Kaunas University of Technology

Faculty of Mechanical Engineering and Design

Automated Visual Inspection System Based on Stereo Vision

Master's Final Degree Project

Ahmed Ibrahim Mosad Hassan Elatroush

Project author

Researcher Valdas Grigaliunas

Supervisor

Kaunas, 2021

Kaunas University of Technology

Faculty of Mechanical Engineering and Design

Automated Visual Inspection System Based on Stereo Vision

Master's Final Degree Project

Industrial Engineering and Management (6211EX018)

Ahmed Ibrahim Mosad Hassan Elatroush

Project author

Researcher Valdas Grigaliunas

Supervisor

Researcher Marius Gudauskis

Reviewer

Kaunas, 2021

Kaunas University of Technology

Faculty of Mechanical Engineering and Design

Ahmed Ibrahim Mosad Hassan Elatroush

Automated Visual Inspection System Based on Stereo Vision

Declaration of Academic Integrity

I confirm the following:

1.I have prepared the final degree project independently and honestly without any violations of the copyrights or other rights of others, following the provisions of the Law on Copyrights and Related Rights of the Republic of Lithuania, the Regulations on the Management and Transfer of Intellectual Property of Kaunas University of Technology (hereinafter – University) and the ethical requirements stipulated by the Code of Academic Ethics of the University;

2.All the data and research results provided in the final degree project are correct and obtained legally; none of the parts of this project are plagiarised from any printed or electronic sources; all the quotations and references provided in the text of the final degree project are indicated in the list of references;

3.I have not paid anyone any monetary funds for the final degree project or the parts thereof unless required by the law;

4.I understand that in the case of any discovery of the fact of dishonesty or violation of any rights of others, the academic penalties will be imposed on me under the procedure applied at the University; I will be expelled from the University and my final degree project can be submitted to the Office of the Ombudsperson for Academic Ethics and Procedures in the examination of a possible violation of academic ethics.

Ahmed Ibrahim Mosad Hassan Elatroush

Confirmed electronically

Kaunas University of Technology

Faculty of Mechanical Engineering and Design

Task of the Master's final degree project

Given to the student – Ahmed Ibrahim Mosad Hassan Elatroush

1. Title of the project

Automated Visual Inspection System Based on Stereo Vision

(In English)

Automatinės vizualinės apžiūros sistemos, pagrįstos stereo vizija, projektavimas

(In Lithuanian)

2. Aim and tasks of the project

Aim: To design and evaluate a stereo vision-based system for accurate depth measurement of precisely fitted electronic elements (holtite sockets).

Tasks:

1. Implement a deep learning technique using object detection to differentiate and localize both holtite sockets and PCB holes.
2. Design a mechanical setup for implementing a stereo vision-based depth measurement system.
3. Develop a stereo vision system to measure the depth of the pressed holtite sockets into their corresponding holes.

3. Initial data of the project

The fitted socket has an allowable elevation tolerance from the PCB hole's surface of ± 100 μm

4. Main requirements and conditions

AI development board (Nvidia Jetson Nano A02).
Cameras that support interchangeable lenses.
Opensource platforms for developing the code in python programming language.

1

Project author	Ahmed Ibrahim Mosad Hassan Elatroush		
	(Name, Surname)	*(Signature)*	*(Date)*
Supervisor	Valdas Grigaliunas		
	(Name, Surname)	*(Signature)*	*(Date)*
Head of study field programs	Regita Bendikienė		
	(Name, Surname)	*(Signature)*	*(Date)*

Elatroush, Ahmed Ibrahim Mosad Hassan. Automated Visual Inspection System Based on Stereo Vision. Master's Final Degree Project, supervisor Researcher Valdas Grigaliunas; Faculty of Mechanical Engineering and Design, Kaunas University of Technology.

Study field and area (study field group): Production and Manufacturing Engineering (E10), Engineering Sciences (E).

Keywords: Stereo vision, Deep learning, Object detection, Automated visual inspection, Quality control.

Kaunas, 2021. Number of pages 78.

Summary

An automated visual inspection system incorporating computer vision was investigated, and a prototype for a proof of concept was developed in this final degree project. The extensive usage of deep learning in the 4th industrial revolution was the inspiring factor for implementing it as part of the system. A traditional computer vision method known as stereovision integrated with deep learning-based artificial intelligence to simplify the algorithm used for depth measurement. This technique can be used widely in various inspection systems in different industries. The industry of focus is the electronics industry, where precise small components have their depths of fit being measured. The proposed system is cost-effective. The deep learning-based neural network was deployed on the Nvidia Jetson Nano board, which operates on the Linux OS. Python programming language was used to develop the code, and an x-y translation stage was integrated into this prototype. The system incorporated the newly released Raspberry Pi HQ cameras that support interchangeable lenses, allowing for the mounting microscopic lenses.

Elatroush, Ahmed Ibrahim Mosad Hassan. Automatinės vizualinės apžiūros sistemos, pagrįstos stereo vizija, projektavimas. Magistro baigiamasis projektas, vadovas mokslo darbuotojas Valdas Grigaliunas

; Kauno technologijos universitetas, Mechanikos inžinerijos ir dizaino fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Gamybos inžinerija (E10), Inžinerijos mokslai (E).

Reikšminiai žodžiai: Stereo vizija, Gilusis mokymasis, Objektų aptikimas, Automatizuota vizualinė apžiūra, Kokybės kontrolė.

Kaunas, 2021. Puslapių sk. P 78.

Santrauka

Šiame darbe buvo nagrinėjama automatizuota objektų atstumo, statmena paviršiui kryptimi, matavimo sistema, pagrįsta kompiuterine rega. Taip pat sukurtas koncepcinis prototipas. Platus gilaus mokymosi panaudojimas 4-ojoje pramonės revoliucijoje buvo įkvepiantis veiksnys įgyvendinant tai kaip sistemos dalį. Tradicinis kompiuterinės regos metodas, žinomas kaip stereo-vizija, buvo integruojamas kartu su Giliuoju mokymusi grįstu dirbtiniu intelektu, siekiant supaprastinti gylio/atstumo matavimui naudojamą algoritmą. Šio tipo sistemos gali būti naudojamos įvairiose pramonės srityse. Darbe dėmesys skiriamas elektronikos pramonėje naudojamiems tiksliems, mažiems komponentams, kuriems reikia atlikti atstumo/gylio matavimus. Kuriama sistema yra pakankamai ekonomiška. Giliuoju mokymusi grįstas neuroninis tinklas buvo realizuotas naudojant „Nvidia Jetson Nano“ vystymo plokštę naudojant „Linux“ operacinę sistemą. Kuriant programą buvo naudojamas „Phython“ programavimo kalba. Sistemoje taip pat panaudotas jau sukurtas xy koordinatinis staliukas. Sistemoje panaudotos Raspberry Pi HQ kameros, su keičiamais lęšiais, kurios šiuo atveju yra integruojamos su mikroskopiniais lęšiais.

Table of contents

List of figures

List of tables

List of abbreviations and terms

Abbreviations:

CNN – convolutional neural networks

DNN – Deep Neural networks

DL– Deep learning

VI– visual inspection

AVI– automated visual inspection

CV – computer vision

FDM – Fused Deposition modelling

ONNX – Open Neural Network Exchange

PCB – printed circuit board

Introduction

Automated visual inspection systems have been developing, especially with the rising trend of artificial intelligence and, more specifically, deep learning. Traditional computer vision techniques for perceiving depth information about objects have been used in industries for quality inspection. However, some objects have various irregular shapes, sizes, and colour intensities and have difficulties extracting their features accurately through traditional image processing techniques. Therefore, deep learning-based computer vision can be integrated with the traditional means to extract features having variances and thus improve the accuracy of depth measurement performed by traditional computer vision techniques. An application where the integration of both techniques becomes very useful is the inspection of pressed holtite sockets into PCB holes. The holtite sockets are pressed either manually or using a machine into corresponding PCB holes. The sockets need to be inspected visually to detect defects and, more importantly, to decide whether the depth of which the socket is pressed into the hole complies with the manufacturer's tolerances or not. The components are minute in size, and the depth into which they are pressed into the PCB holes is usually in the micrometres range. Furthermore, according to the manufacturer, the traditional method of inspection is using human-based visual inspection. Therefore, proposing an automated visual inspection system that can accurately measure the depth of the pressed socket into the holes would be an effective, more accurate, and time-saving solution.

Aim:

To design and evaluate a stereo vision-based system for accurate depth measurement of precisely fitted electronic elements (holtite sockets).

Tasks:

1. Implement a deep learning technique using object detection to differentiate and localize both holtite sockets and PCB holes.
2. Design a mechanical setup for implementing a stereo vision-based depth measurement system.
3. Develop a stereo vision system to measure the depth of the pressed holtite sockets into their corresponding holes.

Initial data of the project:

The fitted socket has an allowable elevation tolerance from the PCB hole's surface of ± 100 μm.

1. Literature review

1.1.Manual visual inspection

Visual inspection is simply the use of the naked eye to investigate equipment and its working conditions. Where flaws, defects, or any issues can cause the equipment to malfunction. However, precise measurements cannot be performed by manual inspection [1]. VI is considered a method of quality control that would be implemented for processes, products. Moreover, human-based VI was considered reliable during the 20th century. During the 1950s, the human factor was realized to be the weakest link in the quality control process [2].

Errors arising from manual visual inspection vary depending on various factors and also differ with different industrial sectors. Some of the technical factors affecting the error are the defect types, standards used, skilfulness of the inspector. Moreover, psychophysical factors such as age, experience, creativity, organizational factors such as training, clarity of instructions influence the inspection error [2 p. 4]. Furthermore, social factors play a vital role during crises like global pandemics (covid-19), placing additional pressure on inspectors. Time is another vital factor to consider for large production lines. For complex parts that need a thorough inspection, the process can take a considerable amount of time. Thus, to achieve more reliable and accurate VI, the mentioned factors should be evaluated and optimized by the operating companies. Optimizing the factors involves additional costs and resource allocation.

1.2.Automated visual inspection

Automated visual inspection replaces the human factor and uses machines incorporating computer vision to perform inspection tasks. CV systems are constructed from cameras and computer processors. The cameras mimic the human eye to observe the scene, and the processors mimic the human brain to process observations[3]. Research and development of AVI systems started back in the 1980s, and the field has been expanding tremendously since then [2]

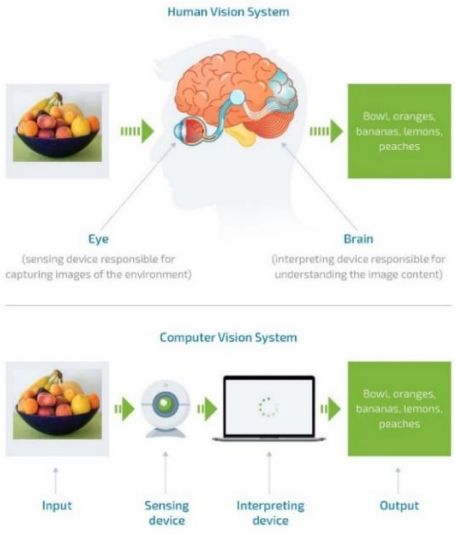


Fig. 1. Analogy between human and computer vision [3]

The Advantages of AVI in comparison to manual VI are summarized below [3]:

– Suitable for precise measurements, unlike manual VI

– Faster, objective inspection
– Inspected data can be stored and is thus traceable
– Real-time feedback is crucial for process improvement
– Ability to deploy in hazardous environments

1.2.1. Traditional computer vision

Traditional CV systems work in the following way:

4. Collect an image database of the object to be inspected.
5. An engineer manually extracts the features of interest for each object that the model needs to learn.
6. Those extracted features are then fed into a machine-learning algorithm that can classify and detect objects based on the extracted features.

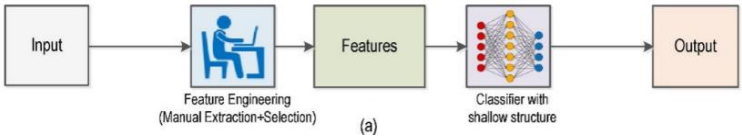


Fig. 2. A traditional CV system's workflow [4]

Thus CV can recognize patterns, shapes, edges, colours, sizes. In other words, the features that were identified initially by the engineer during the system setup and programming phase. After that, the system can operate in an automated manner. Famous applications in the industry are barcode reading and identification, identifying the presence and absence of objects, robotic guidance, and making 3D measurements of objects inspected. The main drawback of a traditional CV system is manually extracting the features needed to identify objects. Suppose the use of CV for inspection in the electronics industry is considered. Then for detecting defects, the engineer would have to extract and define all features that define the defect by using appropriate image processing techniques for detecting defects. For example, edge detection techniques are used for detecting edges. Hough transform is a technique used to detect circles. Other techniques are used for detecting size, colour, but the engineer needs to define the features and corresponding techniques himself in the programming phase while setting up the system. Therefore, the drawbacks of manual feature extraction and thus traditional CV systems would be :

– Time-consuming.
– Requires skilled engineers.
– Personal training by engineers can induce errors in the detection algorithm.

1.2.2. Deep learning-based computer vision

Deep learning is a subcategory of machine learning, which is a subcategory of artificial intelligence. Deep learning utilizes deep neural networks, multi-layered artificial neural networks that imitate the human's methodology of processing information. The Deep Neural Networks attempt to reproduce the way the human neurons function. A DNN processes information by identifying patterns, relations, and classifications of different types of information [5].

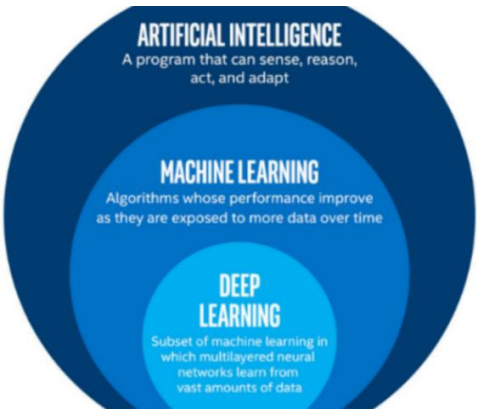


Fig. 3. Deep Learning, machine learning, and artificial intelligence [5]

Concerning computer vision, the real benefit of deep learning lies in overcoming the biggest drawback of traditional CV, which is the manual extraction of features. In deep learning, the whole classification process occurs from A to Z through the DNN, as shown inFig. 4. It extracts and the necessary features during the training process from the provided dataset, then the extracted features are processed in a similar way humans use neurons to process information. The feature extraction is achieved through artificial neural networks, which are the basis on which DL is built. Considering a car as an example, a traditional CV system would take the input as an image of a car, and then human intervention is needed to identify all the features that define a car for the system. This process is executed for every image in the training dataset. However, with deep learning, only images of cars are used as input. The DNN then identifies and extracts all features that define a car. Therefore, human intervention is eliminated.

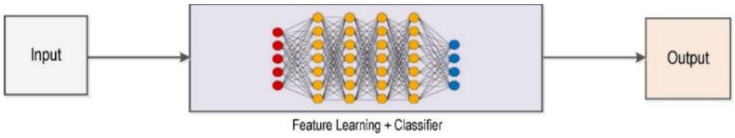


Fig. 4. Workflow of a deep learning-based CV system [4]

An essential criterion in comparing traditional CV with deep learning-based CV is performance. Deep learning models outperform traditional techniques. In fact, with robust training of the model through increasing the amount of training data, the performance of the DNN can increase linearly. Whereas traditional CV algorithms reach a saturation point after a certain amount of data, this is illustrated in Fig. 5. Therefore, DNNs improve quality control from an industrial perspective, and thus, the industry giants are incorporating robust DL techniques for controlling the quality of their products.

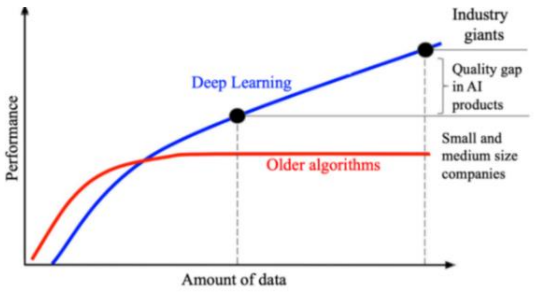


Fig. 5. Comparing the performance of DL vs traditional CV techniques [6]

Other manufacturing-related advantages of using DL in CV applications would be [6] :

- Lower operation costs (due to the elimination of the human factor in operating the system).
- Reduction in machine downtime, where no maintenance is required after setting up the model.
- Flexibility, adaptation to variations in consumer demands.
- Enhancement of productivity.
- Gain a competitive edge in respective markets.

DL-based CV systems show enormous potential for integration in the Internet of things systems, which drives the fourth industrial revolution. Listed below are some application of DL-based CV systems in Industry 4.0 [6] :

- Object detection in smart factories during packaging, assembling products.
- Detecting surface defects.
- Self-driving cars.
- Smart traffic light control.

1.3.Deep Neural Networks

DNN has the same structure as a simple ANN, with the difference being that it uses multiple hidden layers, as shown in Fig. 6. The layers increase the network's complexity and accuracy in making predictions.

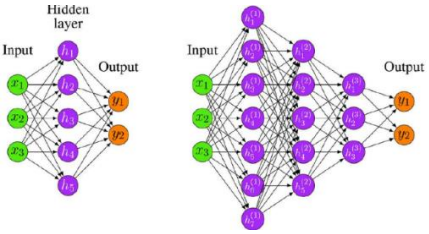


Fig. 6. ANN (left) vs DNN (right) [7]

Each layer in a DNN consists of multiple neurons connected to other neurons in the following and preceding layers to constitute the network. A simple example of what occurs at one neuron is shown below in fig Fig. 7. Input data is fed into neurons of hidden layers with a set of coefficients known as weights. Weights assign the significance of inputs concerning the task that the network is aiming to learn. The inputs are multiplied by their weights, and the weighted sum is computed where a bias is also added. The sum is then passed through the neuron's activation function, determining whether or not to activate this neuron. The activation function thus produces an output from the calculation. This output is then fed as input to other nodes in the following layers in the network, where a relationship may exist between those nodes. This way, deep neural networks can be used for complex tasks where lots of dependent features are linked and aggregated instead of simple ANNs, which only have one hidden layer.

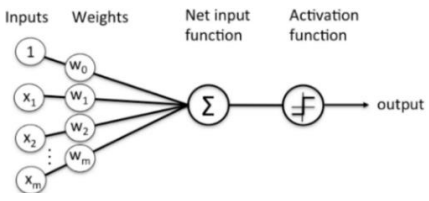


Fig. 7. Neuron activation in an artificial neural network [8]

The neuron's input function, onto which the activation function is applied, is shown in equation 1.1 below:

$$z = \sum_{i=1}^{n} x_i w_i + b \tag{1.1}$$

DNN's are feedforward networks, where feedforward networks are the simplest type of neural network. In Feedforward networks, information flows directly from the input layers through the network's hidden layers to produce an output layer. There are no feedback connections into which outputs of the model are fed back [7].

1.3.1. Loss function

Training a neural network in general, whether it is a simple ANN or a sophisticated DNN, contributes significantly to the accuracy of the DNN. The purpose of a good training algorithm is to feed the DNN with enough information such that it can later detect that information on its own. To contextualize CV and AVI, generally, the more images a DNN is trained on representing a task-related scenario, the more accurate the network will be executing inspection tasks. There are two types of learning algorithms from which a DNN can learn, supervised and unsupervised learning. In supervised learning uses labelled images to train a DNN, whereas unsupervised learning does not. Labelled images identify the objects of interest and differentiate them from other objects or backgrounds in the image. Unsupervised learning uses images containing several different types of objects, and the algorithm allows the DNN to establish relations between objects of the same class and group them by extracting standard features between the objects. The difference between both learning methods is shown in Fig. 8.

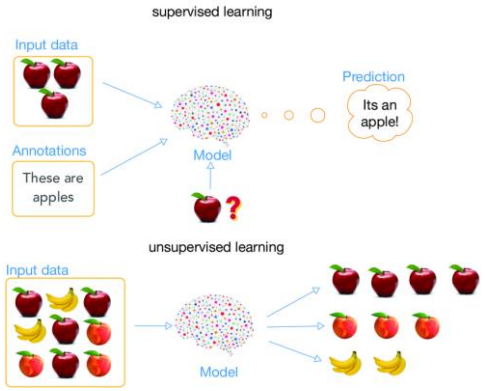


Fig. 8. Supervised vs unsupervised learning [9]

Only supervised learning will be considered in the thesis, and thus the unsupervised learning's working algorithm will not be reviewed. Before discussing how supervised learning works, first, the loss function is to be explained. A loss function calculates the loss in the algorithm's accuracy, that is, the error between the network's predicted output and the actual one. The loss function compares the predicted value of the neurons in the final layer of the DNN with the actual value for a single training data, for example, one image in the context of CV. Thus, the loss function calculates the error of the network. This process of calculating the loss function occurs for each training image used to build the network. The final error is then interpreted differently for different loss functions. For example, the means squared error is a function that squares all the errors of the training data and then calculates the mean of the total error. Moreover, the most common loss functions used in CV

applications are the mean squared error used in regression and the categorical cross-entropy loss function used for multi-class classification [10].

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y_i} - Y_i)^2 \tag{1.2}$$

Where :

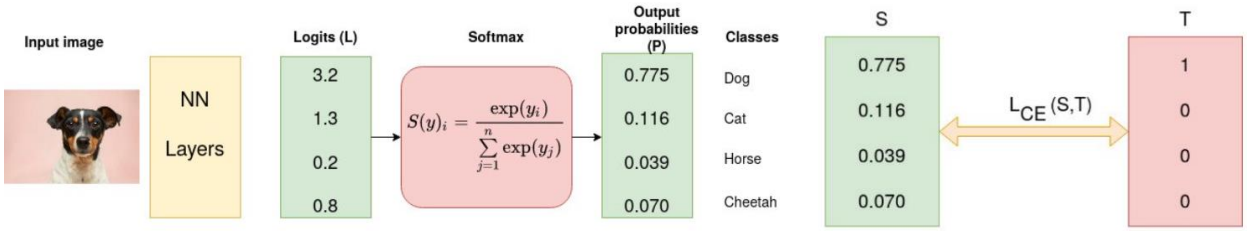
- $\hat{Y_i}$ is the vector denoting n number of prediction values representing the dataset.
- Y_i is the vector denoting n number of actual values representing the dataset.

The categorical cross-entropy function works by comparing the output probabilities for multi-class classification with the truth values. Considering the example in Fig. 9. The input image of a dog passes through a convolutional neural network which is a type of DNNs and is discussed in the following section 0. a SoftMax function which is an activation function, is used to convert logits into output probabilities, which are then compared to the true values as shown in the same figure. The categorical cross-entropy loss function is expressed as follows [11] :

$$L_{CCE} = -\sum_{i=1}^{n} t_i \log(p_i) \tag{1.3}$$

Where :

- n is the number of classes.
- t_i is the truth value (0 or 1).
- p_i is the softmax probability for the class of index i.



a) Output probabilities for classes and actual values.

b) CCE loss function comparing predicted

Fig. 9. Multi-class classification using the categorical cross-entropy loss function

1.3.2. Backpropagation and optimizers

The purpose of calculating the loss function is to reduce the error with each iterative process where a backpropagation algorithm is used to propagate backwards in the network to reduce the classification error of the network. Furthermore, backpropagation works by adjusting the weights of important neurons (activated ones) that detect relevant features for classifying objects within the acquired images. The loss function is recalculated after adjusting the weights, and the process is iterated. Thus the loss of the network reduces over time. Backpropagation works by calculating the gradient of the loss function with regards to the model's variables, which are the weights. The gradient shows how much the weights need to be changed to minimize the loss function. The chain rule is used to compute the gradients. The formulas and derivation for calculating the gradient using backpropagation are

shown below in Equations 1.4 – 1.7 [12]. where the gradient is only calculated for one weight here as an example denoted as w_{jk}^{l}. Where j is the index of the neurone in layer Land k is the index of the neuron in the preceding layer, layer L-1.

$$\frac{\partial L}{\partial w_{jk}^{l}} = \frac{\partial L}{\partial z_{j}^{l}} \frac{\partial z_{j}^{l}}{\partial w_{jk}^{l}} \tag{1.4}$$

$$z_{j}^{l} = \sum_{k=1}^{n} w_{jk}^{l} x_{k}^{l-1} + b_{j}^{l} \tag{1.5}$$

Where:

- $\frac{\partial L}{\partial w_{jk}^{l}}$ is the partial derivative of the loss function concerning the considered weight
- $\frac{\partial L}{\partial z_{j}^{l}}$ is the partial derivative of the loss function concerning the input to neuron at layer L
- z_{j}^{l} is the weighted sum of the product of the input neuron at layer L-1 and the considered weight and the chosen bias.

