

CAAP Quarterly Report

Date of Report: December 29, 2020

Prepared for: *U.S. DOT Pipeline and Hazardous Materials Safety Administration*

Contract Number: 693JK31950002CAAP

Project Title: AI-enabled Interactive Threats Detection using a Multi-camera Stereo Vision System

Prepared by: Arizona State University

Contact Information:

Dr. Yongming Liu (PI), Email: Yongming.Liu@asu.edu

Dr. Yang Yu (Co-PI), Email: yangyu18@asu.edu

For quarterly period ending: December 29, 2020

Business and Activity Section

(a) Contract Activity

Discussion about contract modifications or proposed modifications:

None

Discussion about materials purchased:

1. Intel® RealSense™ L515 LIDAR camera
2. Drilling set

(b) Status Update of Past Quarter Activities

In this quarter, the research team works on Task 1.1 and Task 2.1 to develop algorithms for estimating distance using stereo vision and unsupervised image segmentation using disparity map and color images. Experiments were conducted to verify the effectiveness and accuracy of the developed algorithms.

Student Training Activities

- Sampri Neog (MS student) works on stereo vision algorithm development for pipeline imaging, distance estimation, and preliminary demonstration to test the effectiveness and accuracy of the developed algorithm. (Task 1)
- Rahul Rathnakumar (PhD student) works on developing an unsupervised algorithm for image segmentation as a method for preprocessing the image for pipeline defect detection. (Task 2)

(c) Cost Share Activity

All cost share requirements have been satisfied in the past quarter and detailed financial report will be submitted by ASU financial department.

(d) Detailed Description of Work Performed

1. Background and Objectives in Q5 (2020)

Pipeline anomalies such as fatigue cracks, stress corrosion cracking, corrosion pits, and seam weld defects are major threats to the integrity of pipeline systems. The detection and characterization of these pipeline anomalies are critical for the safe operation of pipeline infrastructure, which is the objective of this ongoing project. The objective of this project is to develop a vision-based inspection tool using stereo vision and AI-enabled computer vision algorithms to address the pipeline anomaly detection and characterization.

Stereo vision uses two or more cameras to extract three-dimensional (3D) information by estimating the relative depth of points observed in digital images. The principle of stereo vision is illustrated in Fig. 1. In Fig. 1(a), C1 and C2 represent the optical centers of two cameras; b is the baseline distance between two cameras; P is the object point; and P1 and P2 are the projection of point P in the image plane. Points C1, C2, and P form a plane known as the epipolar plane. Fig. 1(b) shows a top view of the epipolar plane where f is the focal length. Based on similar triangles, we have:

$$\frac{z}{f} = \frac{x}{x_l} \quad \frac{z}{f} = \frac{x-b}{x_r} \quad \frac{z}{f} = \frac{y}{y_l} = \frac{y}{y_r} \quad (1)$$

where (x,y,z) is the global coordinate of the object point P, and (x_l, y_l, z_l) and (x_r, y_r, z_r) is the coordinate of the projection of point P in the left and right image planes, respectively.

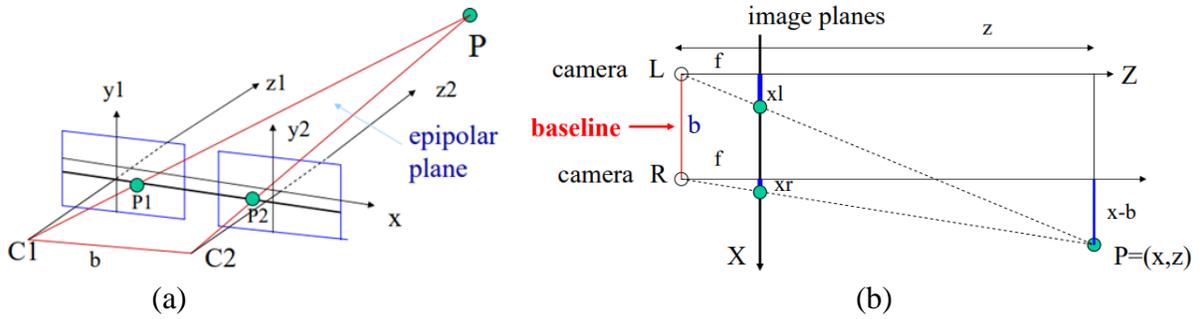


Figure 1: Principle of stereo vision: (a) epipolar plane; (b) triangulation

Based on the relationships given in Eq. (1), the global coordinate of point P can be calculated as:

$$z = \frac{fb}{(x_l - x_r)} = \frac{fb}{d} \quad x = \frac{b}{d} x_l \quad y = \frac{b}{d} y_l \quad (2)$$

where the difference $d = (x_l - x_r)$ is known as the disparity. Using Eq. (2), we can determine the depth of any scene point and thus construct a depth map of the observed scene. This method of determining depth from disparity is called triangulation. In practice, triangulation will be used to find the 3D locations of critical points on pipeline defects, from which we can estimate the distances between critical points to accurately characterize pipeline defects.

