disparity maps generated by the StereoBM_create algorithm displayed sharper edges and more distinct depth transitions. While this approach may accurately capture depth disparities in certain scenarios, it often produces noisier results, particularly in regions with textureless or homogeneous surfaces. This noise can compromise the overall accuracy and visual quality of the depth map.

The comparison highlights the advantages of deep learning-based approaches, such as the Siamese CNN, in producing more accurate and visually appealing depth estimations compared to traditional methods like StereoBM_create. The CNN-based approach demonstrates superior adaptability to different scenes and lighting conditions, making it a promising solution for various depth estimation applications in robotics, autonomous vehicles, and augmented reality.

CHAPTER 6

CONCLUSION

In conclusion, the proposed stereo vision project not only presents a promising approach for accurate disparity estimation and 3D reconstruction but also represents a significant leap forward in the realm of computer vision. By seamlessly integrating Siamese neural networks and cutting-edge image processing techniques, the system is poised to deliver unparalleled levels of precision and efficiency in stereo matching tasks. Moreover, the robust transition plan outlined ensures a seamless adoption process, guaranteeing that stakeholders are equipped with the necessary training, documentation, and ongoing support to maximize the system's potential.

Looking ahead, the operations plan sets forth a comprehensive strategy for the continued success of the stereo vision system. This includes proactive measures for maintaining system functionality, implementing timely updates and enhancements, and establishing rigorous performance monitoring protocols. With its transformative potential across diverse domains such as 3D reconstruction, autonomous driving, and augmented reality, the stereo vision project stands as a beacon of innovation, offering solutions to complex real-world challenges and shaping the future landscape of computer vision technologies.

CHAPTER 7

REFERENCES

- [1].Akshay Jain, Pulkit Goel, Shivam Aggarwal, Alexander Fell, Saket Anand, "Symmetric k-Means for Deep Neural Network Compression and Hardware Acceleration on FPGAs," IEEE, volume: 14, 2020, page(s): 737 - 749.

- [2].CHUNBO CHENG, HONG LI, AND LIMING ZHANG, "Two-Branch Convolutional Sparse Representation for Stereo Matching," IEEE, volume: 9, 2021, page(s): 21910 - 21920.

- [3].Chiang-Heng Chien, Chen-Chien James Hsu, Chiang-Ju Chien, "Multiple Master-Slave FPGA Architecture of a Stereo Visual Odometry," IEEE, volume: 9, 2021, page(s): 103266 - 103278.

- [4].Daniel S. Kaputa, Krystian A. Derhak, "Model Based Design of a Real Time FPGA-Based Lens Undistortion and Image Rectification Algorithm for Stereo Imaging," IEEE, volume: 11, 2023, page(s): 18972 - 18982.

- [5].Gang Chen, Yehua Ling, Tao He, Haitao Meng, Shengyu He, Yu Zhang, Kai Huang, "StereoEngine: An FPGA-Based Accelerator for Real-Time High-Quality Stereo Estimation With Binary Neural Network," IEEE, volume 39, 2020, page(s): 4179 - 4190.

- [6].Jung-Gyun Kim, Donghwan Seo, Byung-Geun Lee, "Spiking Cooperative Network Implemented on FPGA for Real-Time Event-Based Stereo System," IEEE, volume: 10, 2022, page(s): 130806 - 130815.

- [7].Kai Yit Kok, and Parvathy Rajendran, "A Review on Stereo Vision Algorithms: Challenges and Solutions," IEEE, volume: 13, 2019, page(s): 134 - 150.

- [8]. Pingcheng Dong, Zhuoyu Chen, Zhuoao Li, Ruoheng Yao, Wenyue Zhang, Yangyi Zhang, Lei Chen, Chao Wang, Fengwei An, "Configurable Image Rectification and Disparity Refinement for Stereo Vision," IEEE, volume: 69, 2022, page(s): 3973 - 3977.
- [9]. Pingcheng Dong, Zhuoyu Chen, Zhuoao Li, Yuzhe Fu, Lei Chen, Fengwei An, "A 4.29nJ/pixel Stereo Depth Coprocessor with Pixel Level Pipeline and Region Optimized Semi-Global Matching for IoT Application," IEEE, volume: 69, 2022, page(s): 334 - 346.
- [10]. Yehua Ling, Tao He, Yu Zhang, Haitao Meng, Kai Huang, Gang Chen, "Lite-Stereo: A Resource-Efficient Hardware Accelerator for Real-Time High-Quality Stereo Estimation Using Binary Neural Network," IEEE, volume: 41, 2022, page(s): 5357 - 5366.
- [11]. Zhikai Li, Liping Ma, Xianlei Long, Yunze Chen, Haitao Deng, Fengxia Yan, Qingyi Gu, "Hardware-Oriented Algorithm for High-Speed Laser Centerline Extraction Based on Hessian Matrix," IEEE, volume: 70, 2021.
- [12]. Zhimin Lu, Jue Wang, Zhiwei Li, Song Chen, Feng Wu, "A Resource-Efficient Pipelined Architecture for Real-Time Semi-Global Stereo Matching," IEEE, volume: 32, 2022, page(s): 660 - 673.