Thus, we can derive the gradient as follows:

$$\frac{\partial L}{\partial w_{jk}^{l}} = \frac{\partial L}{\partial z_{j}^{l}} x_{k}^{l-1} \tag{1.6}$$

The same set of equations is used to calculate the derivative of the loss function for the chosen bias [26].

$$\frac{\partial L}{\partial b_{j}^{l}} = \frac{\partial L}{\partial z_{j}^{l}} \tag{1.7}$$

1.3.3. Optimizers and hyperparameters

After using Backpropagation to calculate the gradient of the loss function concerning the weight, the weights need to be updated to reduce the loss function. Before explaining and comparing optimizers, it is important to define at first the hyperparameters of the network. The hyperparameters are parameters of the network that can be fine-tuned to improve the accuracy of the neural network and reduce the computational power and speed required by the network. The following list includes common hyperparameters used by all optimizers for training the network. Some optimizers include other hyperparameters related only to their optimization algorithms.

- Learning rate: the rate at which the neural network learns
- Number of epochs (iterations)
- Batch size: in the context of CV, the number of images grouped in a batch where one training iteration is performed on the batch, for datasets of large numbers of images, a batch size of 32 and more are used

Choosing the common hyperparameters can either be achieved manually through trial and error or Bayesian optimization. The optimization functions are reviewed in the following section

At first, according to [13], a review of two classification MNIST datasets CIFAR-10 using various optimizers is shown in Fig. 10 and Fig. 11. The MNIST dataset is a classification dataset for numbers, and CIFAR-10 is a dataset of 10 classes of animals and vehicles. According to the two datasets, the following three optimizers performed the best: SGD- Nesterov, Adam, and AdaMax, as they were the most accurate three optimizers.

	CNN-1		CNN-2		CNN-3	
Algorithm	**Test Loss**	**Test Acc.**	**Test Loss**	**Test Acc.**	**Test Loss**	**Test Acc.**
SGD	0.0468	99.00	0.0724	99.03	0.0780	99.24
SGD - momentum	0.0395	99.26	0.0633	99.18	0.0718	99.26
SGD - Nesterov	0.0366	99.34	0.0511	99.35	0.0589	99.41
AdaGrad	0.0600	98.57	0.0753	98.79	0.0730	99.08
AdaDelta	0.0306	**99.50**	0.0395	99.49	0.0460	99.40
RMSProp	0.0505	99.26	0.1899	98.70	0.1108	99.32
Adam	0.0425	99.35	0.0597	99.00	0.0536	99.29
AdaMax	0.0337	99.37	0.0418	**99.51**	0.0454	**99.42**
Nadam	0.0364	99.32	0.0567	99.19	0.0565	99.16
AMSGrad	0.0401	99.24	0.0561	99.28	0.0497	99.36

Fig. 10. Comparing classification accuracy between optimizers for the MNIST dataset

	CNN-1		CNN-2		CNN-3	
Algorithm	**Test Loss**	**Test Acc.**	**Test Loss**	**Test Acc.**	**Test Loss**	**Test Acc.**
SGD	0.9428	67.98	0.8899	70.84	1.0112	68.19
SGD - momentum	1.2479	74.96	1.3368	76.74	1.1978	78.21
SGD - Nesterov	1.2235	76.01	1.2909	77.96	1.1651	79.97
AdaGrad	1.2428	56.62	1.1769	59.61	1.1531	60.55
AdaDelta	1.6619	74.08	1.7222	76.23	1.4789	78.55
RMSProp	1.2670	75.01	1.1354	76.14	0.9821	74.81
Adam	1.2279	76.05	1.1988	76.94	0.9393	79.03
AdaMax	0.8054	**76.89**	1.1240	**79.17**	1.0437	**81.41**
Nadam	1.2735	76.11	1.3102	76.60	1.0672	78.98
AMSGrad	1.1940	76.45	1.2000	77.11	1.0135	78.27

Fig. 11. Comparing classification accuracy between optimizers for the CIFAR-10 dataset

To sum up:

- Optimization algorithms performance is related to the dataset they train on, and some can perform better than others with specific datasets and network architectures.
- Optimizing hyperparameters such as learning rate during training a network is necessary to reduce the loss of the network and improve computational efficiency.
- Adam and SGD-Nesterov momentum algorithms are computationally efficient optimization algorithms, with Adam having the advantage of adaptive learning rate optimization.
- In some cases, SGD based algorithms are used instead of Adam, where Adam can have difficulties generalizing on the training dataset, where they perform well at the beginning of training but are outperformed by SGD at later stages, and a room for improvement is still available for adaptive algorithms such as ADAM [14].
- For optimal results, while using a DNN, it is recommended to implement and compare various optimization algorithms with the developed datasets before implementing one in the final implementation of the network.

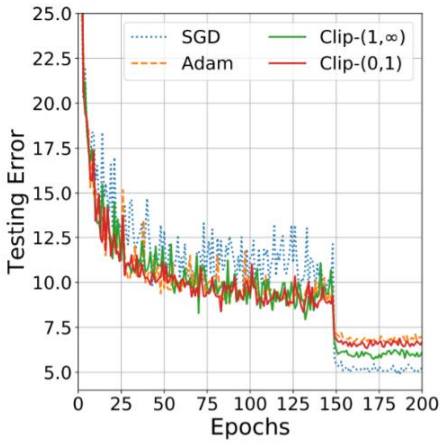


Fig. 12. Training the DenseNet architecture on the CIFAR-10 data

1.4.Deep learning-based object detection

Object detection tasks built based on deep neural networks have gained popularity in the last five years, especially after the evolution of image classification based on convolutional neural networks. Image classification simply identifies the type of object present in an image by assigning a class to it, based on the dataset provided for training the network. Object detection is a more complicated task that combines image classification and object localization. For object localization, bounding box regression is used to return the bounding box coordinates, which localize and border the classified objects. Image classification can only detect one object in an image. With object detection, a various number of objects can be classified and localized.

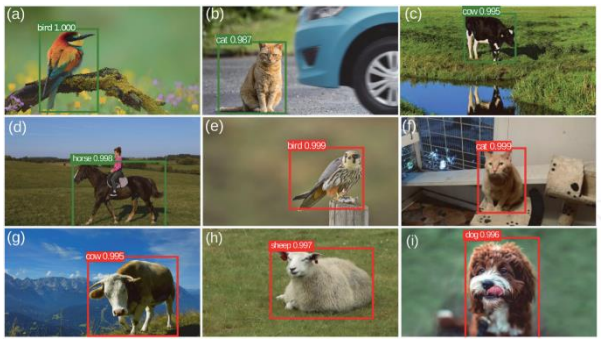


Fig. 13. Object detection based on deep learning [15]

The three main object detection techniques built on deep learning and commonly used on embedded systems are the Single-Shot-Detectors, You-Only-Look-Once and the Faster region convolutional neural network. The following figures compare between the mentioned techniques where the main criteria are the mean average precision of the detection network and their speed which is concerned with the time spent on detecting an image after training the dataset. The PASCAL VOC dataset was used for comparing the performance of the SSD and Faster RCNN networks. According to [15], Faster RCNN had the highest mean average precision amongst all networks with a value of 96.07 %. SSD detector had a mAP value of 84.35 %. The detection time taken for Faster RCNN had a mean value of 30 ms with a standard deviation of $\pm$ 2 ms whereas, SSD had a mean value of 17 ms with a standard deviation of $\pm$ 2 ms. Thus SSD performs almost twice as fast as Faster RCNN at the expense of a lower network detection precision. The results of the literature are shown in Fig. 14.

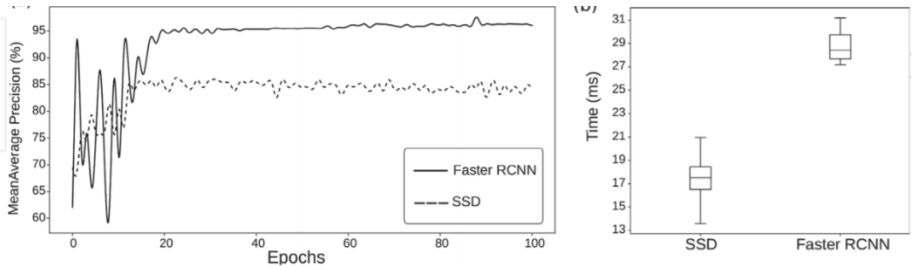


Fig. 14. Performance comparison between SSD and Faster RCNN object detection DNNs

1.5.Depth measurement techniques incorporating computer vision

Acquiring three-dimensional information about components undergoing inspection is an essential aspect of constructing automated visual inspection systems. Moreover, due to the rapid growth in computer vision application in the quality inspection sector, three-dimensional vision systems have been widely integrated into quality inspection systems. The focus in the thesis is the electronics industry, where inspecting the depth of pressed holtite sockets into PCB holes is the application under focus.

1.5.1. Standard techniques used for 3D measurement and inspection in the electronic industry

The most common techniques used for non-contact depth measurement incorporating computer vision are the time of flight, structured light, stereo vision, and laser triangulation systems.

1. Time of flight

The time-of-flight principle works by illuminating light onto the object's surface, whose depth is measured. The light beam is reflected onto a camera sensor, and the time taken is calculated. The depth can be measured using the time taken for the pulses of light emitted and the speed of light. The pulses of light are usually very short, which last for few nanoseconds. The principle is visualized in Fig. 15.

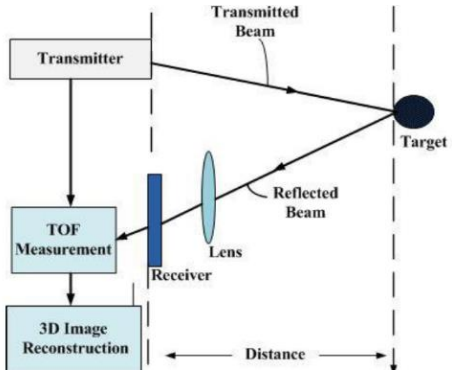


Fig. 15. TOF working principle [16]

2. Stereo vision

Stereo vision is a well-known widespread technique that uses two cameras to estimate both cameras' distance from an object. The setup of such a system is shown in Fig. 16.

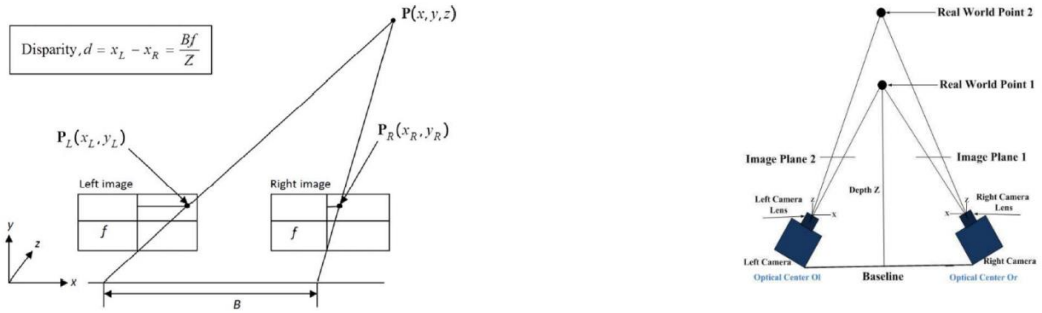


Fig. 16. Stereovision system setup and triangulation for depth calculation

The cameras are separated by a horizontal distance called the baseline, and by using triangulation, the distance to the object can be measured. This setup is known as a passive stereo vision setup that does not use light sources or projection.

The distance is expressed as follows: [17]

$$z = \frac{fB}{d} \tag{1.15}$$

$$d = x_L - x_R \tag{1.16}$$

Where:

- Z is the depth from the object to the camera lens
- f is the focal length of the cameras.
- B is the baseline, the horizontal distance between the cameras
- d is the disparity.
- x_L is the horizontal position of point p in the frame of the left camera
- x_R is the horizontal position of point p in the frame of the right camera

Matching the same point in both cameras is known as the correspondence problem. The main challenge of stereo vision systems is solving the correspondence problem to find the disparity and calculate the distance. This requires extensive computational efforts for extracting features and matching them to represent the same object in both camera frames [16].

The general formula for calculating the depth measurement error of a stereovision system is shown below in Equation 1.17.

$$\delta Z = \frac{Z^2}{bf}\delta d \tag{1.17}$$

Where:

- δZ is the depth error
- δd is the disparity error

3. Structured light

This technique utilizes a setup consisting of a camera, a high-quality projector (usually an LCD/DLP), and an image processing unit. The projector emits a light pattern towards the object of interest. The

light pattern gets obstructed by the object and thus detected by the camera. The camera captures images that are used by the processing unit to calculate the depth of the object. The calculation uses the triangulation technique shown for passive stereovision; however, since one camera is replaced with the projector, the correspondence matching is avoided.

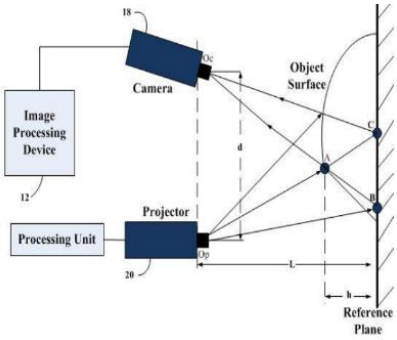


Fig. 17. Structured light system's working principle

Having discussed each technique, the following table compares between them.

Table 1. Comparing between depth measurement techniques used in CV applications [16]

criteria	Stereovision	Structured light	Time of flight	Laser triangulation
Material costs	Low	Medium/high	Medium	High
Low light performance	Weak	Depends on the chosen light source	Good	Good
Software complexity	High	High	Low	High
Depth measurement accuracy	Depends on selected system parameters	µm-cm	mm-cm	µm
Measurement range	Short to mid-range	Very short range	Short-range	Very short range
Image resolution	High resolution. Depending on cameras selected	High resolution. Depending on cameras selected	Up to 204x204	Camera dependent
Frame rate	Camera dependent	Camera dependent	Up to 25 fps	Camera dependent

The following table outlines the advantages and disadvantages of each discussed depth measurement techniques.

Table 2. advantages and disadvantages of techniques summarized [16, 18]

Technique	Advantages	Disadvantages
Stereovision (passive)	-Cost-effective solution - Performs well on objects with high texture -high resolution of the captured scene. -Depending on the setup, it can achieve high distance measurement accuracy	-Does not perform well for scenes of weak texture -Limited to a well-defined object -Weak performance under low light exposure
Structured light	-High data acquisition rate -Performance does not depend on ambient light	-Since it does not implement correspondence, occlusions can occur -Harder to measure depth difference between objects having interference.

Time of flight	-High measurement accuracy for medium-range measurements -performance is independent of ambient light	- Accuracy degrades when objects are placed at a close range to the camera -Expensive

1.5.2. Stereovision systems active and passive

From the comparison shown in Table 2. it can be seen that a stereovision system could be the best solution for depth measurement if it can overcome its bad performance under low light and with scenes of weak texture. Active stereo vision systems solve those problems, where an active system also introduces a light source to add texture to objects of weak texture and contrast. Moreover, it solves the issues with low light. Furthermore, any light illuminator can be chosen; it does not necessarily need to be an expensive projector like in the cases of structured light and time of flight methods. Thus, active stereo vision systems add to the cost of a passive system but with significant performance improvements. The following paper [19] compares the depth measurement accuracy between passive and active stereovision systems. 2 setups were compared for each system, one setup where the cameras are perpendicular to the measured object, which was a wall, and another set up with the cameras tilted at angles to the wall. A wall was chosen as the object to measure due to the soft textures and low contrasts, which exposes passive systems to its low unfavourable conditions where their performance is low. The setup and results of the experiment are shown in Fig. 18 and Fig. 19. The results show that active systems have a reduced measurement error where the range error is quadratic in range for passive stereo systems and cubic in range for active stereo systems. Thus it can be deduced that in cases where the distances of the objects are to be measured where the surface texture is weak, or the contrast between the objects is low, then an active stereo system setup would be needed and would still be cheaper than other techniques discussed above in The following table outlines the advantages and disadvantages of each discussed depth measurement techniques.
Table 2.

Fig. 18. Experimental setup, closeup (left), perpendicular set up at night time (centre), and tilted setup at day time (right)

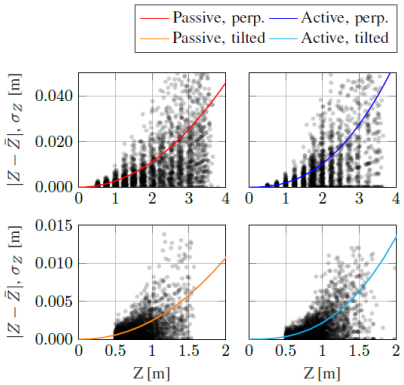
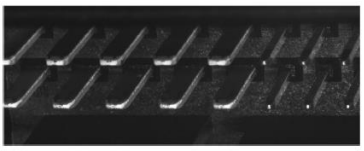


Fig. 19. Experimental results on different setups for passive and active systems

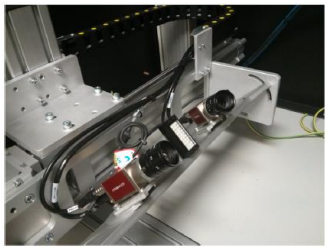
1.5.3. Review of an active stereo vision system used in the inspection of electronic connectors.

An active stereo vision system is implemented in a similar application measuring the depth of an electronic connector used in a production line [18]. The system setup is shown in Fig. 20. The system uses a tilted setup of ethernet-based GigE vision cameras with a white light illuminator used as the active source. The illuminator enriches the texture and contrast of the connector pins as they have shiny surfaces, where a passive setup would not perform well. The system has the following setup:

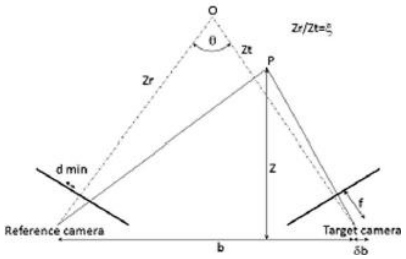
- A baseline distance between the camera sensors of 250 mm
- A distance from the cameras to the connector of 160 mm
- Two 3.75 µm pixel size 1.2 MP GigE vision ethernet-based cameras
- Two 25 mm focal length high-resolution lenses for machine vision
- An angle of convergence between the tilted cameras of 73°
- The field of view of the setup is 30 x 40 mm, and the depth of field of the cameras is 4mm



a) Connector pins



b) System setup



c) Triangulation used to calculate depth

Fig. 20. inspected object and system layout

Pattern matching algorithms with pyramidal decomposition and pixel refinement are used to solve the correspondence problem for matching the same point of the object (p) in both camera frames (right and left). Moreover, this algorithm works well because the pattern of the connectors can be recognized by image processing techniques, as the connector's geometric cross-sectional shape is not complex. The distance computing algorithm consists of the following:

1. camera calibration is done to computing the intrinsic and extrinsic properties of the camera with the lens and remove lens distortion.
2. Feature extraction is achieved using pattern matching algorithms with the pyramidal decomposition technique.
3. The images taken after feature extraction are rectified; since the cameras are tilted, the position of the connector in both camera frames will not lie on the same horizontal line. Rectification is used to align the position of the connector and thus its extracted features on the same horizontal line in both the right and left camera frames. This, in turn, simplifies the correspondence algorithm

where only the x values differences of the measured points representing the connector are used to compute the disparities.

4. The disparity values are calculated.
5. The real-world 3D coordinates of the connector are extracted.

The setup and algorithm both perform well in terms of depth resolution and accuracy. The resolution was found to be 2,6 µm, and the maximum error of the depth measurement was found to be 11.7 µm. The obtained values prove that stereovision systems can indeed be used to measure the depth of objects in a concise range with high accuracy compared to more expensive techniques such as laser triangulation. For measuring the depth accuracy, a translational stage was moved in steps of 1 µm for up to 10 mm. The error between the known movement and the measured depth of the connector is computed. The distribution of errors is shown in Fig. 21. It is also noticeable that when the distance increases, the accuracy reduces.

Moreover, the distance moved to 10 mm by the stage means that the total distance between the camera and the connector increased to 170 mm at the complete translation stage. Since the depth of field of the chosen lenses is only 4mm, it means that from 4-10 mm ranges of movement of the stage, the scene was not in good focus. However, the maximum error was only 11.7 µm. To have a better interpretation of the error, the maximum error taken at until 4 mm, which is the maximum depth of field of the cameras with the lenses, from the graph its seen to be one value's deviation to 11 µm, however for precise points clustered together, the accuracy is less than 5 µm. The linearity of the measurement is also high, with only a 1.25 % error [18]. The experiment's reproducibility on a real production line where 3300 measurements were taken was 84 µm for the distance measurement.

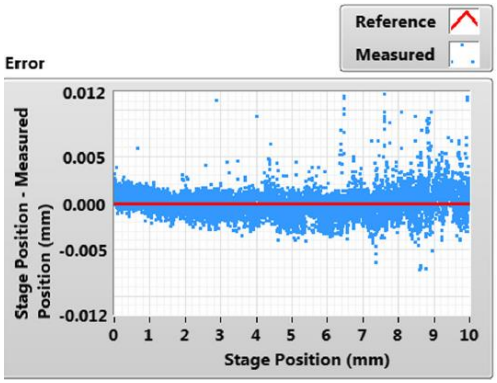


Fig. 21. Distance measurement accuracy experimental results

The total cost of the system is quite expensive as just one industrial camera model used: The Allied Vision G-125 costs around $600. Thus, the total cost of the proposed can exceed a price tag of $2000, including the lenses used and remaining costs. Moreover, the algorithm also adds to the distance measurement error and affects the time taken for measuring the connectors' depth. The total time taken to inspect and measure distances using seven scenes and 156 pins of various shapes and dimensions was 65 seconds. Moreover, the processing was executed on an Intel i7 processor with 16 GB of RAM.

To conclude from the paper and previous literature regarding active stereovision systems:

- Choosing cameras of high resolution and a smaller pixel size improve the distance measurement accuracy

- Choosing lenses of high focal length and larger baseline distance between both cameras improves the measurement's resolution and accuracy
- the matching algorithm mainly feature extraction, and the choice of processor used for computations improves the accuracy of the measurement as well as the total cycle time
- active means incorporating an illumination source is necessary to achieve good depth measurement results, especially for objects having low texture and contrast

1.6.Embedded vision

Embedded vision systems are compact vision systems, similar in concept to CV, where the main difference is that embedded vision systems do not require an industrial PC to perform vision tasks. The systems are based on integrating camera modules, central processing units, and graphical processing units into a compact device, as shown in Fig. 22. Therefore, such systems are small in size and acquire low power consumption [20]. Additionally, such systems have lower purchase costs and less maintenance required compared to computer vision PC-based systems. Embedded vision systems are one of the tools that offer a massive potential in the fourth industrial revolution and can be used in the following smart factory applications [20] :

- Manufacturing of vehicle components.
- Industrial robotics.
- Packaging solutions.
- Manufacturing and assembly of electronic components.

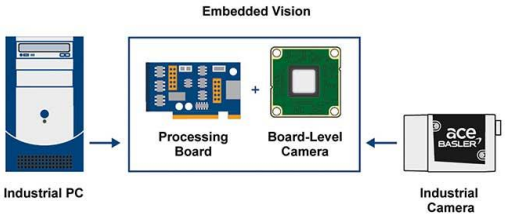


Fig. 22. Replacing traditional computer-based vision with embedded vision [21]

1.7.Cameras used in computer vision

The cameras are the most significant component of the automated visual inspection system. They are the sensor that is analogous to the human eyes for manual VI systems. Therefore, caution must be taken while selecting them. In this review, the camera sensors used in CV applications and the interfacing technology are presented.

1.7.1. Camera sensor

The camera sensor is the central component that a camera is built around. The main two types are CMOS (Complementary Metal Oxide Semiconductor) and CCD (charge-coupled device) sensors. Both operate with the same fundamental concept, where pixels collect light energy (photons) and convert them into an electrical charge to represent the image electronically. However, the process is carried out differently. Sony announced in 2015 that it would halt the production of CCD image sensors. To further contextualize with regards to CV applications, CMOS sensors have the following advantages over CCD [22] :

- Higher light sensitivity, which is beneficial in lower light applications.

– Improved pixel depth (saturation capacity), leading to a higher dynamic range.
– Lower Power consumption.
– Lower cost.

1.7.2. Camera interfacing technology

Industrial cameras used in inspection implement various interfacing technologies. The cameras could be interfaced with PCs or with embedded electronic systems. In the review, the concern is regarding embedded vision systems, as mentioned in the Embedded vision section. The standard camera interface technologies in embedded vision systems are MIPI CSI-2 D-PHY, USB Vision, and GigE Vision interfaces. Mobile Industry Processor Interface (MIPI) CSI-2 is a standard interface used in mobile devices such as smartphones to connect the sensor to the processing unit. Typical applications of using the interface are the automotive and IoT industries. USB vision is a standard of industrial interfacing cameras based on the USB 3.0 interface. GigE Vision is a standard that incorporates the use of ethernet cables to transmit data up to a distance of 100 meters at a bandwidth speed up to 115 Mb/s [23]. Table 3 compares the three standards, including the most important criteria.