AI-enabled methods for threat detection can serve a key role in risk assessment of structures. Multi-modal analysis of a scene gives us multiple sources of information from which we can determine pertinent risks and threats. Fusing these sources could potentially give us enhanced assessments of system condition. In this report, we discuss a method to fuse multi-modal information sources for scene segmentation. Data obtained from multiple sensors can be processed, combined and manipulated in innovative ways to provide better predictions. The most commonly used definition of data fusion was proposed by the Joint Directors of Laboratories (JDL) workshop: “A multi-level process dealing with the association, correlation, combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete and timely assessments of situations, threats and their significance.” This is relevant for our project as we use depth and color information jointly for making predictions on pipeline condition.

The objective of the research in this quarter is to:

- (1) Perform sensitivity analysis using the D435i and LIDAR camera to determine the accuracy of depth and dimensional measurement of the object of interest.
- (2) Analyze the impact of IR dot projector density on the wall by adding two D435i cameras in series.
- (3) Propose a new representation for the depth information stream.
- (4) Update the semantic segmentation architecture to incorporate more recent developments in the computer vision field and test whether this brings any new observations and improvements to the results.
- (5) Acquire new data to approximate pitting and cracking defects.
- (6) Use the acquired data to obtain segmentation results using the updated architecture and estimate defect dimensions and depth.

2. Task 1: Development of A Novel Multi-Camera Stereo Vision System for Pipeline Inline Inspection

2.1 Sensitivity Analysis – D435i

2.1.1 Analyzing the camera parameters for thickness estimation

In this task, the analysis of camera parameters is performed by estimating the thickness of the object from the depth map and plotting the error concerning the parameters. The parameters being analyzed are:

- Exposure ($\frac{W}{m^2} s$): the amount of light per unit area reaching the surface of an electronic image sensor.
- Gain (dB): Amplifies the entire image signal
- Laser Power (W): Power provided to the IR projector
- DS_SECOND_PEAK_THRESHOLD: Determines how different the second-highest matches can be from the first highest matches for stereo depth computation
- DS_NEIGHBOR_THRESHOLD: Determines the number of neighboring pixels to be considered in the left image which will be compared with the right image for depth computation
- Disparity Shift (Min-Z and Max-Z change in Pixels): Controls the modification of the Z-min and Z-max values for the camera to visualize.
- Resolution: The fineness of detail in an image measured in pixels per inch (ppi)