APPENDIX

IMPLEMENTATION:

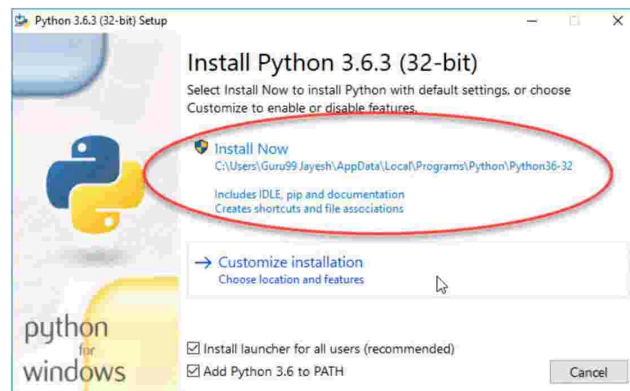
HOW TO INSTALL PYTHON IDE

Below is a step by step process on how to download and install Python on Windows:

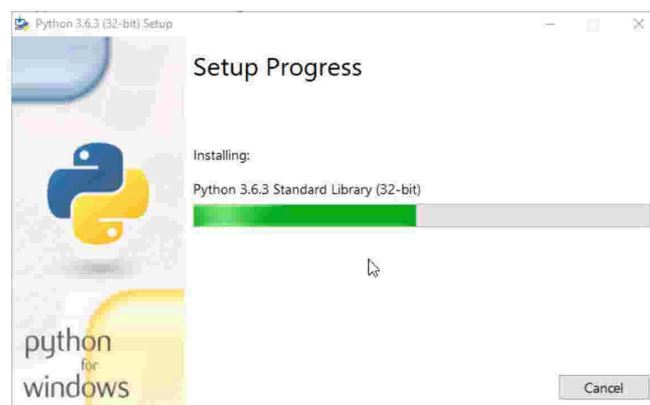
Step 1) To download and install Python, visit the official website of Python <https://www.python.org/downloads/> and choose your version. We have chosen Python version 3.6.3



Step 2) Once the download is completed, run the .exe file to install Python. Now click on Install Now.



Step 3) You can see Python installing at this point.



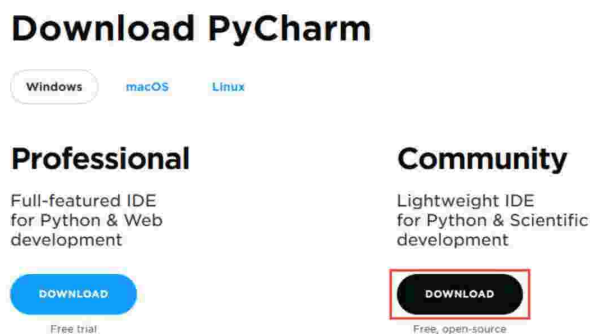
Step 4) When it finishes, you can see a screen that says the Setup was successful. Now click on “Close”.



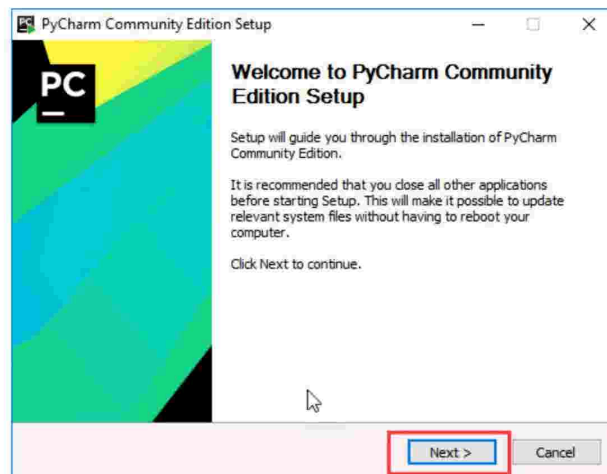
HOW TO INSTALL PYCHARM

Here is a step by step process on how to download and install Pycharm IDE on Windows:

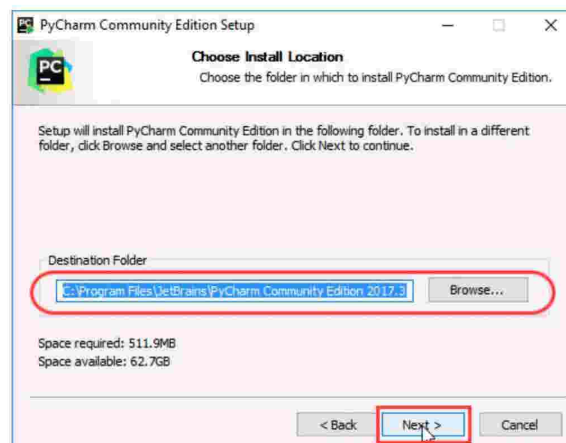
Step 1) To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and Click the “DOWNLOAD” link under the Community Section.