Table 3. Comparison between camera interfaces used in embedded vision systems [23, 24]

	MIPI CSI-2 D-PHY	USB 3.0 Vision	GigE Vision
Bandwidth	1.5 Gb/s v1.1 (oldest)	Up to 400 Mb/s	115 Mb/s
Cable Length	Up to 0.6 m	Up to 0.8 m	Up to 100 m
CPU usage	Low	Medium	High
Software complexity	High	Low)	Low
Size	Very small	Small	Medium
Cost	Low	Medium	Medium

As seen from the table, the MIPI CSI-2 interface is the best option due to its very high bandwidth speed of transmitting image data and consuming the least power. Furthermore, combining performance with low cost gives it a winning edge over the other options. However, the only drawback would be the less user-friendly interface and the required programming knowledge to program the cameras.

a) MIPI CSI-2 camera

c) USB 3.0 vision camera

c) GigE Vision

Fig. 23. Cameras of various interfacing technologies

2. Deep learning implementation

2.1.Background and implemented algorithm

Deep learning-based object detection was needed to classify and localize both the holtite socket and the PCB hole into which the socket is pressed. The algorithm works by incorporating an SSD-Mobilenet V2 object detection network that uses an SSD-300 single shot Multibox detector with the Mobilenet V2 convolutional neural network. Mobilenet is an image classification CNN with the advantages of low power consumption and high inference speed, which uses a depthwise separable convolution to speed up the network inference time while maintaining high accuracy. With regards to Nvidia jetson nano, as shown in Fig. 24 The fastest object detection network was the SSD (300x300) Mobilenet V2 network, which takes in an image and resizes its resolution to 300 ×300 pixels before training the network. Moreover, the steps of implementing the algorithm are discussed below.

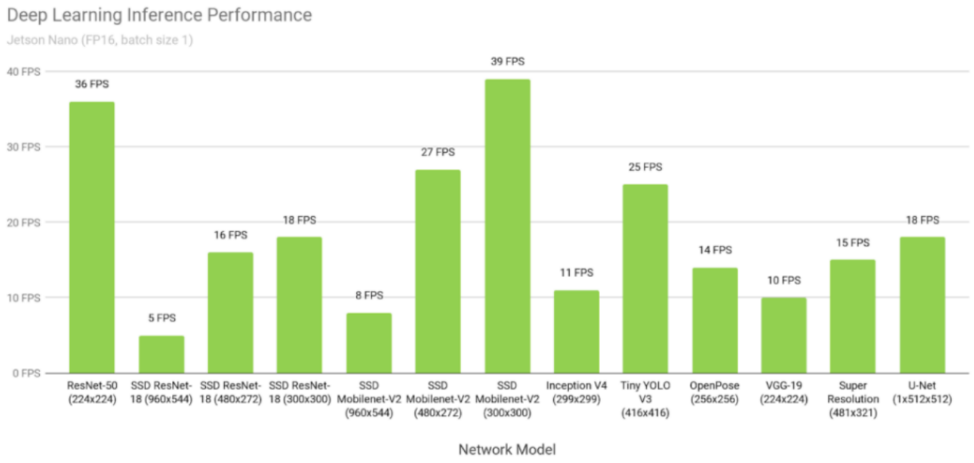


Fig. 24 Speed performance on popular deep learning object detection networks used with Nvidia Jetson Nano [25]

The Nvidia Jetson nano board version A02 was available to use to implement the object detection network. An inference by the Nvidia company known as jetson inference was used as the open-source platform to implement object detection on the holtite sockets of the PCB. Moreover, the algorithm followed is shown in The algorithm can also be simplified into three main steps:

1. Take different pictures of the PCB consisting of the pressed holtite sockets from both cameras individually
2. Before network training, labelling is used to identify for the network what each component is and localize its co-ordinates
3. The images and labels created are input to the network to train it.
4. During the training procedures, the network's hyperparameters such as the learning rate and optimization functions are tuned to speed up the training time and seek sufficient network training.
5. The training files are then exported to a format known as ONNX, which is used to deploy the generated neural network on the GPU of the Nvidia Jetson Nano board.
6. The following step is to test the trained network by running detections on pictures of PCBs that it had not seen before.
7. The localization results of the sockets and holes will later be used for stereo rectification to measure the depth of the holtite socket pressed into the PCB holes.

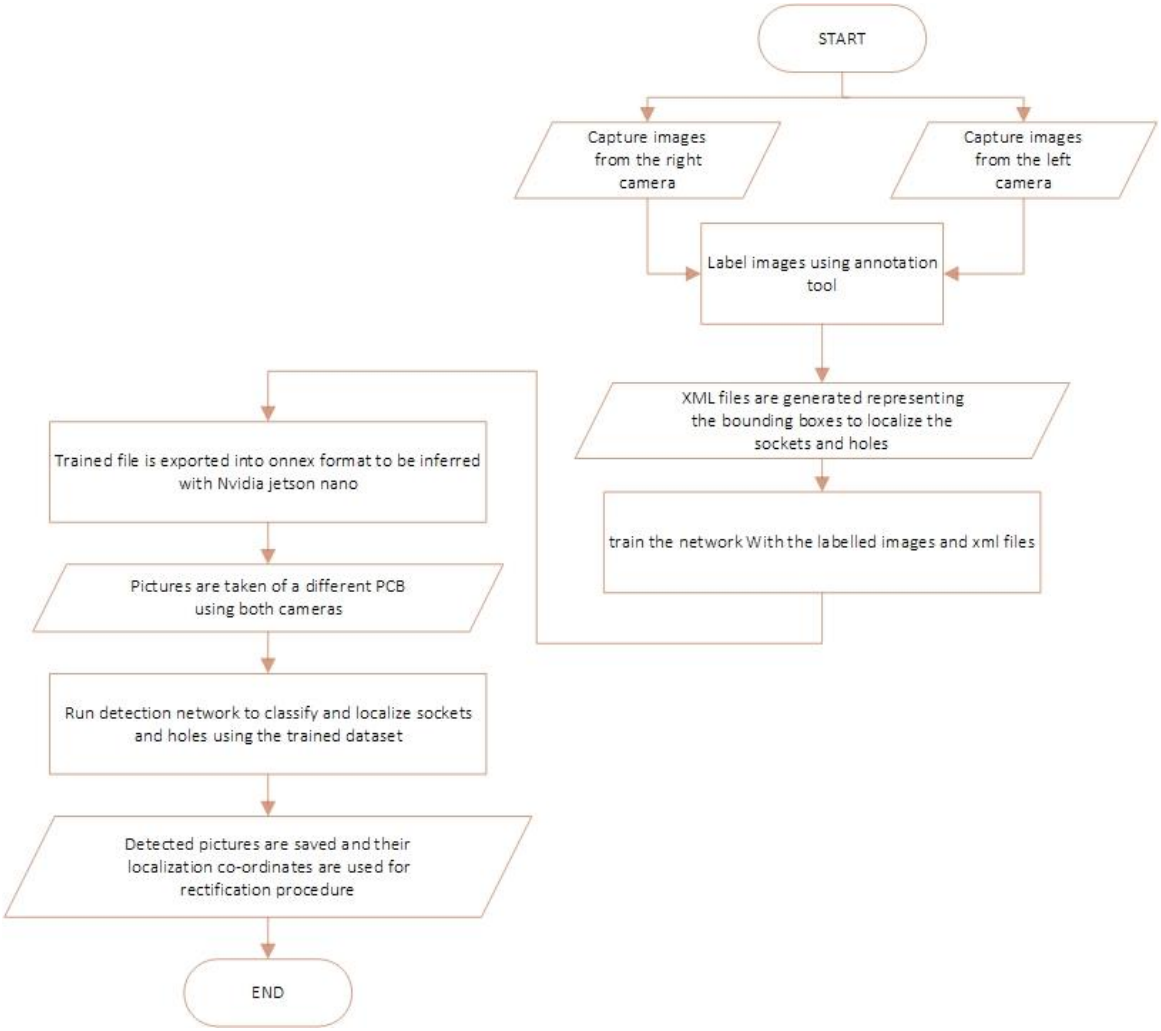


Fig. 25. Flowchart for implementing deep learning-based object detection

2.2.Hyperparameters and tuning

The stochastic gradient descent with Nesterov momentum was used as the optimization function, which showed good detection results with the CIFR-10 dataset, as shown in the literature review. The momentum value was set to 0.9, which is a standard momentum value used with Nesterov. The implementation of this is shown in Fig. 26. Moreover, the learning rate was initially set to 0.01 to speed up the training process. However, not all detections were precise. Therefore, the learning rate was later modified to a slower learning rate of 0.005. As a result, a more precise localization of the bounding box and classification of the object was obtained. The difference can be visualized in Fig. 27 and Fig. 28

```
# Params for SGD
parser.add_argument('--lr', '--learning-rate', default=0.01, type=float,
                    help='initial learning rate')
parser.add_argument('--momentum', default=0.9, type=float,
                    help='Momentum value for optim')
parser.add_argument('--weight-decay', default=5e-4, type=float,
                    help='Weight decay for SGD')
parser.add_argument('--gamma', default=0.1, type=float,
                    help='Gamma update for SGD')
parser.add_argument('--base-net-lr', default=0.001, type=float,
                    help='initial learning rate for base net, or None to use --lr')
parser.add_argument('--extra-layers-lr', default=None, type=float,
                    help='initial learning rate for the layers not in base net and prediction heads.')
```

Fig. 26. Implementation of Nesterov SGD optimization function in the code

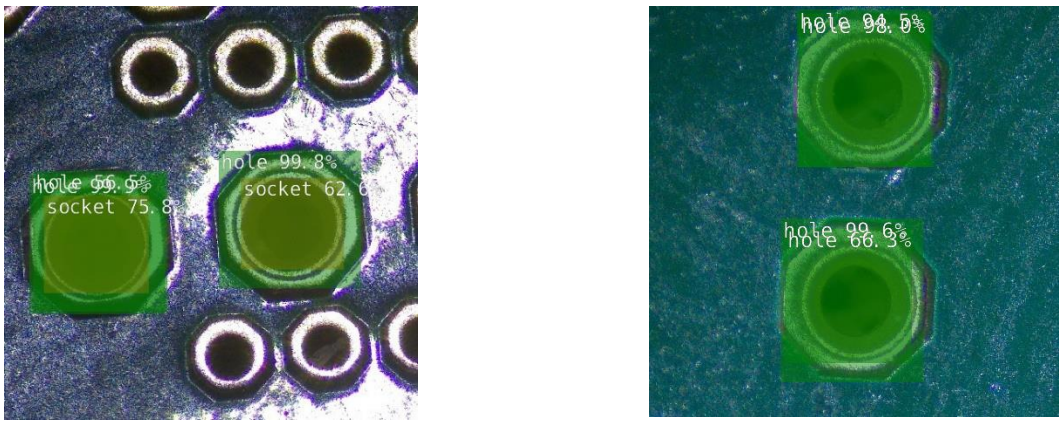


Fig. 27. Bad detections with a learning rate of 0.01

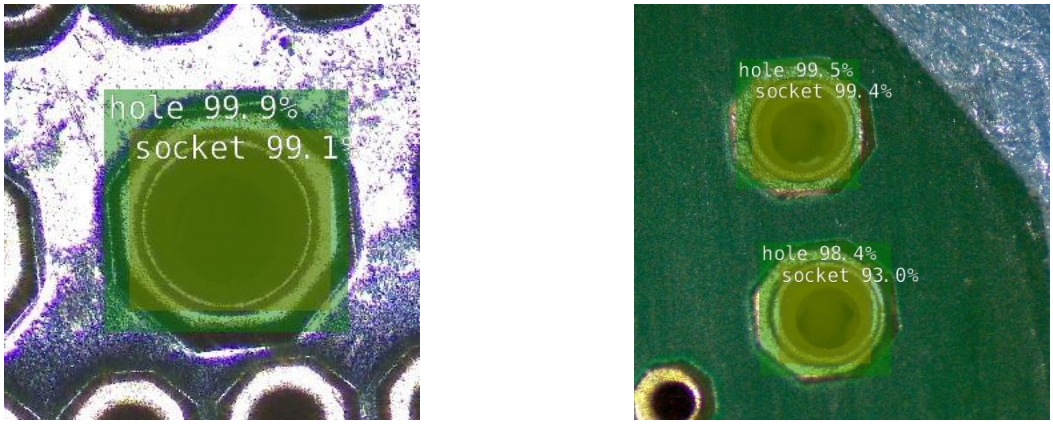


Fig. 28. Improved detections with a learning rate of 0.005

2.3.Training and validation

The labelling of the images was done using the MakeSense AI annotation tool, where images were uploaded to the tool, and the tool allowed for drawing rectangles around the object classes, which in this case were the holtite socket and PCB hole. The labelled files were then exported to pascal VOC dataset format, which assigns a class ID to the socket, for example, Class ID 1 and another ID to the hole, e.g. Class ID 2. Moreover, the coordinates of the bounding box, which are the rectangles shown in yellow and green, are the localization coordinates of each object. The number of epochs was 20. Epochs are simply iterations executed by the network where the loss function is calculated, and the weights are updated to reduce the error of the network, as discussed previously in the literature review. The dataset used for training was carried out in the following manner:

- 70 % of the dataset was used for training
- 20 % of the dataset were used for validation. The network tries to predict the results of the validation dataset, and according to the predictions, the weights are updated to improve the predictions.
- 10 % of the dataset were used for testing, where the network tries to detect those images according to what it learned through the training phase when it sees the test dataset for the first time.

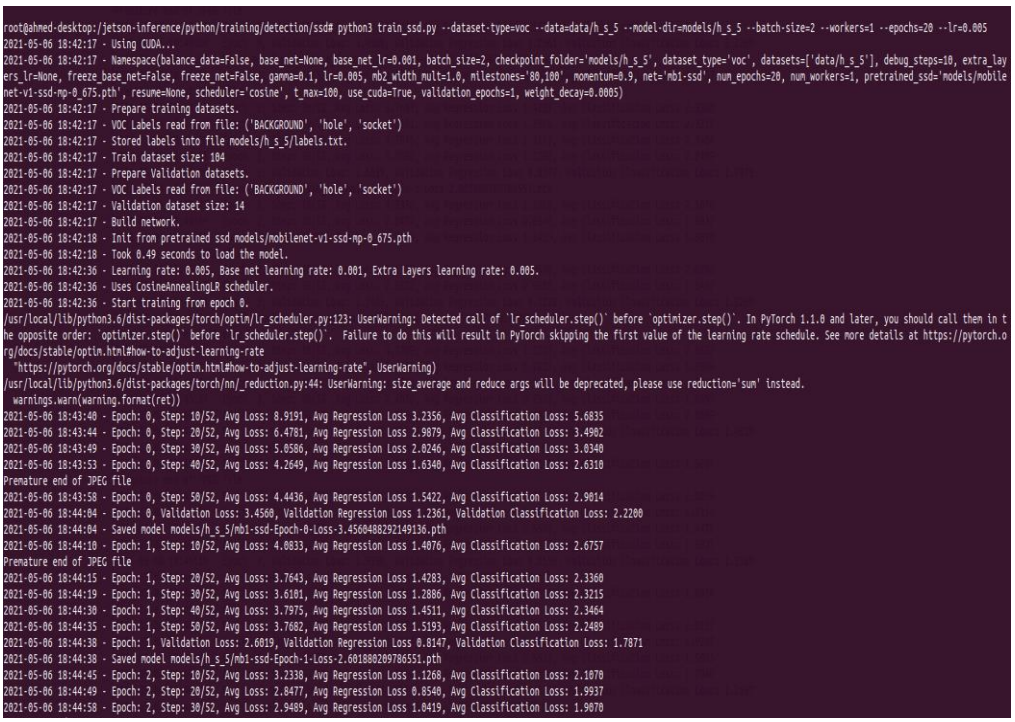


Fig. 29. Training the network

The iterations taken during training are shown in Fig. 30. 104 pictured were trained with a batch size of 2. The reason for such number was due to having 2 PCB samples where each had 32 sockets and holes. Moreover, 10 measurements were taken later where those sockets and holes were not used for training the network. They were only used for the detections in order to accurately test the network's performance. Moreover, the losses of the network are the sum of both classification and regression errors of the network. The losses were seen to decrease as the number of epochs was increasing. In Fig. 30, the number of epochs (iterations) is plotted against the network loss. The network loss consists of the regression loss and the classification loss. The classification loss is the loss in classifying the object, in this context, classifying whether the object is a PCB socket or a PCB hole.

Moreover, the regression loss is the loss of the network with regards to localizing the object. That is the loss in calculating the accurate bounding coordinates of the objects. It can be seen that the relationship is not linear, where increasing the number of iterations reduces the loss. Oscillations occur such that at higher numbers of iterations, the losses start to increase again. However, at epoch 19, representing the twentieth iteration, the losses started to stabilize and reduce. Especially for the regression loss, which was almost constant from the seventeenth to the nineteenth iteration. The results are reflected in the samples shown in Fig. 28, where the sockets and holes are detected with good accuracy.

Moreover, the bounding boxes drawn around them are precise, which proves the success of the network. However, one problem is that not many sockets and holes can be detected at once with this network. The reason is that the network operates with image sizes of 300 x 300 pixels. This means that each input image is resized to 300 x 300 before the images are either trained or detected. When having many holes and sockets in one image and then resizing the entire image into a 300 x 300 pixel-sized image, the detection becomes very hard and inaccurate. One solution to overcome this problem would be to ensure that not many sockets and holes are in the field of view of the cameras and

microscopes. Another solution would be to use a slower detection network that resizes images to bigger pixeled image sizes.

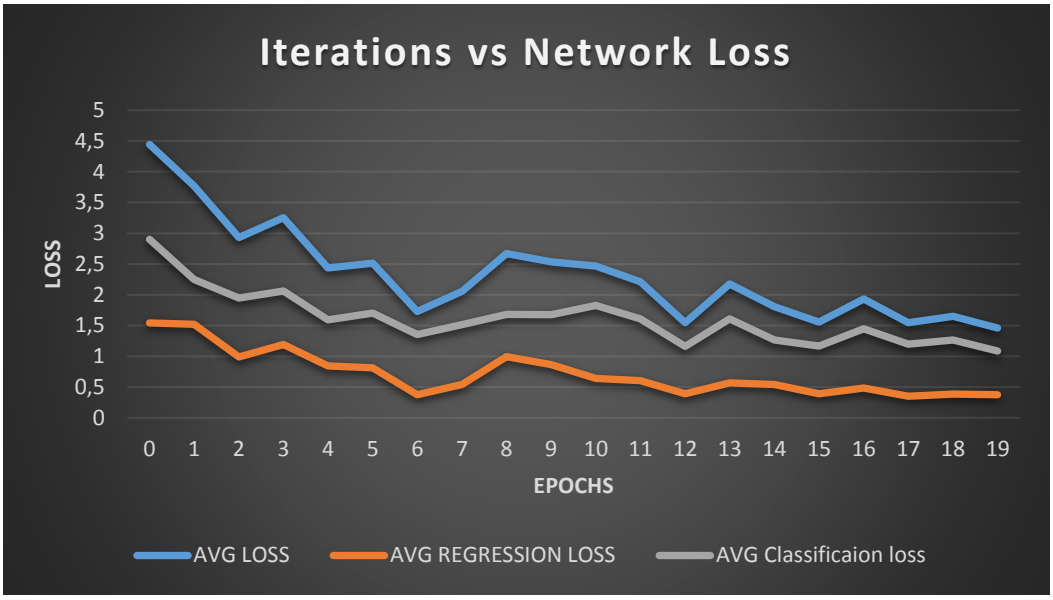


Fig. 30. Network Loss vs number of epochs

The following figure shows the time taken for detecting the number of sockets in both the left and right images shown in Fig. 57. The detection time taken for the left image containing two sockets and two holes is 87.5 ms and the detection time taken for the right image which contains four sockets and four holes is 65.8 ms.

```
root@ahmed-desktop: /initial_recog
detected 4 objects in image
[OpenGL] glDisplay -- set the window size to 716x540
[OpenGL] creating 716x540 texture (GL_RGB8 format, 1159920 bytes)
[cuda]   registered openGL texture for interop access (716x540, GL_RGB8, 1159920
 bytes)
[image] saved '/initial_recog/rectified/rect_out_0.jpg'  (716x540, 3 channels)

[TRT]    ------------------------------------------------
[TRT]    Timing Report ssd-mobilenet.onnx
[TRT]    ------------------------------------------------
[TRT]    Pre-Process   CPU   0.11932ms  CUDA   1.57698ms
[TRT]    Network       CPU  87.54333ms  CUDA  85.94422ms
[TRT]    Post-Process  CPU   1.10505ms  CUDA   1.10318ms
[TRT]    Visualize     CPU  28.94146ms  CUDA  29.29615ms
[TRT]    Total         CPU 117.70917ms  CUDA 117.92053ms
[TRT]    ------------------------------------------------

[TRT]    note -- when processing a single image, run 'sudo jetson_clocks' before
                 to disable DVFS for more accurate profiling/timing measurements

[image] loaded '/initial_recog/rectified/rec_2.jpg'  (716x540, 3 channels)
detected 8 objects in image
[image] saved '/initial_recog/rectified/rect_out_1.jpg'  (716x540, 3 channels)

[TRT]    ------------------------------------------------
[TRT]    Timing Report ssd-mobilenet.onnx
[TRT]    ------------------------------------------------
[TRT]    Pre-Process   CPU   0.08693ms  CUDA   1.34859ms
[TRT]    Network       CPU  65.84052ms  CUDA  64.48771ms
[TRT]    Post-Process  CPU   1.28724ms  CUDA   1.28588ms
[TRT]    Visualize     CPU   0.42094ms  CUDA   4.37870ms
[TRT]    Total         CPU  67.63563ms  CUDA  71.50089ms
[TRT]    ------------------------------------------------

root@ahmed-desktop:/initial_recog#
```

Fig. 31. Time taken for detecting sockets and holes in left and right images of a rectified stereo pair.

3. Mechanical design of the system

In the following section, a mechanical design developed to implement the stereo vision depth measurement system will be introduced to the reader. To better understand the final design of such a system, it is necessary to understand some of the issues that needed to be tackled to develop an assembly that would perform accurately and avoid failures within its parts. In the early stages of the design, three main concerns were identified and solved:

- Electrostatic discharge, which would affect the camera electronic board, had to be avoided and housing the cameras needed to be designed carefully.
- Deformations occurring onto the camera board due to the weight of the microscopes, influencing the accuracy of the stereo rectification and depth measurements, which lead to the design of a support system that would sustain the weight of the mentioned components.
- Translation along all three spatial axes, which is deemed necessary to allow the stereo vision system to detect and analyse all the sockets within the PCB. Moreover, when needed, correction allows for a correction of the distance between the cameras and the PCB itself to correct the focus.

Given the time limitations of the project, the whole assembly was built using additive manufacturing 3D FDM technologies, which allowed for a prototype of the design to be manufactured and assembled in a short amount of time and provide flexibility to the system. For example, the convergence angle, the angle at which the cameras are tilted and changing the baseline. Such modifications can be implemented easily and quickly. Additionally, utilizing plastics to print all the different components allowed the costs of the prototype to be contained, which would have been increased if, instead, aluminium or steel alloys were used.

Table 4. Prusa PLA mechanical properties [26]

Property	Value
Poisson’s Ratio	0.33
Shear Modulus, MPa	2400
Density, g/cm3	1.24
Young’s Modulus, GPa	2.2
Tensile Strength, MPa	42
Yield Strength, MPa	50.8

At the end of the design process, through the use of Autodesk Inventor software's stress analysis simulation tool, the final assembly proved to be rigid enough to withstand the stresses and ensure that no deformation of the camera board would occur during the experiments.

Prusa PLA plastic was used for all the printed parts, and its mechanical properties can be observed in Table 4. Such parts include:

- Housing for the camera and its integrated PCB
- Supports for the microscopic lenses
- Support for the PCB to be inspected

3.1.Design components

For the stress analysis, being such material not included within the software, a custom material had to be created, and its mechanical properties were inserted as shown in Fig. 32

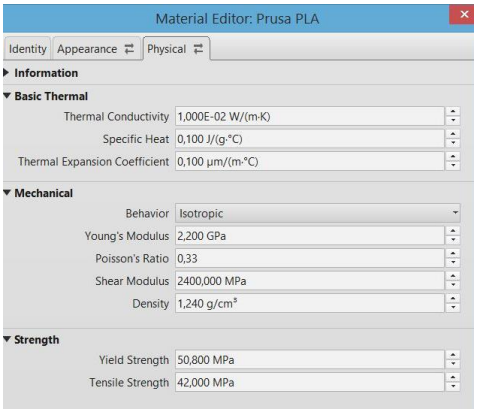


Fig. 32. PLA custom material in Autodesk Inventor creation

For standard additional parts, such as the screws used to assemble the components, steel is included within the Autodesk Inventor material library. The microscopic lenses and the lens mounted used manufactured with aluminium alloy, and the T-6061 alloy was assigned as the material for them in inventor. Moreover, for the PCB of the cameras, polyethene material was assigned to it, as it is one of the usually used materials in manufacturing PCBs. For the stress simulations, the geometries of both microscopes and cameras were reproduced in detail, such as determining the mass of each component with accuracy. The software calculates the weight knowing the density of the material used and the model's volume Fig. 33.