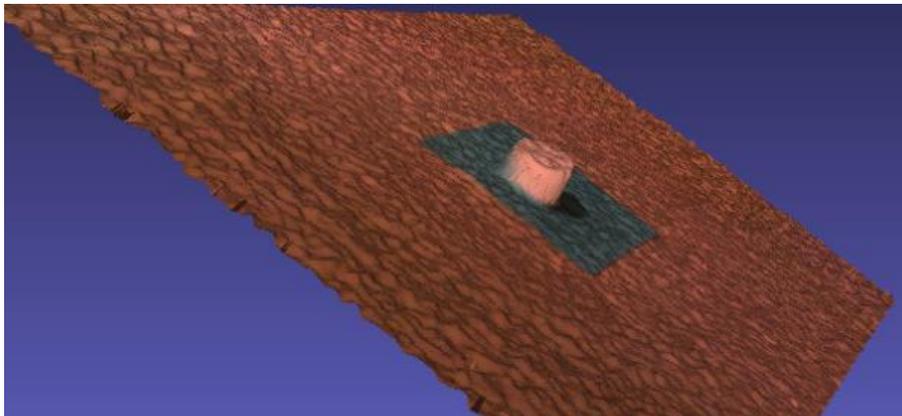


Figure 2: Object of thickness 18 mm thickness used for sensitivity analysis

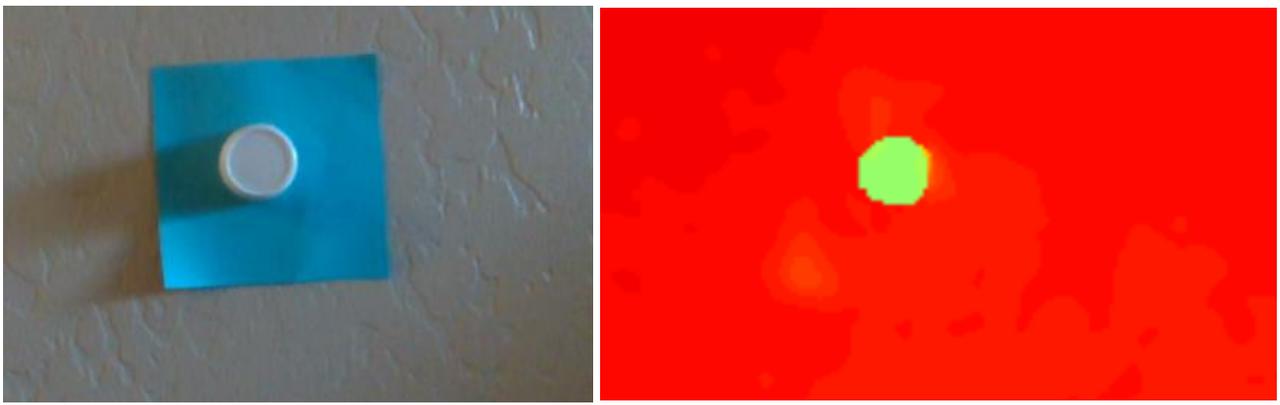


Figure 3: RGB and depth map corresponding to the 3D map

For performing this experiment, an object of 18-mm thickness and 11-mm radius is placed on a relatively flat wall in front of the camera as shown in Figure 2 and 3.

First, hole filling is performed on the raw depth map from the camera using the built-in pyrealsense hole-filling algorithm. Canny edge detection is performed on the image whose thresholding is determined using the pixel histogram. After canny edge detection, morphological dilation is performed on the edges to ensure continuity. Contour detection is then performed on the dilated image to find the boundary points. The boundary points are a discrete set of pixel coordinates, and a curve is generated through these points to create the best approximation. The procedure is shown in Figure 4.

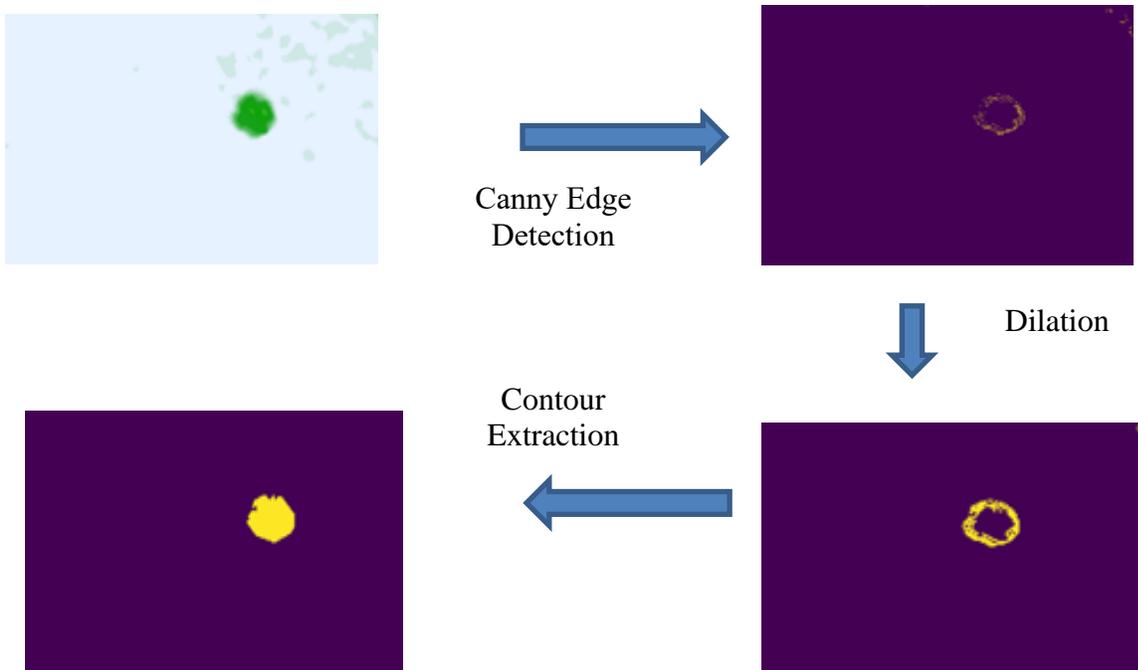


Figure 4: Contour extraction process for thickness estimation

After finding the contour and filling the inside part of the contour with a uniform color opposite from the background, the pixels inside the contour are separated with pixels in the background using the color difference. Then the distance from the camera to the object is obtained by calculating the distance of the camera from the pixels inside the contour and the distance from the camera to the background is obtained by calculating the distance from pixels in the background. Then the erroneous distance for each pixel is removed before calculating the average distance (for all the pixels) from the background and the object separately. Then the thickness of the object is estimated as:

$$\text{Thickness of object} = \text{Distance from the background} - \text{Distance from the object}$$