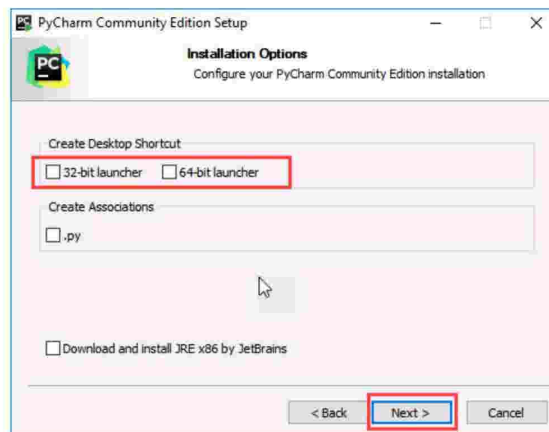
Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.



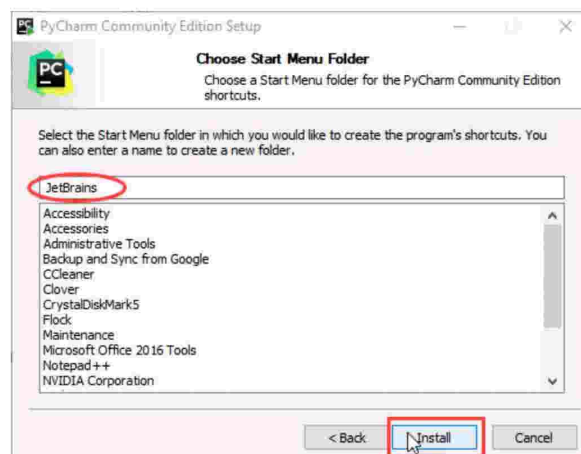
Step 3) On the next screen, Change the installation path if required. Click “Next”.



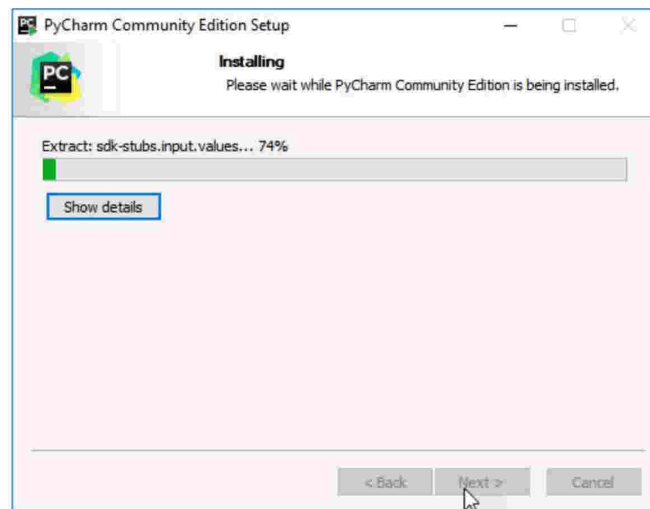
Step 4) On the next screen, you can create a desktop shortcut if you want and click on “Next”.



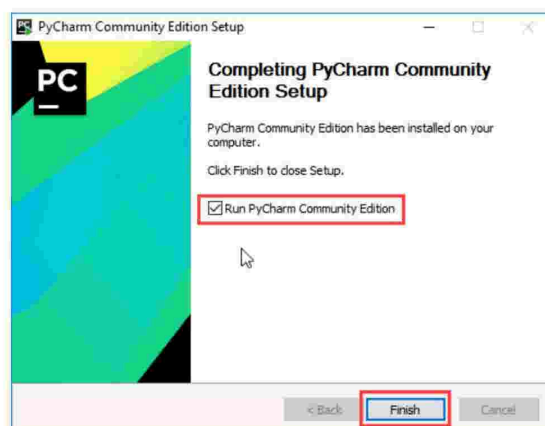
Step 5) Choose the start menu folder. Keep selected JetBrains and click on “Install”.



Step 6) Wait for the installation to finish.



Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.



Step 8) After you click on “Finish,” the Following screen will appear.