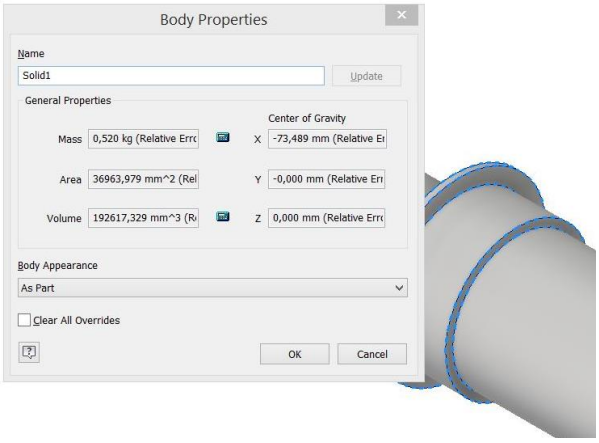


Fig. 33. Mass of the microscope in inventor

The system's final design is presented in Fig. 34, where It can be observed that the cameras with microscopes were mounted on top of the x-y table, such as facilitating the displacement to inspect the different sockets and holes. Moreover, it allows for correcting the distance between the lenses and the inspected PCB to focus the obtained images. In the following part, details for each of the components used within the assembling are provided.

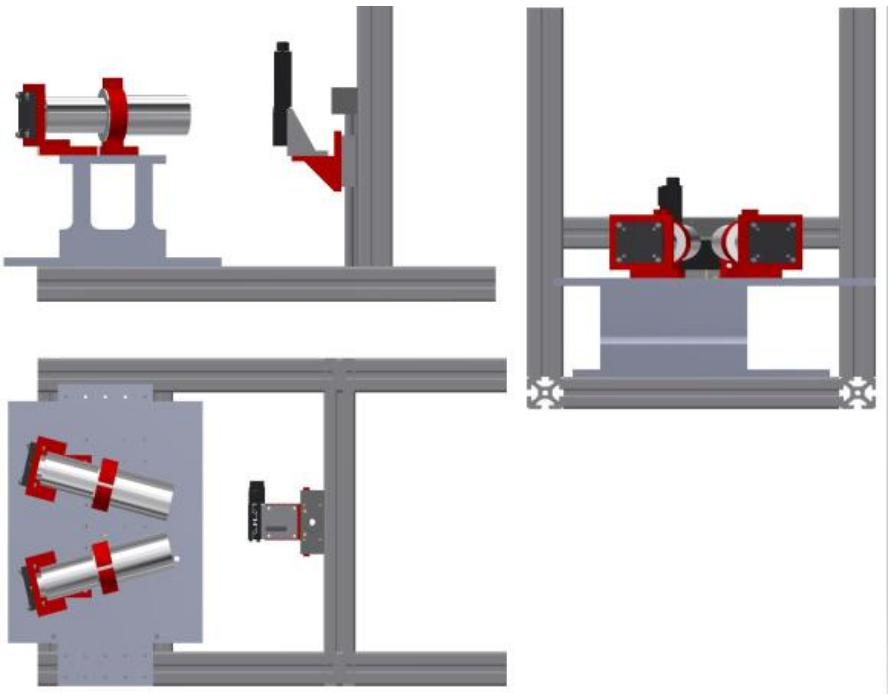


Fig. 34. Full assembly, right view (top left), top view (bottom left), and front view (right)

The first component to be analysed is the housing developed to mount the camera into the x-y table Fig. 35. As mentioned at the beginning of the chapter, an electrostatic discharge and bending induced by the weight of the microscopic lenses generated a considerable amount of concerns. The solution employed to counter the first issue was to house the camera, yellow in the figure, within two covers, the top cover is represented in black, separated from the camera through the use of spacers, to ensure that the electronic components would have no contact area with the plastic material it the top cover is manufactured.

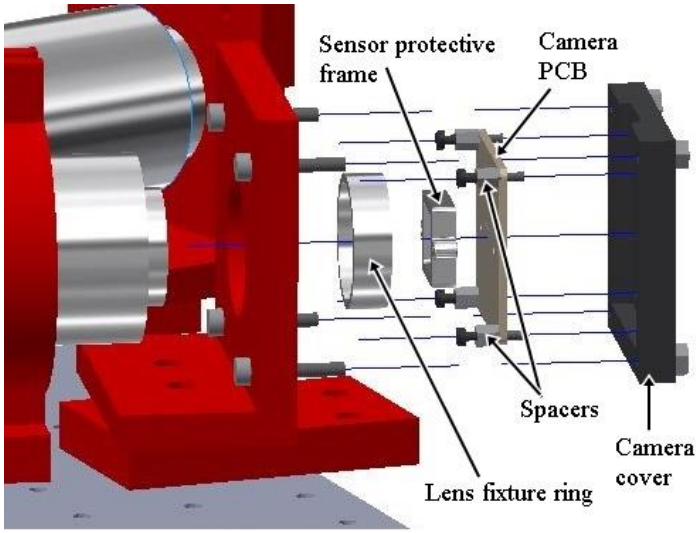


Fig. 35. Exploded view of the camera assembly

The bottom cover, red in the figure, however, did not require such measures to be taken, since no electronic components were present on the side of the PCB, therefore being in contact with the camera, to avoid any bending which could have occurred if spacers or other components were used, due to the weight of the microscopes connected to it.

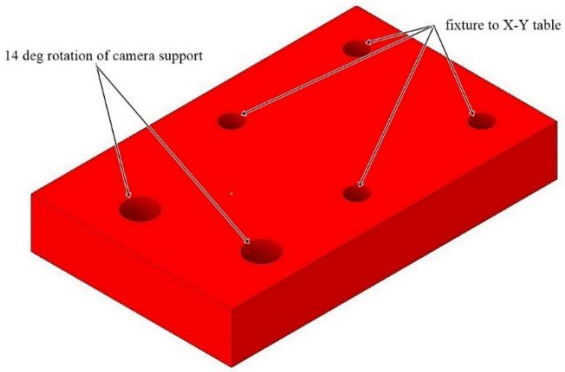


Fig. 36. Camera attachment to the x-y table

In Fig. 36, the attachment used to fix the cameras, covers, and lenses onto the x-y tables is shown. It was designed to constrain the components to the x-y table while providing an orientation angle of 14 degrees. The angle chosen is low due to the length of the microscopes, as they could hit each other if the baseline is not big enough. Moreover, the baseline which is the horizontal distance between the two camera sensors, is constrained to the length of the camera cables. Therefore, the chosen angle is relatively small. Moreover, due to the simple design, short printing time, and low material consumption, it could be easily modified if necessary to change the angles at which the lenses are tilted towards each other and thus change the convergence angle of the stereo vision setup. The baseline is shown in Fig. 37, to be 112.520 mm as a result of the system setup.

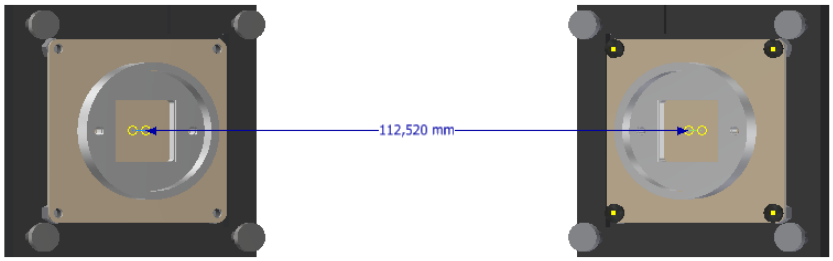


Fig. 37. Baseline measured from the setup

The PCB was mounted onto a one-axis translational stage, as shown Fig. 39, which was already available within the university. Such travel stage had a travel range of 15 mm. The PCB sample used for inspection had the following dimensions shown in Fig. 38, where the width of the PCB is 19 mm, thus this translational stage can be used for inspecting one PCB/

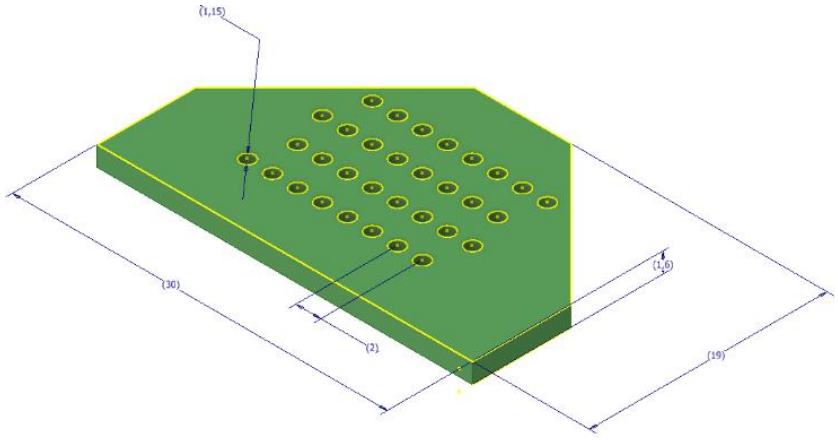
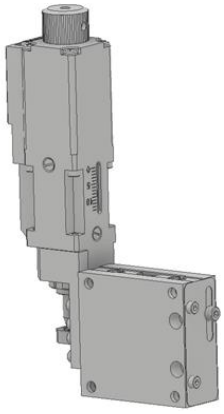
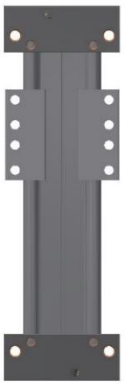


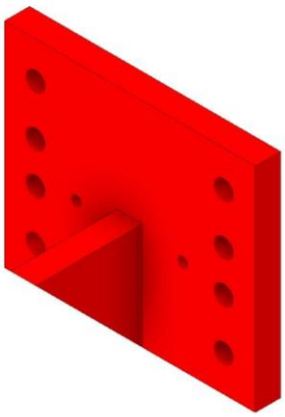
Fig. 38. PCB dimensions



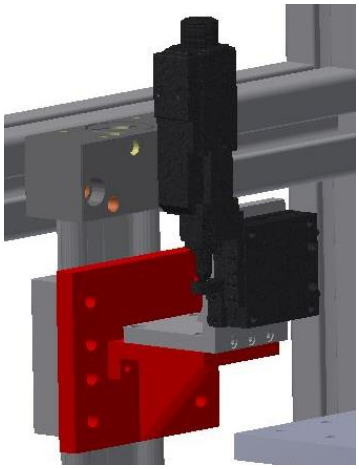
a) One-axis motorized translational stage



b) Linear bearings provided by IGUS



c) Linear bearing plastic carraige



d) Stage mounted to linear bearing

Fig. 39. Z-axis assembly

3.2.Stress analysis

It is now possible to move to the performed simulations, used to verify the capabilities of the design, ensuring negligible deformations would occur on the board of the camera and that the PLA material used would provide enough rigidity to support the whole assembly. One camera assembly was used for the stress analysis. The camera assembly undergoing the stress analysis with the analysis conditions is shown in Fig. 40.

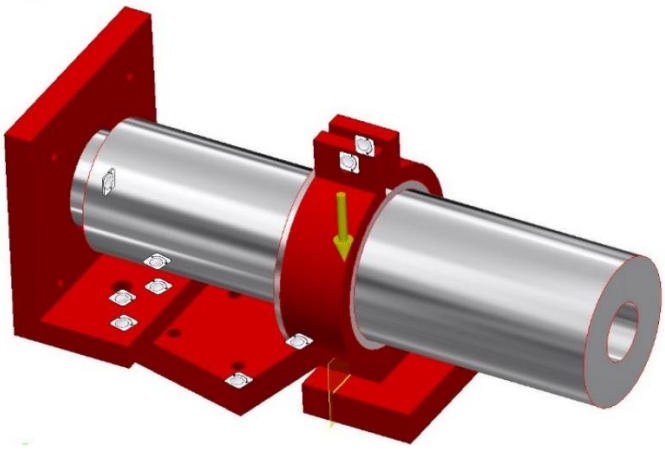


Fig. 40. Added pin constraints (white) and gravity load (yellow arrow)

To simulate the screws which would fix the components onto each other and the total assembly to the x-y table, pin constraints were applied. They are shown as the white components in the figure. while for the applied load, the weight of the microscopic lens was used since no additional loads would act on the assembly. Through the previously mentioned calculations performed by Autodesk Inventor to determine the weight of the lens, the gravity load of such components was added into the simulation, weighing 520 grams according to the model, while in reality weighs 500 grams.

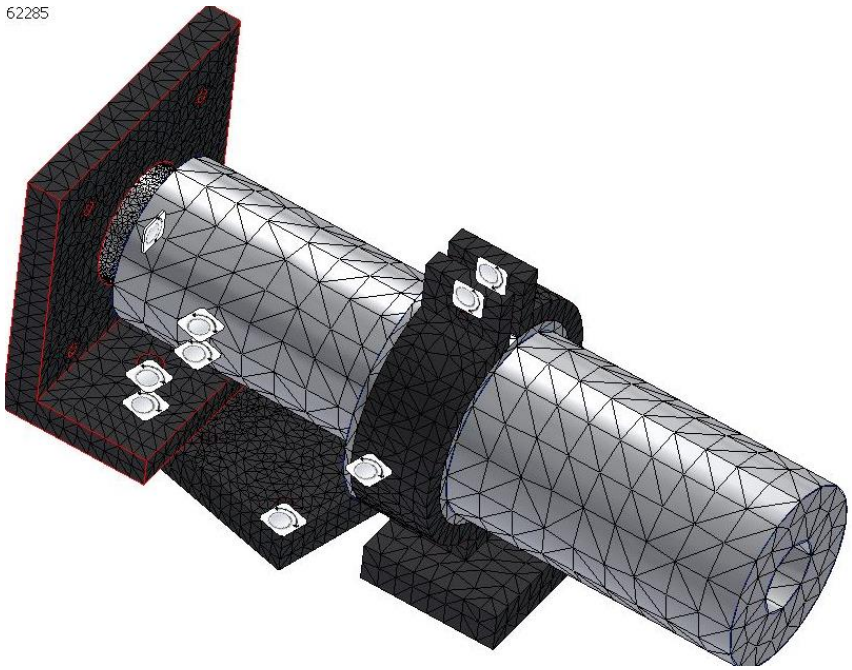


Fig. 41. Meshing of the assembly

Afterwards, adaptable meshing techniques were applied to ensure that, in areas affected by high stresses and where the higher values of deformation would be expected, the meshes would be refined, while in less significant and affected areas of the assembly, the meshes would be coarse, as shown in Fig. 41.

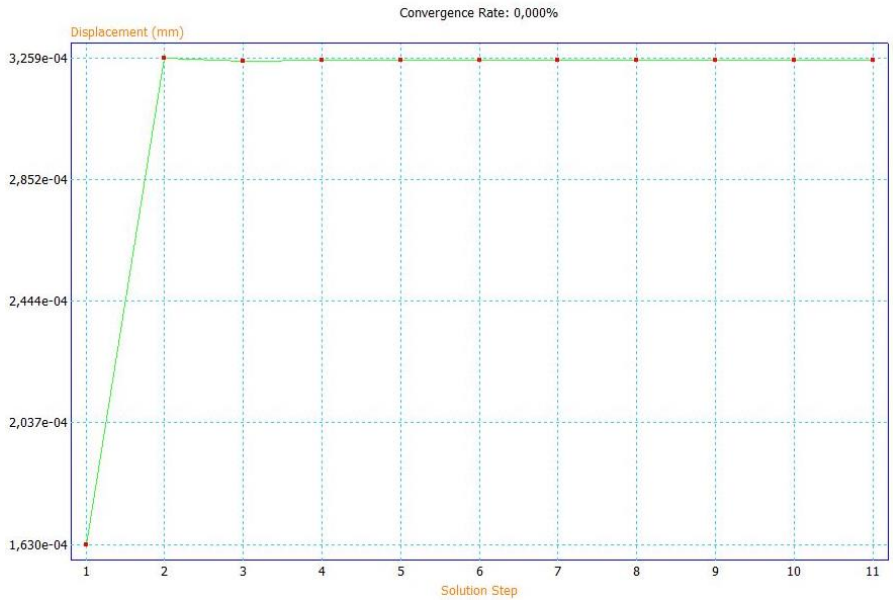


Fig. 42. Results convergence graph for displacement

Afterwards, the analysis was performed, with an analysis terminating criteria set to be reached either after 20 total iterations or when the difference in value between the von mises stress of the previous iteration becomes smaller than 1%. It is possible to observe in Fig. 42, that the convergence for the displacement's simulation was obtained within the 4th step. In fact, after a great initial difference between the first and second iteration, the value obtained stabilized, reaching the required 1% difference at the fifth iteration, not requiring the software to perform the total twenty iterations.

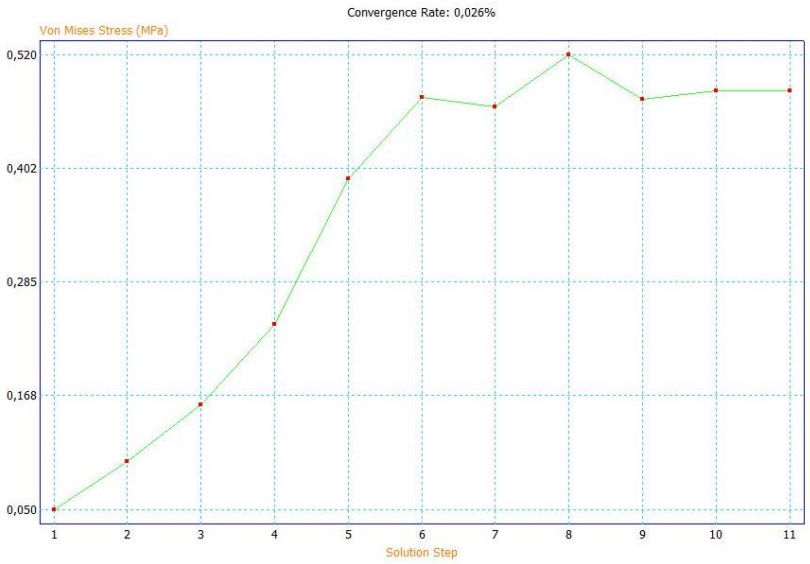


Fig. 43. Results convergence graph for von mises stress

When calculating the stresses acting on the assembly, however, as shown in Fig. 43, it is possible to observe how the software required more iterations to reach a solution difference of 1%. In fact, from iteration one through iteration ten, the solution of the calculation oscillated, reaching, finally, convergence at the eleventh iteration.

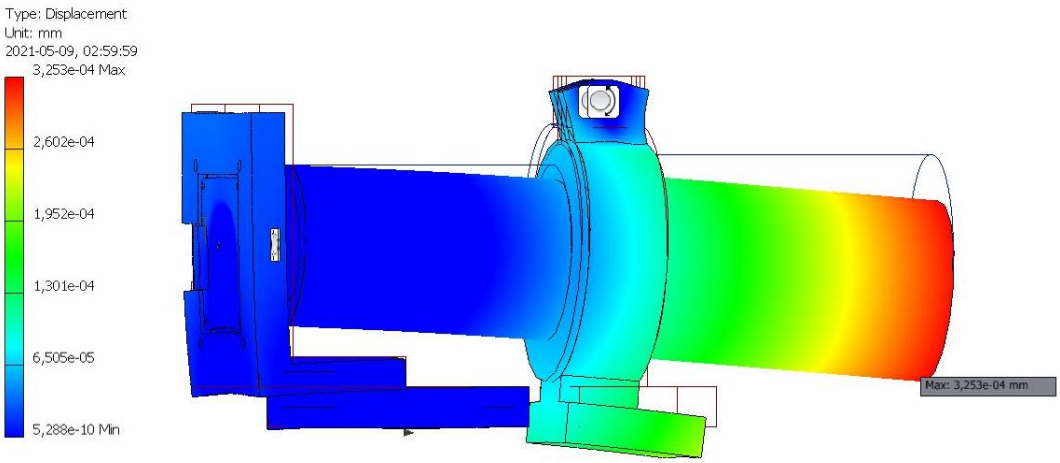


Fig. 44. Displacement's results

Having performed the calculations and reached a convergence, it is now possible to analyse the results obtained.

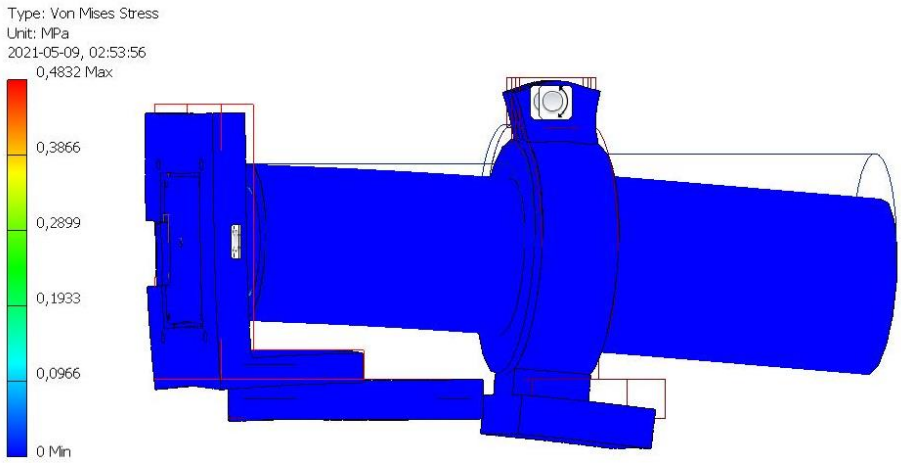


Fig. 45. Von Mises stresses results

The displacement results from Fig. 44 show how the highest displacements values were registered at the loose end of the microscopic lenses with values reaching the 0.325 micrometres, with the holder designed purposely to sustain the lenses absorbing the weight of it, experiencing a maximum deformation of 0.13 micrometres. The bending experienced by the camera board itself and the attachment to the x-y table reach negligible values, achieving one of the primary objectives of the assembly.

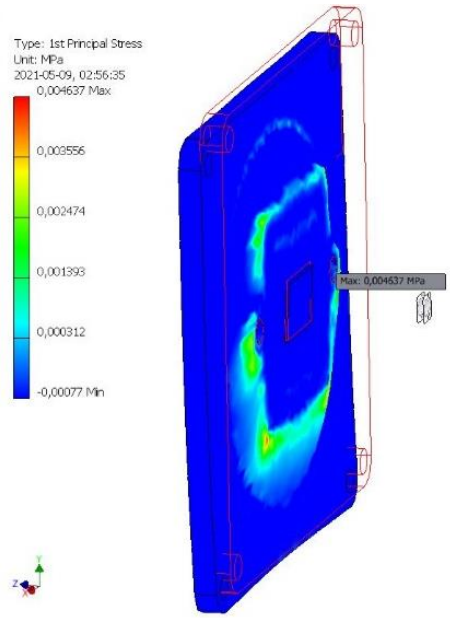


Fig. 46. Principal stresses developed on the camera PCB

According to Fig. 45, the stresses experienced by the assembly as a whole with the applied loads and constraints. It can be observed how the stresses experienced by it are entirely negligible, verifying that the designed structure can withstand the weight of the microscopic lens.

The maximum value of von mises stress is about 0.48 MPa, and this value occurs around the aluminium fixture onto which the lenses are fixed. For the plastic parts, the maximum principal stress, which is the used criteria for brittle material, shows negligible stress values around the camera's PCB in Fig. 46.

4. Set up and implementation of stereo vision

4.1.System setup

The setup of the stereo vision system plays a crucial role in the measurement accuracy provided. At first, the cameras and their respective sensors are chosen. After that, the lens is selected based on several factors such as focal length, angle of view, the field of view, and the range of distance the lens can view with good imaging focus. Having chosen the camera and lens, other system parameters are calculated and tuned to obtain the best possible theoretical depth measurement accuracy based on the provided constraints. Lightning conditions play a crucial role for the deep learning-based object detection algorithm and, consequently, the depth measurement accuracy. Once the setup is achieved, the implementation of the system for depth measurement is discussed.

4.2.Camera selection

The two most significant criteria for selecting the camera would be its performance and price and how it influences the overall cost of the system. One of the main aims of the thesis is to provide a cost-effective solution for the construction of an automated VI system. Therefore, various camera technologies were compared in the literature review, and subsequently, MIPI CSI-2 interface-type cameras were selected due to their low cost, small size, and high-performance capabilities. An essential requirement in selecting the cameras is to mount the lens on a camera supporting interchangeable lenses. To improve the accuracy of the stereo vision system, the focal length is an essential parameter in reducing the error, as shown previously in Equation 1.17. Therefore, the constructed system should be flexible in using different lenses which have different focal lengths. Moreover, the lenses are shown in the following section, in this section, it is important to keep in mind that the selected cameras to be compatible with using interchangeable lenses and not a fixed lens camera.

Furthermore, according to Equation 1.17, disparity error depends on the quality of the camera sensor, its pixel size, and the accuracy of the matching algorithm. The smaller the pixel size of the camera, also the less the disparity error. Therefore, another key parameter to search for is looking for cameras with a relatively small pixel size sensor. A comparison is made below in Table 5**.** Furthermore, a comparison was conducted between two brand new MIPI CSI cameras, namely, the raspberry pi HQ camera released in April 2020 and the e-con systems e-CAM131_CUNX - 4K camera, which was released in December 2020. Both cameras are compatible with Nvidia Jetson Nano. Moreover, a relatively cheap GigE vision industrial camera was included in the comparison to see the difference in performance capabilities between GigE vision and MIPI interface-based cameras.