The thickness estimation is then performed for all camera parameters mentioned above:

I. Error in the thickness estimation with respect to change in Gain

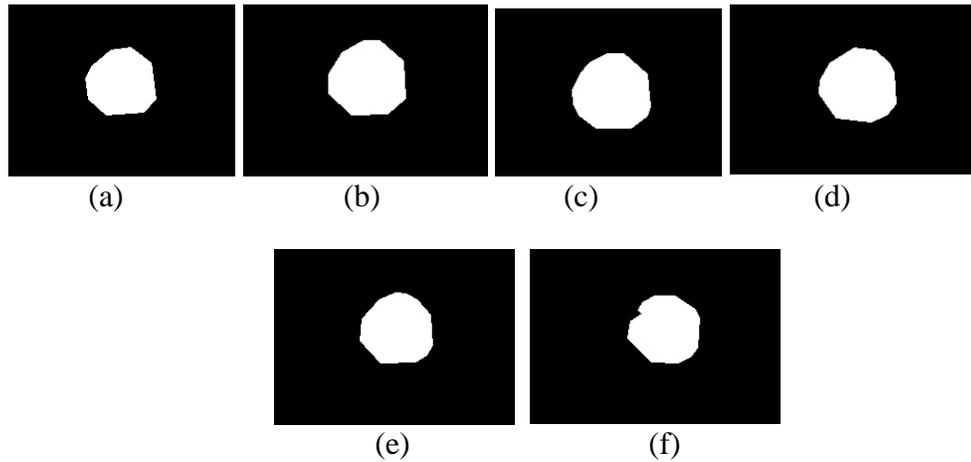


Figure 5: Contour formation for different Gain (dB): (a) 16; (b) 36; (c) 56; (d) 76; (e) 96; (f) 116

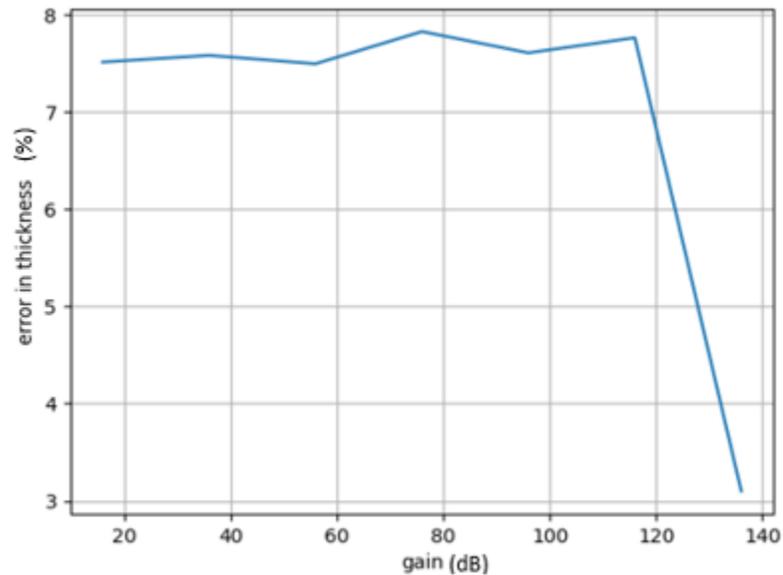
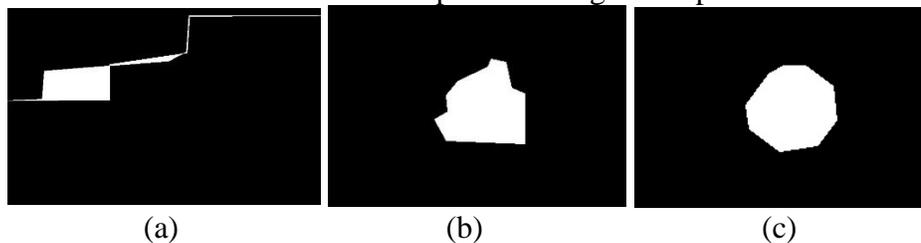


Figure 6: Plot of error in thickness estimation versus camera gain

Figure 5 shows the resulting object contours and figure 6 shows that we can get the depth information with reasonable accuracy in the Gain range of 16-140 dB with a resolution of 848x480.