Table 5. Camera selection [27, 28]

Criteria	Camera		
	Raspberry pi HQ	E-con systems e-CAM131_CUNX	Daheng imaging MER-500-14GM
Resolution (MP)	12.3 (4056 × 3040)	13 MP (4208 × 3120)	5 MP (2592× 1944)
Max frame rate (FPS)	13 FPS at 4K 30 FPS at 5 MP 60 FPS at Full HD	15 FPS at 4k 65 FPS at Full HD	14 FPS at 5 MP
Colour	Colour	Colour	Monochromatic

Sensor	1:2.3” Sony IMX477R CMOS	1/3.2” AR1335 CMOS	1/2.5” onsemi MT9P031 CMOS
Sensor size (mm)	6.29 × 4.72	4.54 × 3.42	5.76 × 4.29
Crop factor	5.6	7.61	5.97
Pixel size (µm)	1.55	1.10	2.2
Lens mount type	C/CS	S	C
Camera interface method	MIPI CSI-2	MIPI CSI-2	GigE vision
Total price (incl shipping cost, VAT, €)	125	107	217
Standard delivery time	7 working days	Up to 4 weeks	4-6 weeks
Supplier, shipping location	Hitech chain, Sweden	E-con systems, USA	Get-cameras, China
Manufacturer	Arducam	E-con systems	DAHENG IMAGING
Delivery cost	Free	30	49
Synchronization required for stereovision	synchronization bundle released in November 2020	Yes	Yes
Total Price of a stereo-setup (€)	240 (synchronized bundle)	282	532

The comparison shows that MIPI interface-based cameras are cheaper than the relatively cheap GigE vision-based camera by a factor of two. Moreover, the resolution, frame rates, and pixel size of the MIPI cameras are smaller than the GigE vision-based camera. Since the choice would be between the raspberry pi HQ camera and the e-con e-cam131, the specifications of both cameras are quite similar. The difference is that the e-con camera has a smaller pixel-sized sensor. However, the delivery time is long. For that reason, the raspberry pi HQ camera was chosen to have enough time for making a prototype to investigate the accuracy of the proposed system.

a) Raspberry pi HQ synchronized stereo kit b) E-con e-cam131 camera module

Fig. 47. MIPI interface-based cameras

4.3.Selection of lenses

Having chosen the cameras needed for the system, the following step is to choose the lenses needed. The main requirements of the chosen lens are:

- Lenses should have a good build quality
- Telephoto type lens with a high focal length to increase the system’s depth measurement accuracy

– Mount type should be C or CS to ensure compatibility with the raspberry pi HQ camera.

Referring to [12], the focal length of the used lens was 25 mm. Therefore, the chosen lens for this proposed system needs to have a higher focal length of that system to improve the depth measurement accuracy and other discussed factors. The lenses compared are manufactured by Seeed Studio company as lenses designed for raspberry pi HQ camera, which are relatively cheap. Moreover, the following table compares three lenses that could be used with the chosen camera.

The camera uses an IMX477 Sony sensor which has the following characteristics:

– Sensor width of 6.287 mm
– An aspect ratio of 1.33
– Sensor diagonal length of 7.857 mm
– A circle of confusion of 5 μm

Table 6. Lens selection [29]

Criteria	Lens		
	35 mm telephoto lens	50 mm telephoto lens	300 x microscopic lens
Focal length (mm)	35	50	Up to 300 mm
Aperture	F1.7-16	F1.4-F16	Image is darker by 3 f-stops at maximum magnification
Angle of view	14.3° × 10.7°	14.5° × 10.9°	magnification factor and focal length dependant
Minimum object distance (mm)	350	500	100-185
Depth of field (mm)	1.7-16	1.4-16	μm range
Objective lens magnification factor	None	None	0.7 – 4.5
Eyepiece magnification factor	None	None	0.5
Price (€)	37	35	62
Standard delivery time	5 working days	2-4 weeks	5 working days
Supplier, shipping location	Mouser (Germany)	Seeed Studio (USA)	Mouser (Germany)
Delivery cost (€)	0	30	0
Total price for 2 lenses (€)	74	100	124

Fig. 48. Chosen microscopic lens

According to the comparison, the microscopic lens would be the best option with regards to having a high focal length that is adjustable, unlike the fixed focal length lenses. Moreover, the minimum object distance is much less for the microscopic lens, which is more suitable for mounting the system assembly on the translational stage available in the university. Another important factor to consider is the depth of field. The depth of field is simply the distance range between the object viewed under the microscope and the surroundings where the surrounding can still be viewed in focus. This is illustrated in Fig. 49. In other words, the depth of field should be enough for the microscope to view both the holtite socket and the PCB in focus. Referring to the tolerance value of ± 100 µm of depth difference between the holtite socket and the PCB hole, the depth of field of the microscopic lens should be bigger than that. The datasheet of the lenses does not provide values for the numerical aperture, which is used to calculate the depth of field, and hence it is unknown whether the depth of field of the lens can focus both the holtite socket and PCB surface**.** Thus, during the design of the stereo vision system, this factor is to be taken into consideration, where a magnification factor ensuring a depth of field that focuses both the hole and socket would be chosen.

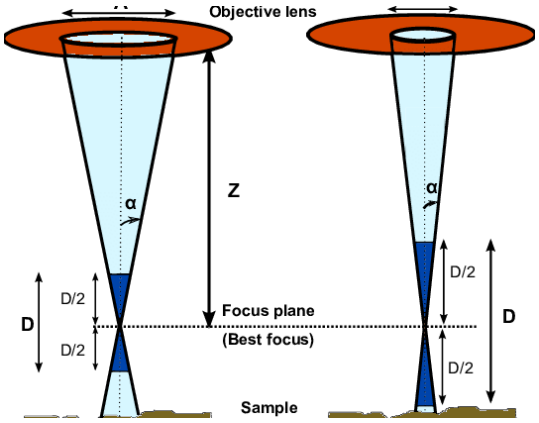


Fig. 49. Depth of field

4.3.1. Focal length

The microscope's objective lens has a range of zoom ratios, and thus the focal length changes as the magnification ratio of the objective lenses do. At maximum magnification, the focal length is 300 mm; however, the focal length value is not provided by the manufacturer for different magnification factors. Moreover, the focal length value is shown for the microscopic lens according to the size of the eyepiece and not the camera sensor. The eyepiece lens has a measured circular diameter of 14 mm; however, the camera sensor has a width of 6.287 mm, as shown in Table 5. This means the focal length of the microscopic lens when integrated with the camera would be reduced. When the microscope is integrated with the camera, the focal length of the total system is calculated below. From the manufacturer, the minimum working distance which is required at maximum magnification is 100 mm. Thus, the following data is given:

- Eyepiece magnification of 0.5
- Objective lens magnification of 4.5
- Total magnification (m) is thus 2.25
- Working distance of 100 mm between the microscope and the object to be viewed.

Fig. 50. Focal length and working distance relationship [30]

$$FOV_h = \frac{H}{m_{tot}} = \frac{6.287}{2.25} = 2.79\ mm \tag{4.3.1}$$

$$\mathrm{f} = m_{tot} \times w_d = 2.25 \times 100 = 225\ mm \tag{4.3.2}$$

$$\alpha_h = 2 \times \tan^{-1} \frac{FOV_h}{2 \times w_d} = 2 \times \tan^{-1} \frac{2.8}{2 \times 100} = 1.6\,° \tag{4.3.3}$$

Where:

- FOV is the horizontal field of view of the microscope.
- H is the width of the camera sensor.
- The total magnification is denoted by m, which is the multiplication of the objective magnification and the eyepiece's magnification.
- The focal length (f) of the lens.
- The horizontal α is angle of view.
- The working distance w_d is the distance where the object viewed will be at the sharpest focus.

The vertical field of view and the vertical angular of view is calculated by dividing the horizontal field of view and angle of view by the aspect ratio of the camera sensor.

The calculations presented above were calculated for the maximum objective lens magnification of 4.5. When the microscope was acquired, the working distances at different magnification factors were measured, and thus the fields of view, the focal length, and the angles of view were calculated and are shown below in Table 7.

Table 7. Calculated specifications for chosen microscope

Eye piece magnification	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Objective magnification	4.5	4.0	3.5	3.0	2.0	1.5	1	0.7
Total magnification	2.25	2	1.75	1.5	1	0.75	0.50	0.35
Working distance (mm)	100	102	103	105	110	115	125	165
Sensor width (mm)	6.287							
Aspect ratio	1.33							
Horizontal field of view (mm)	2.79	3.14	3.59	4.19	6.29	8.38	12.57	17.96
Focal length (mm)	225	204	180.25	157.5	110	86.25	62.5	57.75
Vertical field of view (mm)	2.1	2.36	2.69	3.14	4.72	6.29	9.43	13.47
Horizontal angular field of view (°)	1.6	1.77	2	2.29	3.27	4.17	5.76	6.23
Vertical angular field of view (°)	1.2	1.32	1.5	1.72	2.46	3.13	4.32	4.67

4.4.Other system parameters

4.4.1. Baseline and convergence angle

Setting a baseline for the system is constrained by the length of the camera cables which connect it to the synchronization board. This can be seen from Fig. 47. Each flex cable had a length of 150 mm. The baseline was found from the CAD design to be 112.52 mm, as shown in Fig. 37. Since the angle at which the microscopes are tilted towards each other is 14° degrees, they both converge at an angle of 28° as shown in Fig.50. The combination of the baseline and convergence angle provides a distance from the object to the microscopic lens at the required working distance of the microscope at the selected 1X objective lens zoom. It is crucial to ensure that the PCBs are placed at a distance equivalent to the working distance of the microscope at 1X objective zoom. This ensures that the PCB will be in focus.

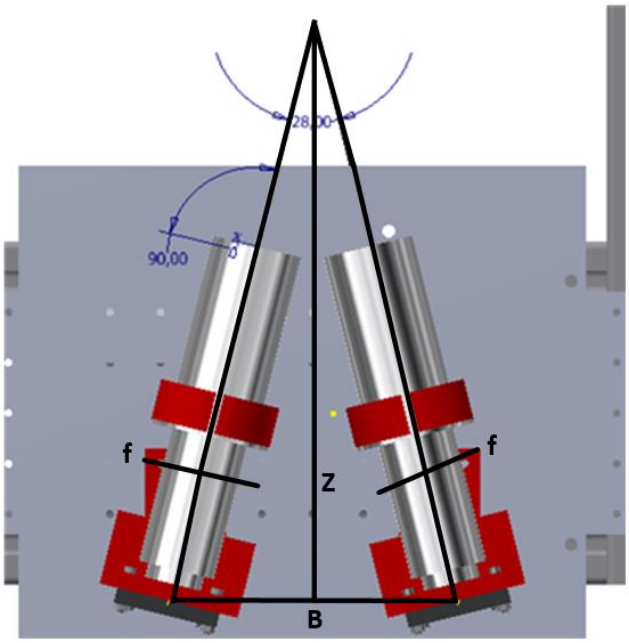


Fig. 51. Stereo system setup showing the baseline, focal length and convergence angle

Moreover, for sockets that have a large clearance of fit to the PCB holes, this would be noticeable. The common field of view for both microscopes at 1X zoom was measured to be around 2 mm, where both cameras view the same scene; this can be seen from Fig. 54. It can also be seen how the depth of field is not large, meaning that at even higher zoom ratios, focusing on both images will not be possible

4.4.2. light conditions and isolation

As concluded in the literature review, active stereo vision systems have improved accuracy over passive ones. Therefore, a light source was used for white light illumination. The room was dark, where only the light source was used. The light source was mounted on the x-y table and directed at the PCB to reduce the reflections caused by the shiny surface of both the PCB sockets and holes. The system setup is shown in Fig. 52.

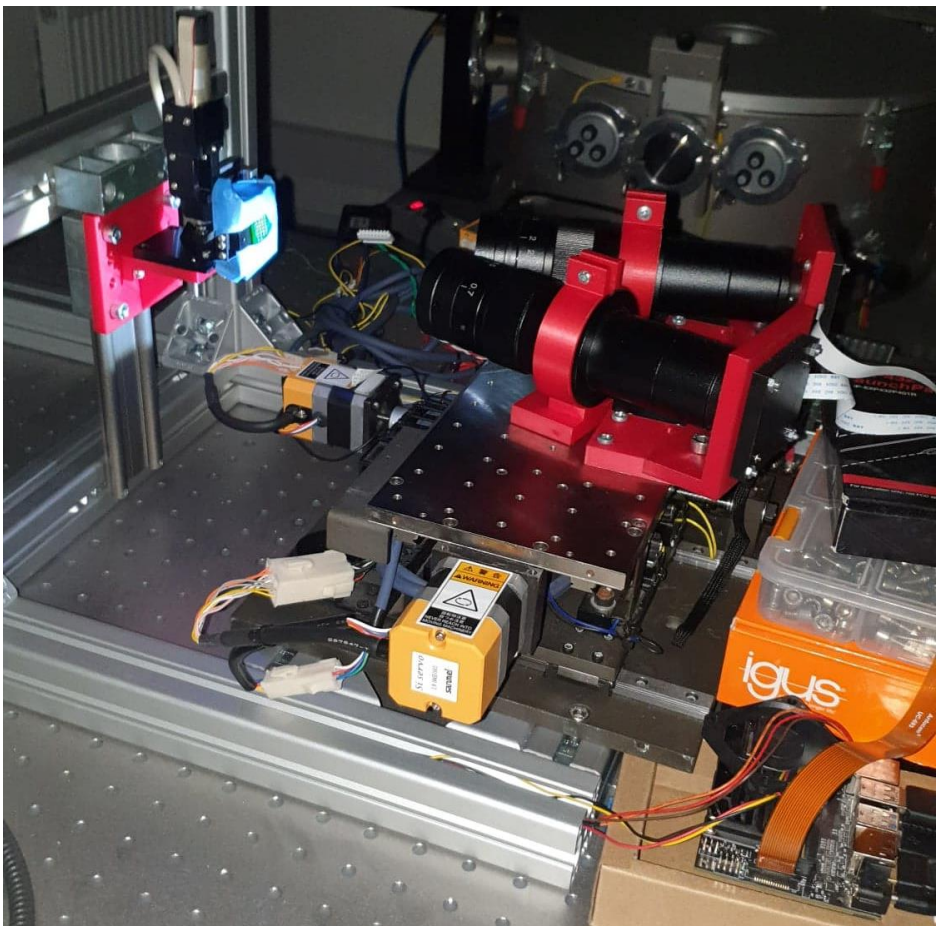


Fig. 52. System setup

4.5.Stereo rectification

In order to rectify the images, that is to have them both projected on two planes, where they will be aligned horizontally. The concept of image rectification is shown in Fig. 53. Horizontal alignment allows the objects in the images to have a horizontal shift of the points in the two camera frames, right and left respectively, this is known as disparity as shown in Fig. 16.

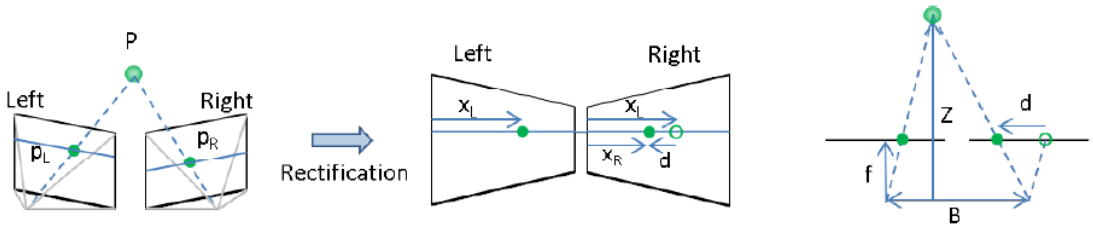


Fig. 53. Image rectification and epipolar geometry [31]

It is shown in Fig. 54, where a stereo pair of images is taken by both cameras, that the hole and socket in region A1 in the left image are the same hole and socket denoted as B1 in the right image. Similarly, the socket and hole pair A2 are the same as the socket and hole pair B2. It is also clear that the unrectified pairs have a vertical shift in their point positions when viewed by each camera and thus are not aligned. The horizontal shift is the disparity, which can only be calculated once both images are aligned vertically, which is known as image rectification.

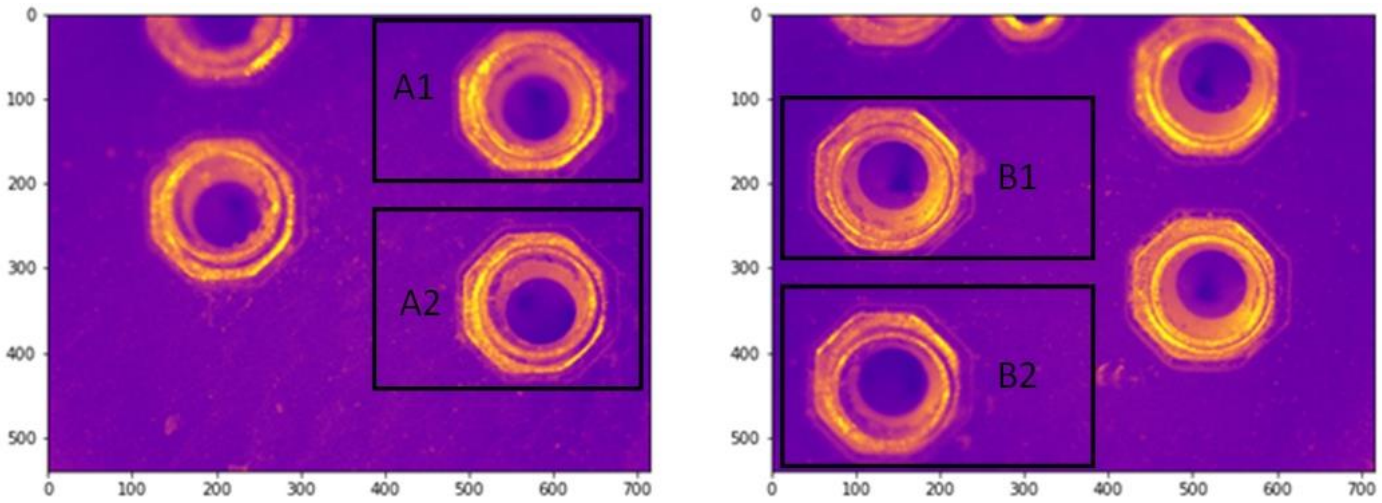


Fig. 54. Unrectified images

The rectification process of the images incorporated the use of stereoRectifyUncalibrated function in OpenCV. To rectify the images, the same feature points are required to be matched in both images. The traditional techniques use feature matching algorithms to scan for critical points in both images and matches the corresponding ones. Those matched points are then fed into a fundamental matrix used by the stereoRectifyUncalibrated function to calculate the transforms necessary to project both images onto a common plane where both images would become horizontally aligned. This function rectifies images taken by cameras that are not calibrated. The algorithm used for estimating the fundamental matrix is the FM_RANSAC algorithm which needs a minimum of 7 points in each image to be matched with the corresponding 7 points in the other image. Traditionally a Scale-invariant feature transform (SIFT) detector or other feature matching detectors are used to detect key points in both images and matching the best 7 points [32]. By incorporating deep learning-based object detection neural network, the points matching step can be avoided and instead, the co-ordinates of the bounding boxes of the detected sockets can be input into the RANSAC algorithm as the matching points. Moreover, for this specific application, feature matching detectors could face problems identifying the matching points in both images. The reason is that due to the light conditions and reflections of the shiny surfaces of the PCB holes and sockets, the pixel values of the same points in both images can change and thus it becomes difficult for the detector to know which points are the same in both images. However, with a deep learning-based object detection approach when the neural network is trained with enough pictures of different light intensities, it can adapt to various light conditions and successfully detect and localize the sockets and holes. Furthermore, traditional matching algorithms match only pixel values, however with DL approach, the localization coordinates have subpixel values and thus subpixel disparities can be calculated to obtain accurate depth measurement. This is important in this application, where the measured depth is in the range of micrometres. The deep learning-based object detection carried out on the unrectified images is shown in Fig. 55.

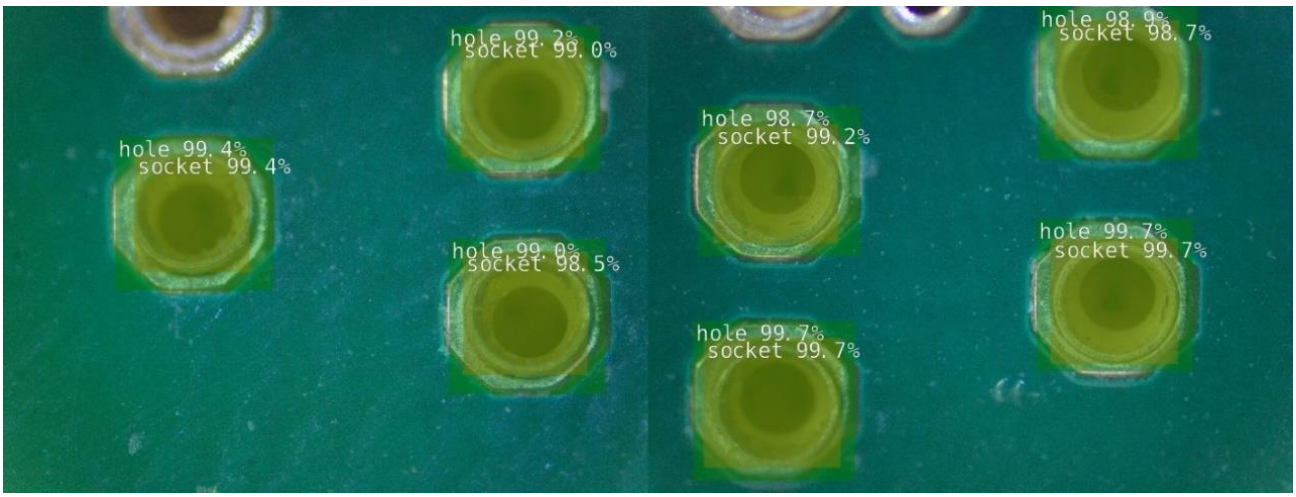


Fig. 55. DNN object detection of unrectified stereo image pair (left and right frames)

Stereo cameras calibration is usually done for the following reasons:

1. To obtain the camera's intrinsic parameters: its focal length and the location of the camera principal centres.
2. To obtain the translation and rotation vectors of each camera in order to know the tilt angles between each camera in a stereo setup
3. To remove lens distortion which usually occurs for wide angle lenses. The microscopic lenses have negligible distortion as lens distortion is usually significant in wide-angle lenses.

Moreover, calibrating the lenses was challenging because the checkerboard pattern used to calibrate cameras was not sharp when viewed by the microscope. The ink on the printed paper was magnified enough where the pattern was not consistent, and any form of dirt is picked up as noise by the cameras. Furthermore, due to the microscopic magnification, the edges of the checkboard pattern are magnified enough that they appear as distorted lines and not simple edge points this would lead to inaccurate calibration results that will significantly influence the measurement results. The difference between an appropriate checkerboard pattern for calibration and the pattern viewed by the microscope is shown in Fig. 56. Moreover, due to the mentioned reasons uncalibrated rectification was performed.

Fig. 56. Checkerboard pattern viewed by the microscopic lens (left) vs an ideal checkerboard pattern for accurate calibration (right)

The rectification results are shown in Fig. 57, where the images have become aligned, and thus, the disparity would only be the difference in the x-coordinate value of the matching points.

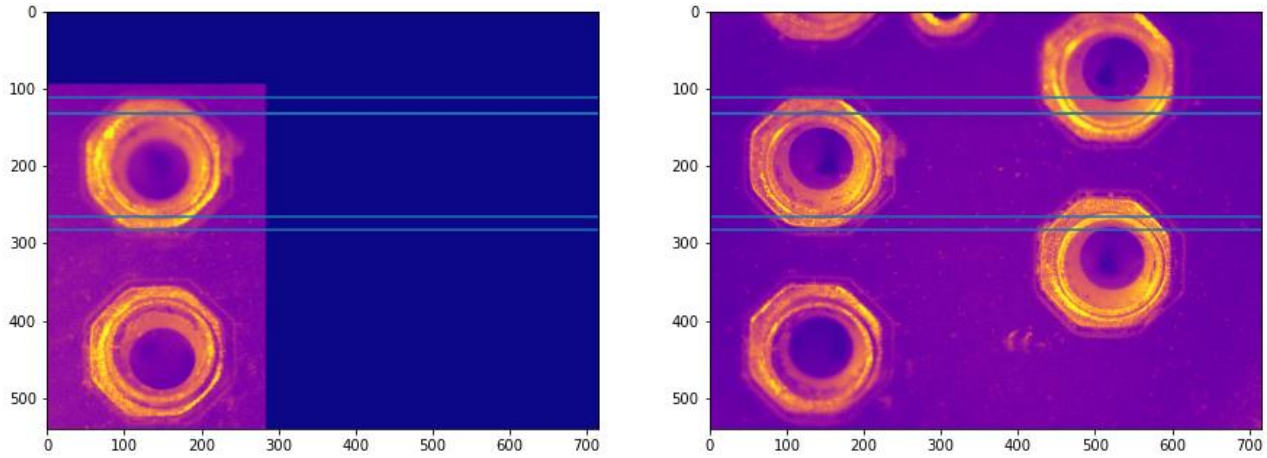


Fig. 57. Rectified images

After rectifying the images, a second network detection is proceeded where the localization coordinates can now be used to calculate the disparity. Since the sockets and holes have become aligned and are both on similar planes, the disparity for socket in the region denoted as A1 can simply be the difference in the x-coordinate value between a point on the socket in the left image and the corresponding point on the same socket in the right image. The same way the disparity can be computed for a hole in the region A1 by knowing its x-coordinate value in the left image frame and the same hole in the region B1 by knowing its x-coordinate value in the right image frame.

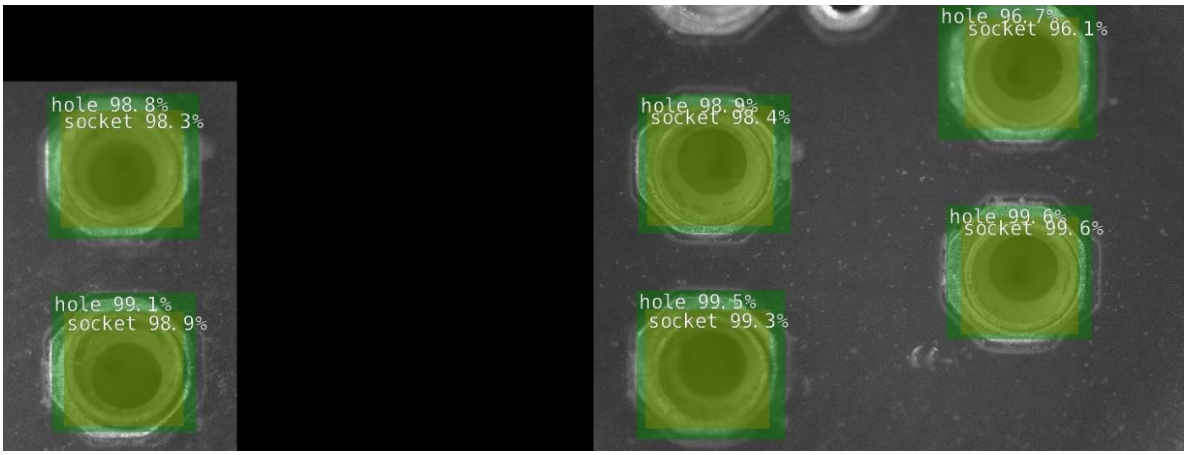


Fig. 58. Object detection performed on rectified images.

The algorithm for calculating the depth of press of a socket into its corresponding PCB hole is shown in the block diagram presented in Fig. 59.

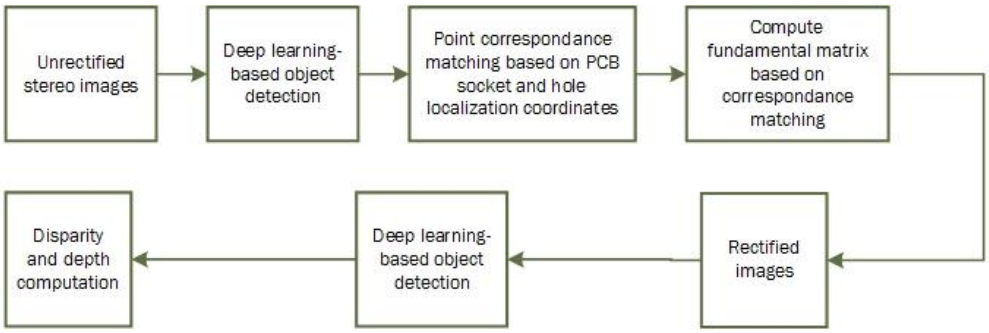


Fig. 59. Block diagram for pressed socket's depth calculation

4.5.1. Depth calculation

To calculate the distance from the cameras to the socket and hole, the equation can be written as follows [17]:

$$z = \frac{fB}{d} = \frac{fB}{x_L - x_R} \tag{4.1}$$

- $xL\ and\ xR$ are the horizontal coordinates of the same point in the left and right camera frames.
- The focal length in mm for 1X zoom was calculated to be 62.5 mm in Table 7.
- The baseline was 112.52 mm as found from the setup
- The focal length in pixels is 7118 pixels and is derived from the formula below
- W_{img} is the width of the image in pixels and is 716 pixels
- W_{sensor} is the width of the camera sensor which is 6.287 mm
- The disparity value obtained after the DL detection applied on the rectified images was 432.148 pixels for the measured hole and 432.156 for the socket.

$$f_{pixels} = \frac{f_{mm} \times W_{img}}{W_{sensor}} = \frac{62.5 \times 716}{6.287} = 7118\ pixels \tag{4.2}$$

Thus, the depth of the socket pressed into the PCB hole is calculated below as follows:

$$\Delta z = \frac{fB}{d_{hole}} - \frac{fB}{d_{socket}} = \frac{7118 \times 112.52}{432.148} - \frac{7118 \times 112.52}{432.156} = 34.31\ \mu m \tag{4.3}$$

10 sockets and their corresponding holes into which they were pressed, have had their depths measured and returned the values shown in Table 8.

Table 8. Socket's measured depth of fit from the PCB holes

socket number	1	2	3	4	5	6	7	8	9	10
depth measurement (µm)	34.31	35.2	46.3	32.8	34.1	46.9	56.8	42.6	31.2	29.4

4.6.Comparing the results to high precision topography system

A topography microscopic high precision system POLYTEC microsystem analyser was used to take depth measurement between the socket and the hole to have a reference to compare the results taken by the proposed system with. The same holes and sockets measured by the proposed system were measured by this high precision system. The setup is shown in Fig.60.

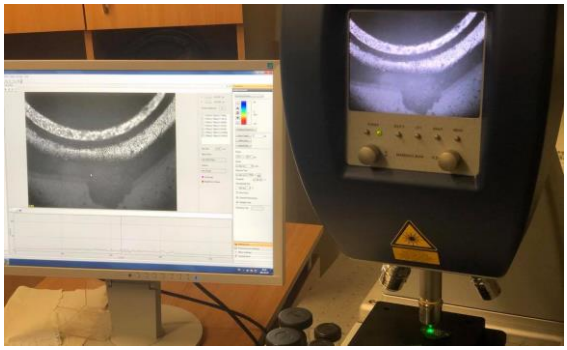


Fig. 60. High precision system setup

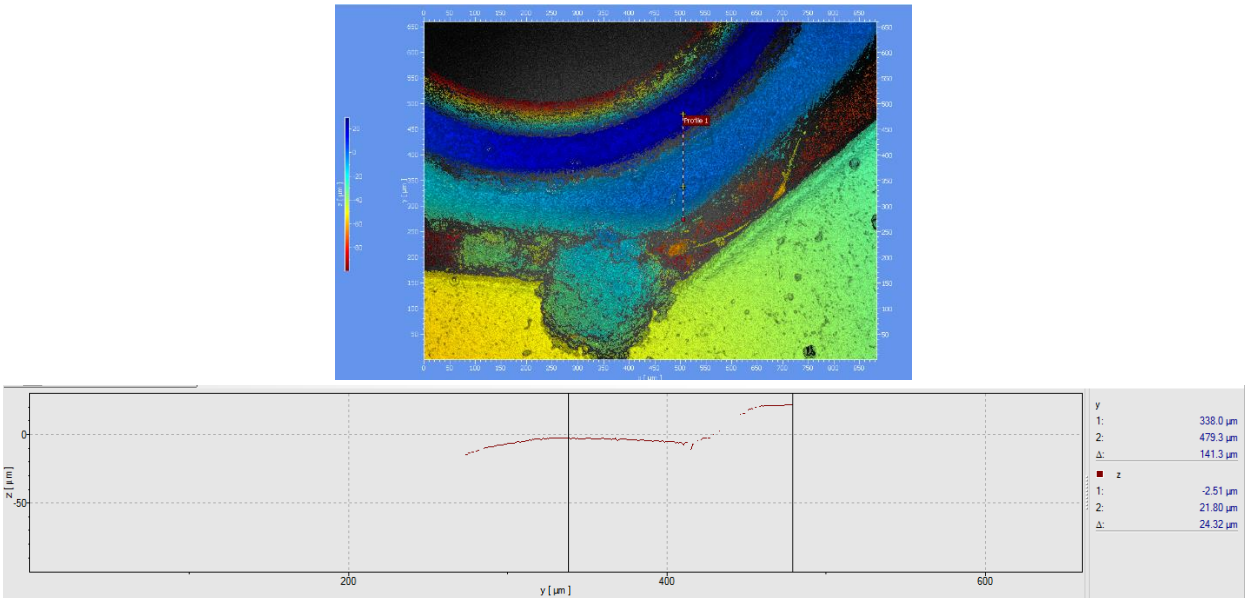


Fig. 61. Depth measurement of one socket using the POLYTEC system

The results of the measurements are shown in the table below:

Table 9 results taken by high precision system for reference

socket number	1	2	3	4	5	6	7	8	9	10
depth measurement (µm)	24.32	20.14	28.32	23.41	21.32	30.24	38.92	34.65	23.41	15.18

The following graph shows the error difference between the measured depth through the POLYTEC system and the proposed stereovision deep learning-based system. The maximum error of the ten taken measurements was found to be 18 µm. Few reasons may arise to having such error:

- Uncalibrated cameras could influence the stereo rectification process.
- The deep neural network's detection localization accuracy can influence some of the taken measurements.
- Mechanical setup may have had minute alignment issues that could also impact the measurement accuracy.

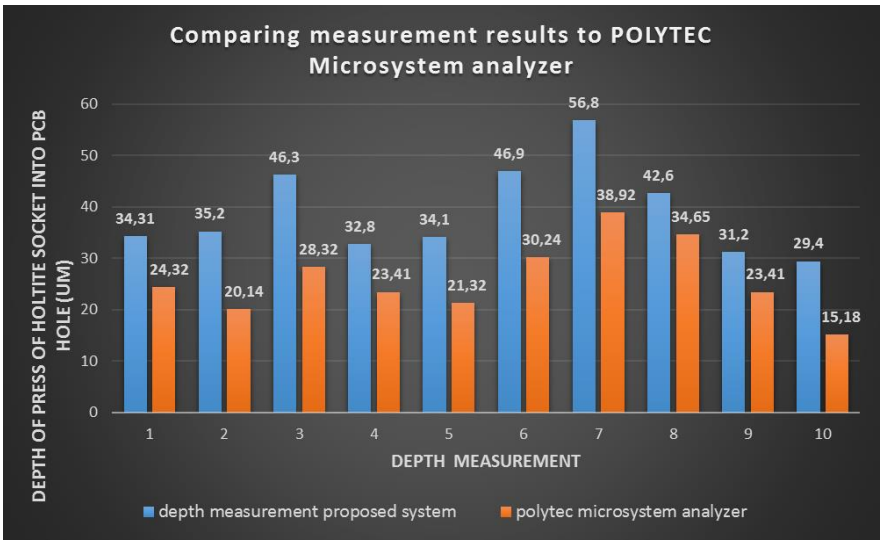


Fig. 62. Measurement comparison between proposed system and POLYTEC microsystem analyser

5. 3D printing and cost calculation

5.1.Print settings

in order to 3D print the designed parts, the Prusa I3 MK3S FDM 3D printer was used. The printing process consisted of 3 main parts: pre-processing, printing and postprocessing. One part, which is the camera housing, is used as an example to show the procedures of optimizing the print settings in order to print it in a smooth manner that ensures the camera's PCB would not be in contact with rough plastic edges, which could affect the PCB's copper traces and increase the risk of exposing the camera's PCB to electrostatic discharge.

5.1.1. Pre-processing

Pre-processing the parts consisted of converting the CAD model generated in Autodesk Inventor of each designed part into STL format. The 3D printing software uses STL format to interpret the geometry of the part to be printed. The STL file should have a good mesh in order for the printer to interpret the part's geometry smoothly. For example, for circular shapes, a rough mesh would be hard to approximate a circular shape, and it could become more of a hexagonal shape. Another reason for having a smooth mesh is avoiding electrostatic discharge, as mentioned above. Before converting the model to an STL file, the aspect ratio of the mesh is chosen to be low with a value of one, and the maximum edge length is chosen to be 0.5 % of the total object's length. This increases the number of polygons that the mesh consists of and thus increases the smoothness and approximation of the model. The number of polygons constituting the mesh is known as the Facets, which is 138 782 due to choosing a small edge length. The parameters are illustrated in Fig. 63. and Fig. 64.

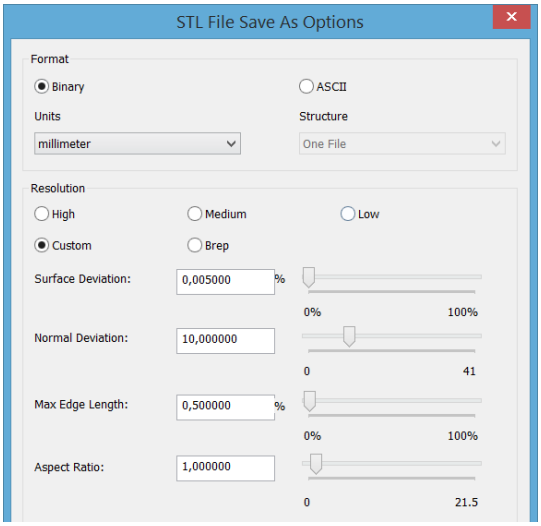


Fig. 63. STL mesh settings

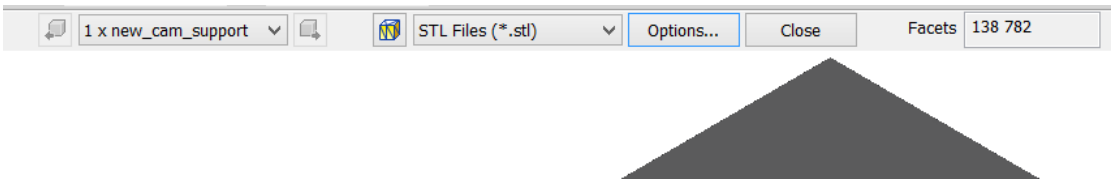


Fig. 64. Number of Facets generated

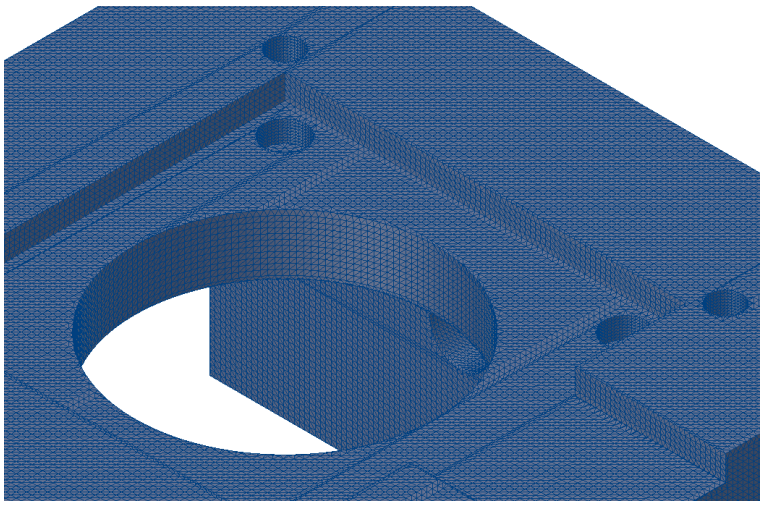


Fig. 65. Resulting smooth mesh

The generated mesh shown above is very smooth, and thus, the following step proceeds. Orienting and slicing the model is the following step where Prusa Slicer software is used to achieve that. The orientation is chosen such that the surface onto which the cameras electronic board is mounted is printed smoothly without rough edges and surfaces. Moreover, the print orientation is shown in Fig. 66 and Fig. 67, where the camera mounting surface is a final layer, and the cover is supported from the bottom. This orientation increases the printing time and material consumption and, as a result, the production cost. However, it is crucial to have the camera's board mounted onto a smooth surface, where if the mounting surface were supported, the printing time and costs would be reduced, but the surface would be very rough. The infill density for the print was chosen to be 50 % with a cubic infill pattern. The cubic pattern provides strength in three dimensions and is also the fastest three-dimensional pattern to print.

Moreover, the infill density is relatively high to ensure enough material is in contact with the fasteners that fix the cover to other design parts. After the model is sliced, the G-codes are generated, and the second step, which is the printing step, is executed. The total printing time for this part is 5 hours and 41 minutes. After the part has finished printing, postprocessing work is required to remove the supports and grinding sharp edges.

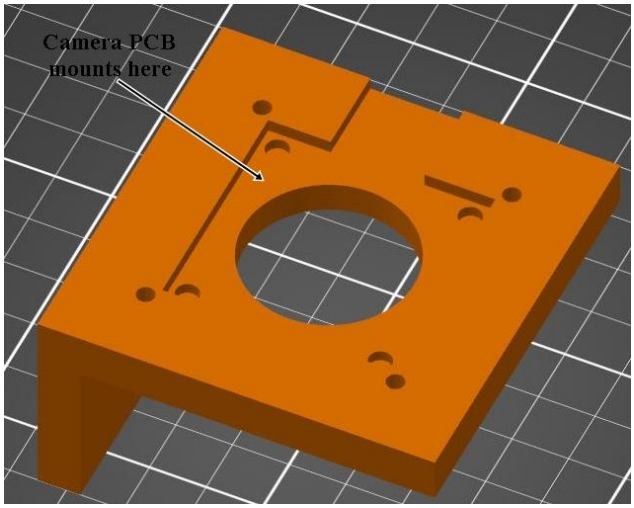


Fig. 66. Print orientation for the camera cover

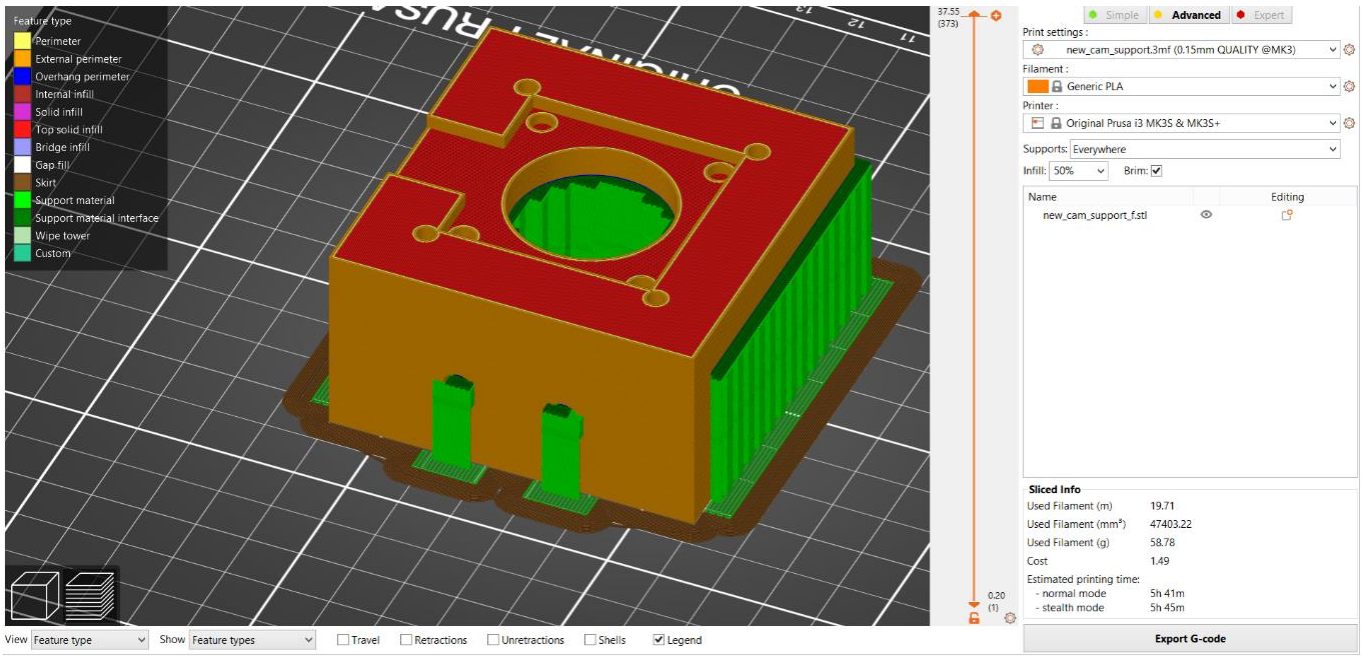


Fig. 67. Print slicing

5.2.System cost calculation

It is necessary to evaluate the costs of the individually 3D printed parts in this project for the necessary attachments for the HQ Raspberry Pi camera, microscopic lens, and the PCBs.

5.2.1. Manufacturing time calculations

The manufacturing time consists of the following:

1. Preparation for printing (adjusting of print parameters and setting, preheating the bed and setting up the actual printer)
2. Actual printing time.
3. Post-processing works (removal of the model from the plate, removal of support.)

5.2.2. Manufacturing (printing) price calculations:

Production costs

It was assumed that pre-processing and post-processing work for 30 mins regardless of parts printing time. The equations for calculating the manufacturing costs are as shown below.

The production costs consist of material costs as well as manufacturing costs. Where manufacturing costs include the machine hourly rate which considers fixed costs, variable costs and labour costs. Fixed costs are depreciation costs, interest costs, and occupancy costs. Variable costs are energy costs, maintenance costs and tool costs.

$$\text{Production costs} = \text{Manufacturing costs} + \text{Material costs} \tag{5.1}$$

$$\text{Manufacturing costs} = \text{Printing time} \times \text{MHR} \tag{5.2}$$

Where MHR is the hourly machine rate and is calculated as follows [33]:

$$MHR = \frac{S_{dep} + S_{int} + S_{are} + S_{ene} + S_{main} + S_{tools} + S_{per} + S_{add}}{MWT} \quad (5.3)$$

In order to calculate the machine hourly rate, the following parameters are found according to the following references [34, 35, 36, 37]:

- The machine's price is 1000 Eur.
- Depreciation of the machine is over five years of service life.
- Interest is assumed to be 8 %.
- The space cost rate is 16 euros per month per square meter in Lithuania.
- Assuming an office area of 10 square meters
- Machine dimensions (250 x 250 x 210 mm).
- Space requirements of one printer is 0,3 square meters.
- Power is 0,12 kW.
- Efficiency 90 %.
- The energy cost is 0,134 Eur/kWh.
- Machine working time is approximately 1500 hours (8hrs shift a day, 5 days a week, 9 months a year).
- Employee costs per hour in Lithuania is 6 euros.
- Effective working time for the employer for pre-printing, post-printing and checking print is approximated to be 30 minutes regardless of the printing time of the model.
- The employee effectively works with the print for 30 minutes with 6 euros per hour.
- Employee costs are added separately and are not included in the machine hourly rate calculation as the employee does not work while the printing process is occurring.

$$S_{dep} = \frac{\text{procurement value}}{\text{service life in years}} = \frac{1000}{5} = €200 \quad (5.4)$$

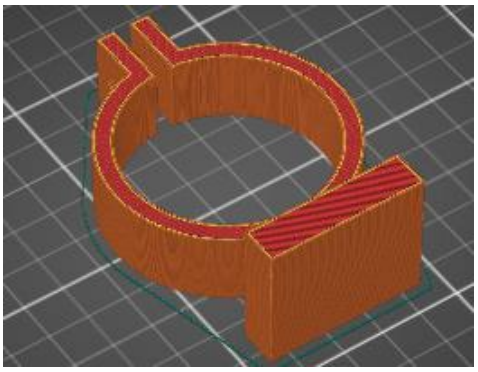
$$S_{int} = \frac{2}{3} \times \text{procurement value} \times \text{interest } \% = \frac{2}{3} \times 1000 \times 0.08 = €54 \quad (5.5)$$

$$S_{are} = \text{space cost rate} \times 12 \times \text{occupancy area } (m^2) = 16 \times 12 \times 0.3 = €57.6 \quad (5.6)$$

$$S_{ene} = \max \text{ machine power} \times \text{efficiency} \times \text{energy costs} \times MWT = 0.12 \times 0.9 \times 0.134 \times 1500 = €21.708 \quad (5.7)$$

One part of the design is considered as an example of calculating the manufacturing costs. The part is shown in Fig. 68, where its printing time and filament length used and consequently the mass to calculate the price are presented. The calculations are shown below. Moreover, the cost of other parts of the assembly are calculated the same way.

$$MHR = \frac{200 + 54 + 57.6 + 21.71}{1500} = €0,22 \quad (5.8)$$



Sliced Info

Used Filament (m)	10.05
Used Filament (mm^3)	24177.28
Used Filament (g)	29.98
Cost	0.76
Estimated printing time:	
- normal mode	2h23m
- stealth mode	2h23m

Fig. 68. microscope holder printing time and filament used

Thus, the manufacturing costs of the holder of the microscope is calculated as shown below:

$$\text{Manufacturing cost} = \text{printing time} \times \text{MHR} + \text{employee working time} \times \text{wage per hour}$$
$$= (2.5 \times 0.22) + (0.5 \times 6) \ = € 3.55 \quad (5.9)$$

Calculating material costs:

One kilogram of PLA material costs 25 euros, and therefore, the price of the printed part based on its mass is calculated below.

Table 10. Material cost of the microcope holder

Part	Material	Mass(g)	Price (€)
microscope holder	Prusa PLA	29.98	0,75

$$\text{Total production cost} = 3.55 + 0.75 = €4.3 \quad (6)$$

Thus, the total production costs of the microscope holder are about €4.3. Finally, the total production costs of all the 3D printed parts are shown in Table 11, where the total production cost of all parts is estimated to be €44.1.

Table 11. Total production costs of the manufactured components.

Part Names	No. of parts	Printing time (h,m)	Filament mass (g)	Machine Hourly Rate (€)	Manufacturing cost (€)	Total Production cost (€)
Microscope-holder	2	2hr 23 m	29.98	0.22	7.1	8.6
Z-Bracket	1	0hr 51 m	11.92	0.22	6.37	6.96
Camera support	2	5hr 41 m	58.8	0.22	8.5	11.44
Camera adapter	2	1hr 18 m	21.1	0.22	6.59	7.64
Actuator adapter	1	3hr 13	39.93	0.22	7.43	9.43

The total system prototype costs, including the prices of the standard parts (including delivery costs and VAT) and the manufactured 3D printed parts, are summarized in Table 12.

Table 12. Total production costs of the system setup

Part name	No. of item	Cost (€)
Nvidia Jetson Nano board	1	100
Microscopic lenses	2	74
Stereo vision camera setup	1	240
3D printed parts	5	44.1
Total price	1	458.1

The total system's production costs are estimated to be 459 euros, where the manufactured parts account for only 9.6 % of the total production costs.

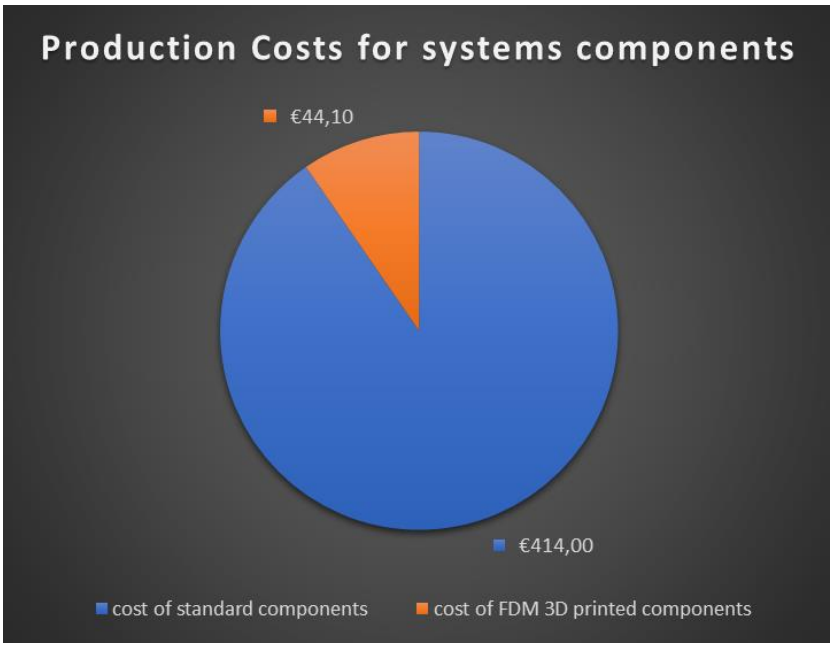


Fig. 69. Estimated production costs for the proposed system

Conclusions

A stereo vision-based depth measurement system has been developed. The system incorporates deep learning-based object detection neural network. The network used was SSD MobileNet V2, which is the fastest network to run on the embedded device Nvidia Jetson Nano. The DNN is used to classify and localize both the sockets and holes. The network is incorporated because the sockets can deform after pressing and lose their regular circular shape. Therefore, traditional image processing techniques used for classifying shapes would have problems identifying and localizing the sockets. The network is also used to simplify the correspondence matching process used for stereo images rectification. Furthermore, the system provides a cost-effective solution of incorporating relatively low-cost cameras, the Raspberry PI HQ cameras, which were released in 2020.

1. A deep learning-based object detection network based on SSD MobileNet V2 was used to classify and localize the PCB holes and holtite sockets. The network had a detection time of 65 ms per images containing two pairs of holtite sockets and holes.
2. A mechanical design was executed to set up the proposed system, where the design relied on additive manufacturing FDM 3D printing technique that was rigid enough for the proposed system. The stress values were negligible, and the maximum deformation of the assembly components was 0.35 μm.
3. Stereo vision depth measurement was developed with the integration of deep learning-based object detection. According to the samples measured, the system's measurement accuracy had a maximum error of 18 μm for the measurement sample taken of 10 measurements. The error is obtained by comparing the measured values to those measured by the high precision POLYTEC microsystem analyser.
4. The total system cost is around €500, which is subject to an increase with further development. However, it is relatively cheaper than similar depth measurement systems incorporating industrial cameras and high precision microsystem analysers.

List of references

1. Everything you need to know about Visual Inspection with AI. [online]. [viewed 2021-03-12]. Available from: https://nanonets.com/blog/ai-visual-inspection/.
2. KUJAWIŃSKA, A. - VOGT, K. Human factors in visual quality control. In Management and Production Engineering Review . 2015. Vol. 6, no. 2, p. 25–31.
3. Automatic Inspection Systems vs Human Visual Inspection. [online]. [viewed 2021-04-12]. Available from: https://www.eines.com/automatic-inspection-systems-human-visual-inspection.
4. O'MAHONY, N. et al. Deep Learning vs. Traditional Computer Vision. In Advances in Intelligent Systems and Computing . 2020. Vol. 943, no. Cv, p. 128–144.
5. What is Deep Learning and How does it work? | Towards Data Science. [online]. [viewed 2021-04-12]. Available from: https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac.
6. KHALIL, R.A. et al. Deep learning in Industrial Internet of Things: Potentials, challenges, and emerging applications. In arXiv . 2020. p. 1–21.
7. RIGUZZI, F. et al. An Introductory Review of Deep Learning for Prediction Models With Big Data. In Frontiers in Artificial Intelligence | www.frontiersin.org [online]. 2020. Vol. 3. Available from: www.frontiersin.org.
8. A Beginner's Guide to Neural Networks and Deep Learning | Pathmind. [online]. [viewed 2021-04-22]. Available from: https://wiki.pathmind.com/neural-network#element.
9. MA, Y. et al. Background augmentation generative adversarial networks (BAGANs): Effective data generation based on GAN-augmented 3D synthesizing. In Symmetry . 2018. Vol. 10, no. 12.
10. REDDY, S.V.G. et al. Optimization of deep learning using various optimizers, loss functions and dropout. In International Journal of Innovative Technology and Exploring Engineering . 2018. Vol. 8, no. 2S, p. 272–279.
11. Cross-Entropy Loss Function. A loss function used in most.. | by Kiprono Elijah Koech | Towards Data Science. [online]. [viewed 2021-04-23]. Available from: https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e.
12. Understanding Backpropagation Algorithm | by Simeon Kostadinov | Towards Data Science. [online]. [viewed 2021-04-23]. Available from: https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd
13. SOYDANER, D. A Comparison of Optimization Algorithms for Deep Learning. In . 2020. no. May.
14. KESKAR, N.S. - SOCHER, R. Improving generalization performance by switching from ADAM to SGD. In arXiv . 2017. no. 1.
15. SANTIAGO TELES DE MENEZES, R. et al. Object Recognition Using Convolutional Neural Networks. In Recent Trends in Artificial Neural Networks - from Training to Prediction [online]. IntechOpen, [viewed 2021-05-09]. Available from: https://www.intechopen.com/books/recent-trends-in-artificial-neural-networks-from-training-toprediction/object-recognition-using-convolutional-neural-networks.
16. ABOALI, M. et al. Review on three dimensional (3-D) acquisition and range imaging techniques. In International Journal of Applied Engineering Research . 2017. Vol. 12, no. 10, p. 2409–2421.
17. BERNHARDT, S. et al. Robust dense endoscopic stereo reconstruction for minimally invasive surgery. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial

Intelligence and Lecture Notes in Bioinformatics) . 2013. Vol. 7766 LNCS, no. October 2012, p. 254–262.

18. STROPPA, L. - CRISTALLI, C. Stereo Vision System for Accurate 3D Measurements of Connector Pins' Positions in Production Lines. In Experimental Techniques . 2017. Vol. 41, no. 1, p. 69–78.
19. HUBER, M. et al. Cubic range error model for stereo vision with illuminators. In Proceedings - IEEE International Conference on Robotics and Automation . 2018. p. 842–848.
20. Understanding embedded vision - Allied Vision. In [online]. [viewed 2021-04-16]. Available from: https://www.alliedvision.com/en/productsfolder/embedded-vision/understanding-embedded-vision.html.
21. Embedded Vision Moves Into the Clinical Realm Features BioPhotonics. [online]. [viewed 2021-04-16]. Available from: https://www.photonics.com/Articles/Embedded_Vision_Moves_Into_the_Clinical_Realm/a62362.
22. What are the benefits of CMOS based machine vision cameras vs CCD? [online]. [viewed 2021-04-16]. Available from: https://www.1stvision.com/machine-vision-solutions/2019/07/benefits-of-cmos-based-machine-vision-cameras-vs-ccd.html.
23. Developments in Machine Vision Camera Interfaces | 2020-11-30 | Quality Magazine. In [online]. [viewed 2021-04-16]. Available from: https://www.qualitymag.com/articles/96288-developments-in-machine-vision-camera-interfaces.
24. Jetson Nano | NVIDIA Developer. [online]. [viewed 2021-04-16]. Available from: https://developer.nvidia.com/embedded/jetson-nano.
25. Jetson Nano: Deep Learning Inference Benchmarks | NVIDIA Developer. [online]. [viewed 2021-05-10]. Available from: https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks.
26. PRUSA, J. Prusament PLA Technical Data Sheet. In . 2018. Vol. Version 1., p. 9–10.
27. Camarray – Arducam 12MP IMX477 Synchronized Stereo Camera Bundle Kit - Arducam. [online]. [viewed 2021-04-29]. Available from: https://www.arducam.com/docs/camera-for-jetson-nano/multiple-cameras-on-the-jetson-nano/camarray-arducam-12mp-synchronized-stereo-camera-bundle-kit/.
28. 4K Camera for NVIDIA Jetson Xavier NX / Nano developer kit. [online]. [viewed 2021-04- 29]. Available from: https://www.e-consystems.com/nvidia-cameras/jetson-xavier-nx-cameras/4k-camera-for-jetson-nano-xavier-nx.asp.
29. 300X Microscope Lens for Raspberry Pi High Quality Camera with C Mount - Seeed Studio. In [Online]. [viewed 2021-04-29]. Available from: https://www.seeedstudio.com/Microscope-Camera-300X-C-Mount-Lens-for-Raspberry-Pi-High-Quality-Camera-p-4627.html.
30. Understanding Focal Length and Field of View | Edmund Optics. [online]. [Viewed 2021-05-09]. Available from: https://www.edmundoptics.eu/knowledge-center/application- notes/imaging/understanding-focal-length-and-field-of-view/.
31. BERNHARDT, S. et al. Robust dense endoscopic stereo reconstruction for minimally invasive surgery. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2013. Vol. 7766 LNCS, no. October 2012, p. 254–262.

32. KO, H. et al. Robust uncalibrated stereo rectification with constrained geometric distortions (USR-CGD). In Image and Vision Computing [online]. 2017. Vol. 60, p. 98–114. Available from: http://dx.doi.org/10.1016/j.imavis.2017.01.001.
33. Juzėnas Egidijus, Rimašauskas Marius, CALCULATION OF PRODUCT PRICE. Methodical instructions for independent work in machine production technology [online]. KTU Publishing House "Technology", 2013, pp. 36-37 [viewed 2021-05-2].e.ISBN 978-609-02-0941-7. Available from: https://www.ebooks.ktu.lt/eb/1205/masinu-gamybos-technologijos-savarankiskumeto diniai-darbu-nurodymai/
34. Original Prusa i3 MK3S 3D printer. [online]. [viewed 2021-05-2]. Available from: https://shop.prusa3d.com/en/3d-printers/181-original-prusa-i3-mk3s-3d-printer.html#.
35. Office real estate prime rent in Vilnius 2019 Statista. [online]. [viewed 2021-05-2]. Available from: https://www.statista.com/statistics/530141/office-real-estate-prime-rent-vilnius-lithuania-
36. Lithuania: household electricity prices 2020 | Statista. [online]. [viewed 2021-05-2]. Available from: https://www.statista.com/statistics/418098/electricity-prices-for-households-in-lithuania/.
37. Mechanical Engineer Average Salary in Lithuania 2021 - The Complete Guide. [online]. [viewed 2021-05-2]. Available from: http://www.salaryexplorer.com/salary-survey.php?loc=124&loctype=1&job=276&jobtype=3.11

Appendices

Programming Code

1. Camera initialization and taking stereo images

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 23:23:00 2021
@author: Ahmed Elatroush
"""
import cv2
import numpy as np
def gstreamer_pipeline(
    capture_width=4032,
    capture_height=3040,
    display_width=1432,
    display_height=540,
    framerate=13,
    flip_method=2,
):
    return (
        "nvarguscamerasrc ! "
        "video/x-raw(memory:NVMM), "
        "width=(int)%d, height=(int)%d, "
        "format=(string)NV12, framerate=(fraction)%d/1 ! "
        "nvvidconv flip-method=%d ! "
        "video/x-raw, width=(int)%d, height=(int)%d, format=(string)BGRx ! "
        "videoconvert ! "
        "video/x-raw, format=(string)BGR ! appsink"
        % (
            capture_width,
            capture_height,
            framerate,
            flip_method,
            display_width,
            display_height,
        )
    )

cam=cv2.VideoCapture(gstreamer_pipeline())
while True:
    ret, img=cam.read()
    cv2.imshow('rpi camera',img)
    if cv2.waitKey(1) == ord('y') :
        cv2.imwrite('img1.jpg',img)
        image = cv2.imread('img1.jpg')
        height, width = image.shape[:2]
        x_start=int(width/2)
        x_end=int(width)
        y_end=int(height)
        #split horizontally
        right_img = image[0:y_end, x_start:x_end]
        left_img = image[0:y_end, 0:x_start]
        cv2.imwrite(right_img.jpg',right_img)
   cv2.imwrite(left_img.jpg',left_img)
        break
cam.release()
cv2.destroyAllWindows()
```

2. Object detection based on the socket and holes model retrained on SSD Mobilenet v2

```
# -*- coding: utf-8 -*-
import jetson.inference
import jetson.utils

import argparse
import sys

# parse the command line
parser = argparse.ArgumentParser(description="Locate objects in a live
camera stream using an object detection DNN.",

formatter_class=argparse.RawTextHelpFormatter,
epilog=jetson.inference.detectNet.Usage() +
                                    jetson.utils.videoSource.Usage() +
jetson.utils.videoOutput.Usage() + jetson.utils.logUsage())

parser.add_argument("input_URI", type=str, default="", nargs='?', help="URI
of the input stream")
parser.add_argument("output_URI", type=str, default="", nargs='?', help="URI
of the output stream")
parser.add_argument("--network", type=str, default="ssd-mobilenet-v2",
help="pre-trained model to load (see below for options)")
parser.add_argument("--overlay", type=str, default="box,labels,conf",
help="detection overlay flags (e.g. --overlay=box,labels,conf)\nvalid
combinations are:  'box', 'labels', 'conf', 'none'")
parser.add_argument("--threshold", type=float, default=0.5, help="minimum
detection threshold to use")
parser.add_argument("--rect", type=str, default="", help="rectification
flag")
is_headless = ["--headless"] if sys.argv[0].find('console.py') != -1 else [""]

try:
        opt = parser.parse_known_args()[0]
except:
        print("")
        parser.print_help()
        sys.exit(0)

# load the object detection network
net = jetson.inference.detectNet(opt.network, sys.argv, opt.threshold)

# create video sources & outputs
input = jetson.utils.videoSource(opt.input_URI, argv=sys.argv)
output = jetson.utils.videoOutput(opt.output_URI, argv=sys.argv+is_headless)

# flag to indication if detection is for rectified or unrectified images.
rectification =opt.rect
# process frames until the user exits
open("detect_rectified.txt", "w").close()
open("detect_unrectified.txt", "w").close()
while True:
        # capture the next image
        img = input.Capture()

        # detect objects in the image (with overlay)
        detections = net.Detect(img, overlay=opt.overlay)
```

```
        # print the detections
        print("detected {:d} objects in image".format(len(detections)))

        if rectification == "true":
        #f = open("detect_unrectified.txt", "a")
                f = open("detect_rectified.txt", "a")
                for detection in detections:
                        f.write(str(detection))
                f.close()
        elif rectification == "false":
                f = open("detect_unrectified.txt", "a")
                for detection in detections:
                        f.write(str(detection))
                f.close()
        # render the image
        output.Render(img)

        # update the title bar
        output.SetStatus("{:s} | Network {:.0f} FPS".format(opt.network,
net.GetNetworkFPS()))

        # print out performance info
        net.PrintProfilerTimes()

        # exit on input/output EOS
        if not input.IsStreaming() or not output.IsStreaming():
                break
```

3. Image rectification and depth calculation

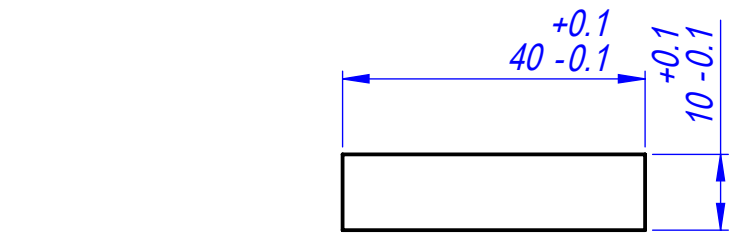
```
# -*- coding: utf-8 -*-
"""
Created on Sun May 2 28 02:53:57 2021

@author: Ahmed Elatroush
"""
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
import re

# stereo setup parameters

#baseline in mm
baseline=112.52
width =1432
#focal length in mm
focal_length_mm = 62.5
#sensor width in mm
s_w=6.287
# focal length conversion to pixels
focal_length_pix=((focal_length_mm *(width/2))/s_w)

img1 = cv.imread('left_1.jpg', cv.IMREAD_GRAYSCALE)
img2 = cv.imread('right_1.jpg', cv.IMREAD_GRAYSCALE)
```

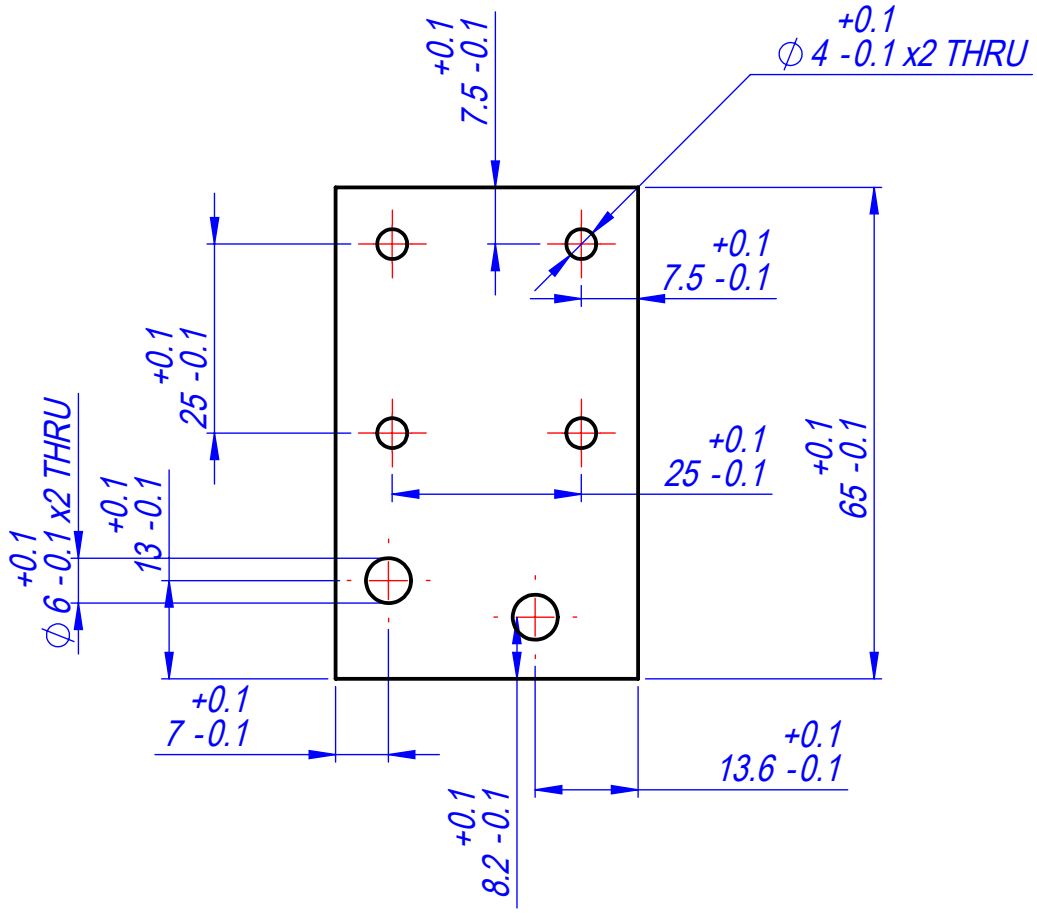
```
# bounding box coordinates of one socket obtained from the deep learning
based object detection network

#top, bottom, left and right coordinates of the bounding box.
#The values are manually input from the detection text file for a selected
socket
#This would be parsed automatically, but for the sake of proof of concept,
#The coordinates of one socket are input manually here.
#The detection coordinates file is shown in the appendix.

ls_x_sp=510.682
ls_y_sp=37.2002
ls_x_ep=646.352
ls_y_ep=168.578

rs_x_sp=73.9424
rs_y_sp=131.177
rs_x_ep=212.443
rs_y_ep=264.043

# height and width of bounding box, used to obtain 4 mid points to get a
total of 8 points
#for the fundamental matrix FM_RANSAC function
ls_height=131.378
ls_width=135.67

rs_height=132.866
rs_width=138.5

#mid points for bounding rectangles of the same socket in the left and right
# image frames.
x_mid_point_ls=ls_x_sp+(ls_width/2)
y_mid_point_ls=ls_y_sp+(ls_height/2)

x_mid_point_rs=rs_x_sp+(rs_width/2)
y_mid_point_rs=rs_y_sp+(rs_height/2)

# coordinates of 8 points of the same socket in the left and right frames ,
# which are used by the fundamental matrix to obtain image rectification.

start_point_ls = (ls_x_sp,ls_y_sp)
end_point_ls = (ls_x_ep,ls_y_ep)
mid_point_w_ls_t=(x_mid_point_ls,ls_y_sp)
mid_point_w_ls_b=(x_mid_point_ls,ls_y_ep)
mid_point_h_ls_t=(ls_x_sp,y_mid_point_ls)
mid_point_h_ls_b=(ls_x_ep,y_mid_point_ls)
start_point_shift_ls=(ls_x_sp,ls_y_ep)
end_point_shift_ls=(ls_x_ep,ls_y_sp)

start_point_rs = (rs_x_sp,rs_y_sp)
end_point_rs = (rs_x_ep,rs_y_ep)
mid_point_w_rs_t=(x_mid_point_rs,rs_y_sp)
mid_point_w_rs_b=(x_mid_point_rs,rs_y_ep)
mid_point_h_rs_t=(rs_x_sp,y_mid_point_rs)
mid_point_h_rs_b=(rs_x_ep,y_mid_point_rs)
start_point_shift_rs=(rs_x_sp,rs_y_ep)
end_point_shift_rs=(rs_x_ep,rs_y_sp)

# store the points in the mp_ls and mp_rs arrays to be used by the
fundamental
# matrix for image rectification.
```

```
# Compare unrectified images
fig, axes = plt.subplots(1, 2, figsize=(15, 10))
axes[0].imshow(img1, cmap="plasma")
axes[1].imshow(img2, cmap="plasma")
axes[0].axhline(250)
axes[1].axhline(250)
axes[0].axhline(120)
axes[1].axhline(120)
plt.suptitle("Original images")
plt.savefig("original_images.jpg")

plt.show()

# same socket's matching points in both left and right images
mp_ls=[(start_point_ls),(end_point_ls),(mid_point_w_ls_t),(mid_point_w_ls_b)
,(mid_point_h_ls_t),(mid_point_h_ls_b),(start_point_shift_ls),(end_point_shi
ft_ls)]
mp_rs=[(start_point_rs),(end_point_rs),(mid_point_w_rs_t),(mid_point_w_rs_b)
,(mid_point_h_rs_t),(mid_point_h_rs_b),(start_point_shift_rs),(end_point_shi
ft_rs)]

# Calculate the fundamental matrix for the cameras

mp_ls = np.int32(mp_ls)
mp_rs = np.int32(mp_rs)

fundamental_matrix, inliers = cv.findFundamentalMat(mp_ls, mp_rs,
cv.FM_RANSAC)


# uncalibrated Stereo rectification using the fundamental matrix and matchin
points
h1, w1 = img1.shape
h2, w2 = img2.shape
_, H1, H2 = cv.stereoRectifyUncalibrated(
    np.float32(mp_ls), np.float32(mp_rs), fundamental_matrix, imgSize=(w1,
h1)
)

# Undistort the images and save them
img1_rectified = cv.warpPerspective(img1, H1, (w1, h1))
img2_rectified = cv.warpPerspective(img2, H2, (w2, h2))
cv.imwrite("rectified_h_1.jpg", img1_rectified)
cv.imwrite("rectified_h_2.jpg", img2_rectified)

# concatenating the left and right images into one to show the rectification
results
combined_img = cv.hconcat([img1_rectified, img2_rectified])
cv.imwrite("opencv_rect_h.jpg", combined_img)
cv.imshow('opencv_Rect', combined_img)
height, width = img1_rectified.shape[:2]

# Draw the combined rectified images
fig, axes = plt.subplots(1, 2, figsize=(15, 10))
axes[0].imshow(img1_rectified, cmap="plasma")
axes[1].imshow(img2_rectified, cmap="plasma")
axes[0].axhline(131)
axes[1].axhline(131)
axes[0].axhline(265)
axes[1].axhline(265)
```

```
axes[0].axhline(111)
axes[1].axhline(111)
axes[0].axhline(131)
axes[1].axhline(131)
axes[0].axhline(283)
axes[1].axhline(283)

plt.suptitle("Rectified images")
plt.savefig("rectified_images.jpg")
plt.show()

# measure the disparity directly from the second detections coordinates
# again manually input from the detection text files.
disp_h=432.148
disp_s=432.156
# calculate the depth of the pressed socket into the hole

# depth into which the socket is pressed into the hole converted from mm to
micrometers
depth_of_press= 1000*(((focal_length_pix*baseline)/disp_h)-
((focal_length_pix*baseline)/disp_s))
print("depth of press of socket into hole", depth_of_press)

cv.waitKey()
cv.destroyAllWindows()
```

Technical Drawings



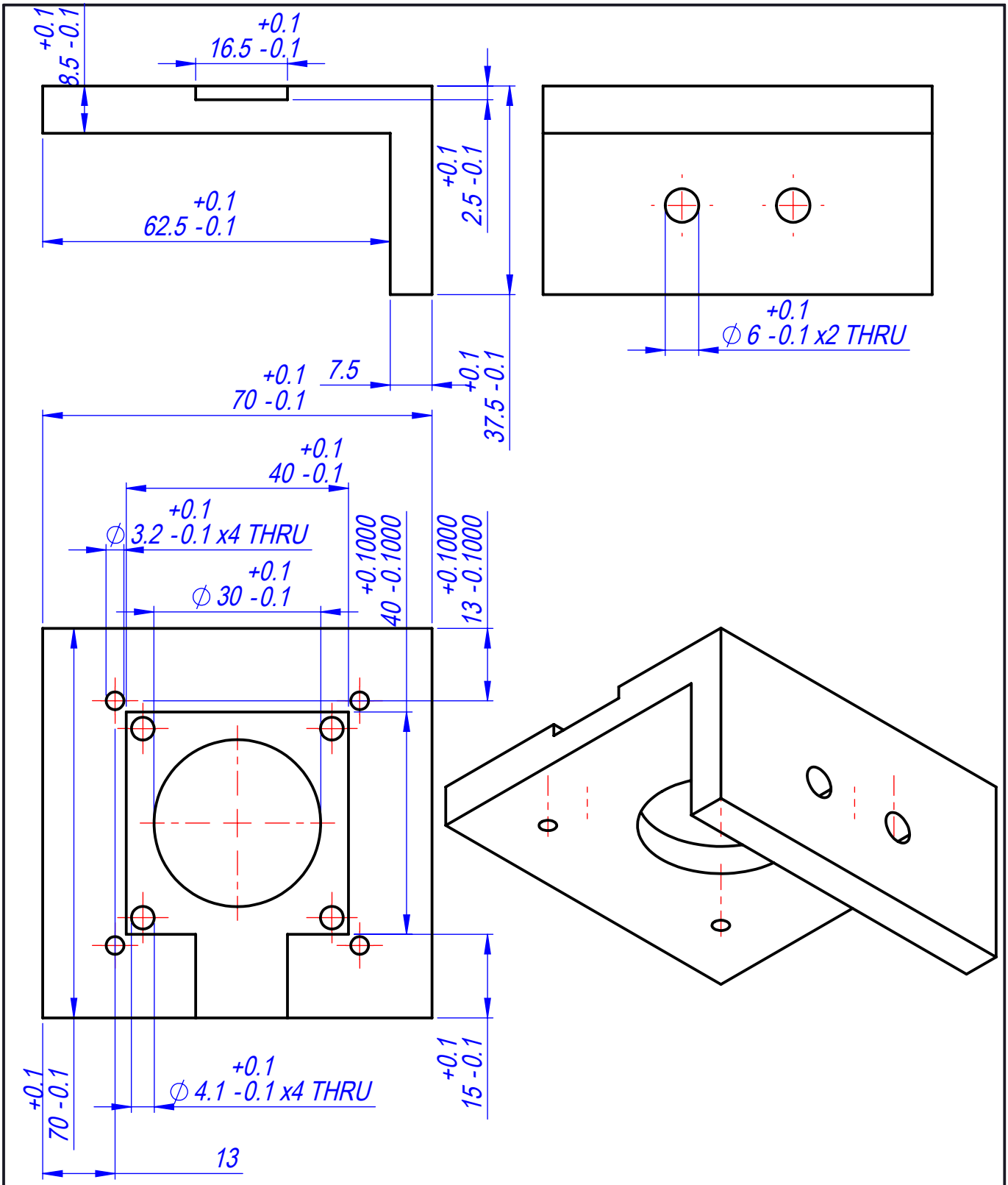


Ra 6.3

	File name	Additional information	Material	Scale
			Prusa PLA	1:1

Resp. department	Technical reference	Dokumento tipas	Document status
MIDF		Part drawing	Training

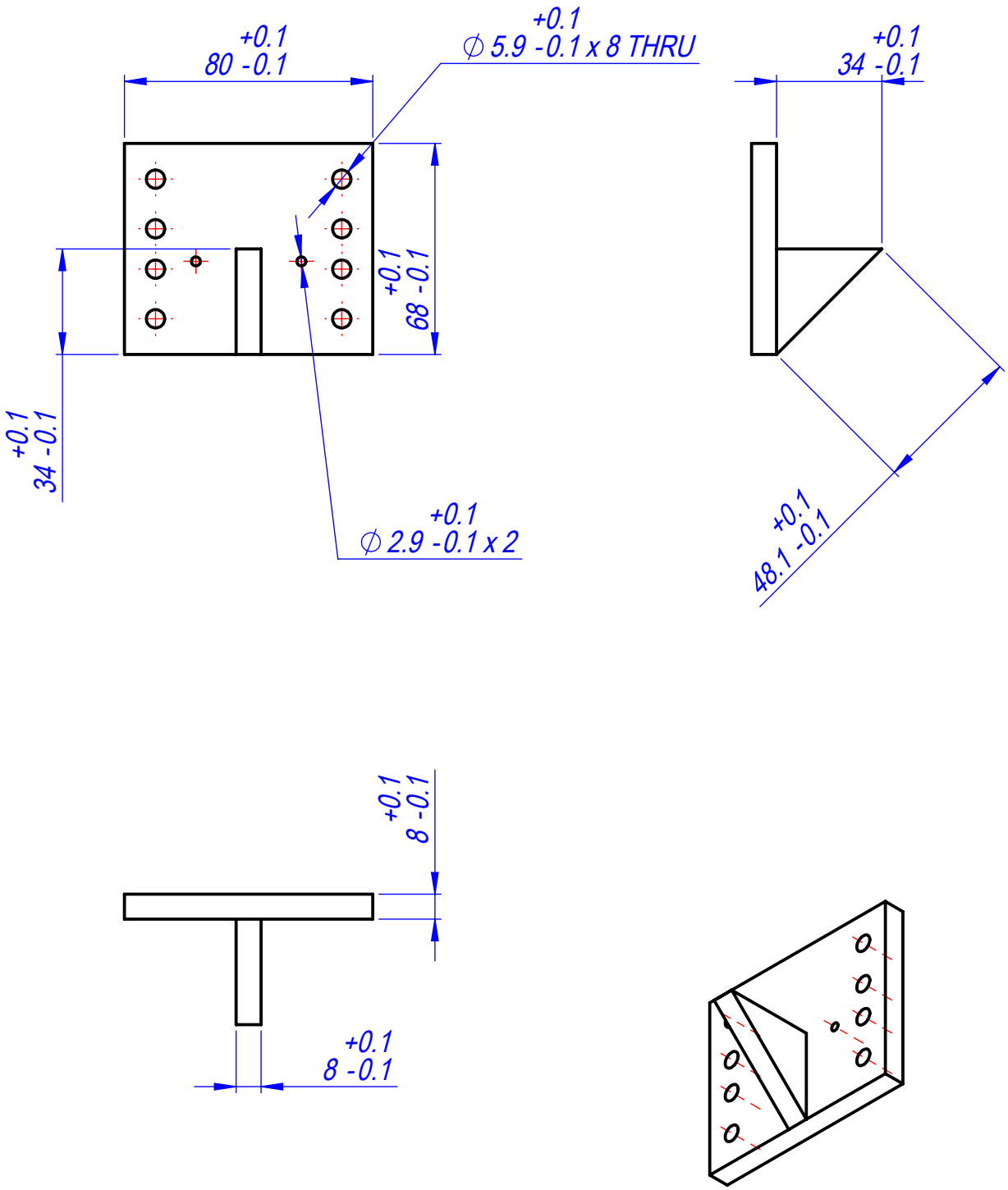
Legal owner	Created by	Title, Supplementary title	SVS-00.00.011			
ktu kauno technologijos universitetas 1922	Ahmed Elatroush	Adapter 1				
	Approved by		Rev.	Date	Lang.	Sheet
	Valdas Grigalunas		A	5/11/2021	En	1/1



	File name	Additional information	Material	Scale
			Prusa PLA	1:2

Resp. department	Technical reference	Dokumento tipas	Document status
MIDF		Part drawing	Training

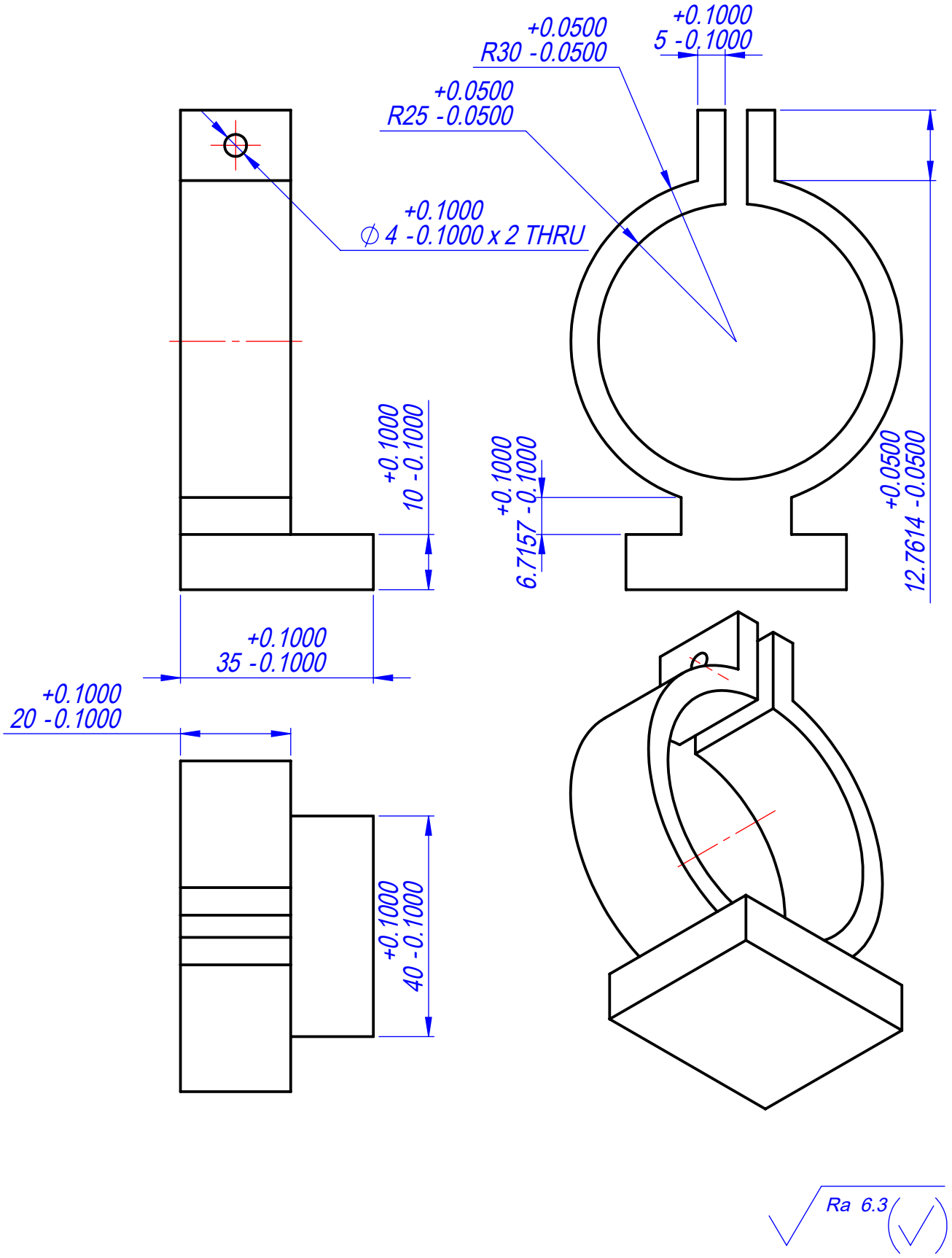
Legal owner	Created by	Title, Supplementary title				
ktu kauno technologijos universitetas 1922	Ahmed Elatroush	New cam support	SVS-00.00.012			
	Approved by		Rev.	Date	Lang.	Sheet
	Valdas Grigalunas		A	5/11/2021	En	1/1



	File name	Additional information	Material	Scale
			Prusa PLA	1:2

Resp. department	Technical reference	Dokumento tipas	Document status
MIDF		Part drawing	Training

Legal owner	Created by	Title, Supplementary title				
ktu kauno technologijos universitetas 1922	Ahmed Elatroush	Actutor attachement Z	SVS-00.00.016			
	Approved by		Rev.	Date	Lang.	Sheet
	Valdas Grigalunas		A	5/11/2021	En	1/1

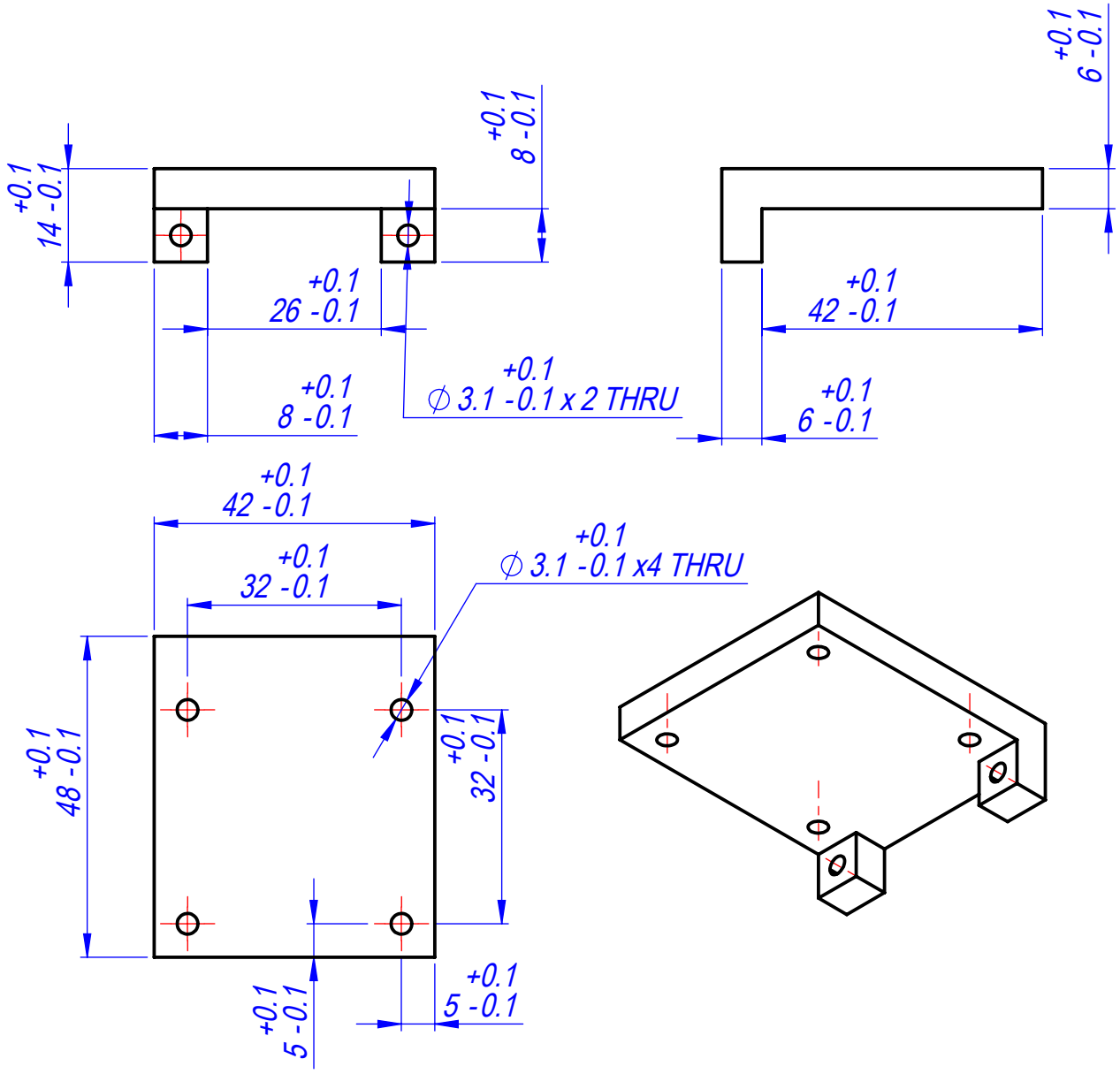


Ra 6.3

	File name	Additional information	Material	Scale
			Steel C45 LST EN 10083-1	1:1

Resp. department	Technical reference	Dokumento tipas	Document status
MIDF		Part drawing	Training

Legal owner	Created by	Title, Supplementary title	SVS-00.00.021			
ktu kauno technologijos universitetas 1922	Ahmed Elatroush	New holder uscope	Rev.	Date	Lang.	Sheet
	Approved by		A	5/11/2021	En	1/1
	Valdas Grigalunas					

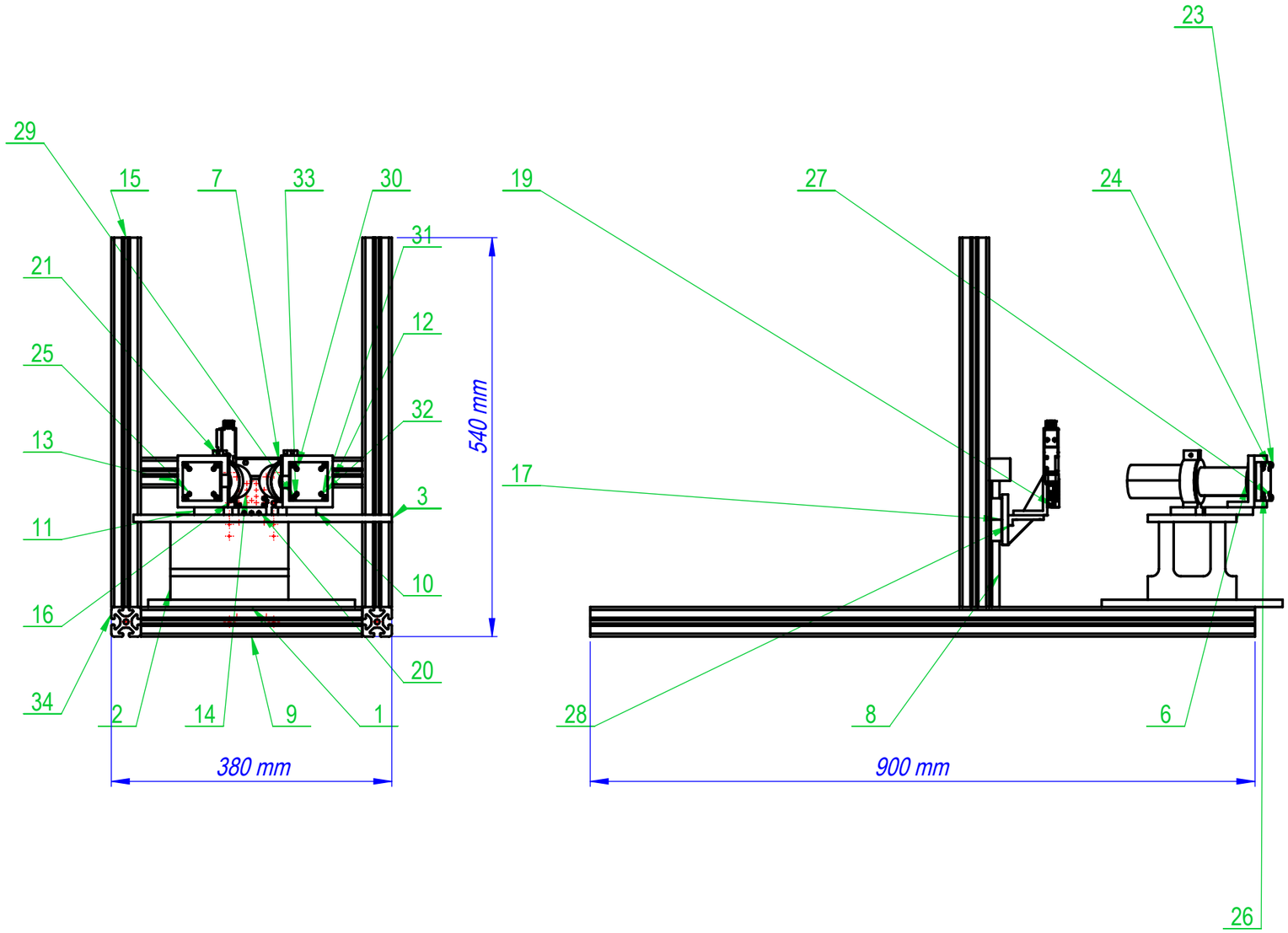


Ra 6.3

	File name	Additional information	Material	Scale
			Prusa PLA	1:1

Resp. department	Technical reference	Dokumento tipas	Document status
MIDF		Part drawing	Training

Legal owner	Created by	Title, Supplementary title	SVS-00.00.028			
ktu kauno technologijos universitetas 1922	Ahmed Elatroush	Straight motorized z bracket	Rev.	Date	Lang.	Sheet
	Approved by		A	5/11/2021	En	1/1
	Valdas Grigalunas					



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	table_base.stp		1
2	height_blank.stp		1
3	x_y_table.stp		1
4	camera_board.stp		2
5	cam_elevation.stp		2
6	cam_lens_holder.stp		2
7	fixture.stp		2
8	linear profile.stp		1
9	profile-40x40_horizontal_300mm.stp		4
10	adapter_2.stp		1
11	adapter_1.stp		1
12	new_cam_support.stp		1
13	new_cam_support_mirror.stp		1
14	microscopic lens 300x.stp		2
15	profile-40x40-vertical_500.stp		2
16	actuator_attachment_z.stp		1
17	carriage_block.stp		2
18	carriage_block_mirrored.stp		2
19	Part4.stp		1
20	motorized_stage_bracket.stp	STEP AP203	1
21	new holder uscope.stp		2
22	m2 screw.stp		4
23	m3_screw.stp		4
24	spacer_m2_5mm_tme_dremec.stp		4
25	cam_top_cov.stp		1
26	m2_nut.stp		4
27	m3_nut.stp		4
28	straight_motorized_z_bracket.stp		1
29	m3_screw_MIR.stp		4
30	spacer_m2_5mm_tme_dremec_MIR.stp		4
31	cam_top_cov_MIR.stp		1
32	m2_nut_MIR.stp		4
33	m3_nut_MIR.stp		4
34	profile-40x40_horizontal_900mm_column.stp		2

	Byla,laikmena	Papildoma informacija	Medžiaga	Mastelis M 1:10
Atsakinga žinyba	Vadovas	Dokumento tipas Assembly , BOM , Numbering	Dokumento statusas Training	
Savininkas KTU	Rengė Ahmed Elatroush	Antraštė Assembly of parts	Žymuo SVS-00.00.35	
	Tvirtino Valdas Grigalunas		Laida A / Data 05/20/2021 / Kalba lt. / Lapas 1/1	