II. Error in the thickness estimation with respect to change in Exposure



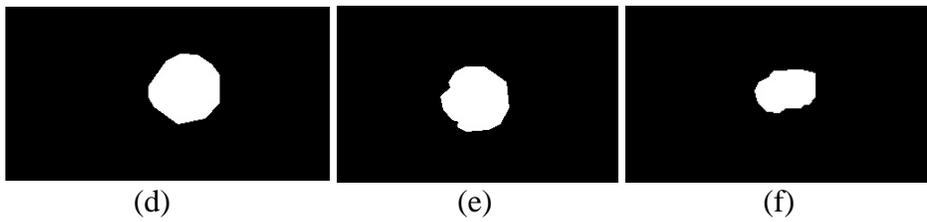


Figure 7: Contour formation for exposure: (a) 0 (b) 500 (c) 1000 (d) 7500 (e) 14500 (f) 16000

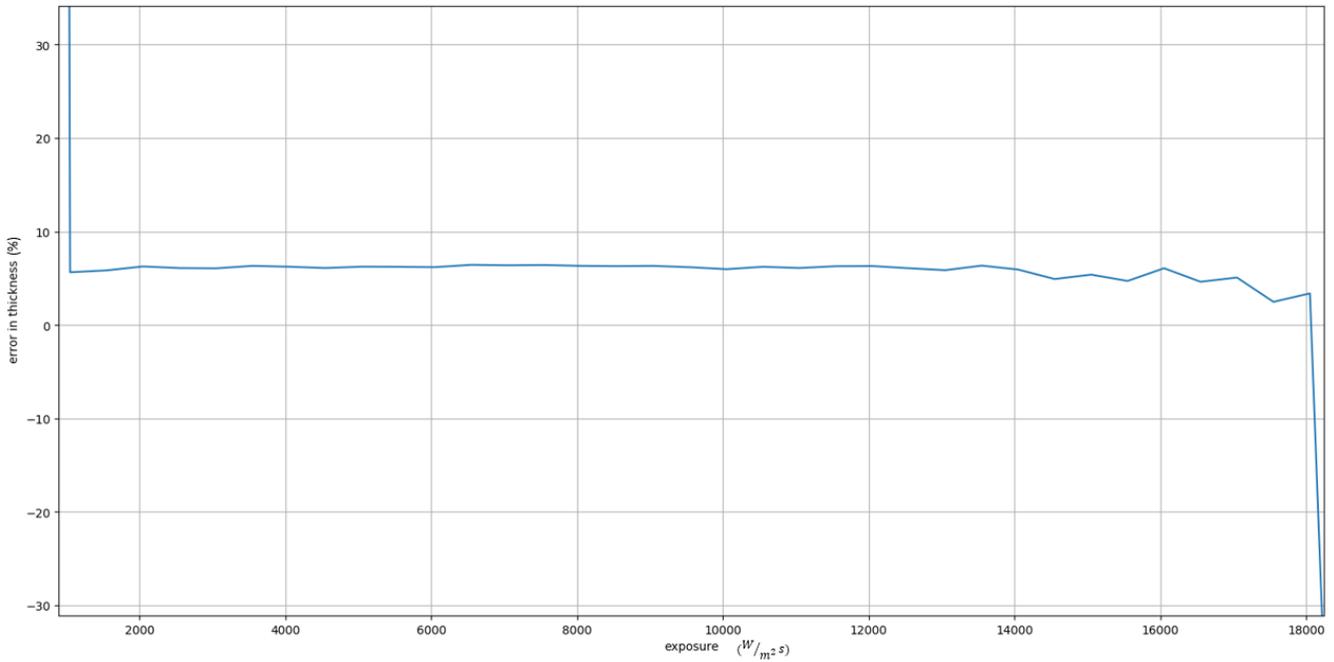


Figure 8: Plot of Exposure versus error in thickness estimation

As shown in figure 8, the thickness estimation is valid only in the range of 2000 to 16000 with a resolution of 848x480 as the plot shows that the error is low in this range. The contour and raw depth map below 2000 is as shown in figure 9. This shows that exposure below 2000 may not be suitable for thickness estimation. Exposure depends on lighting conditions, and these might vary with different lighting conditions, and auto-exposure algorithms are already present in the Realsense camera system to calibrate this according to the scene brightness.

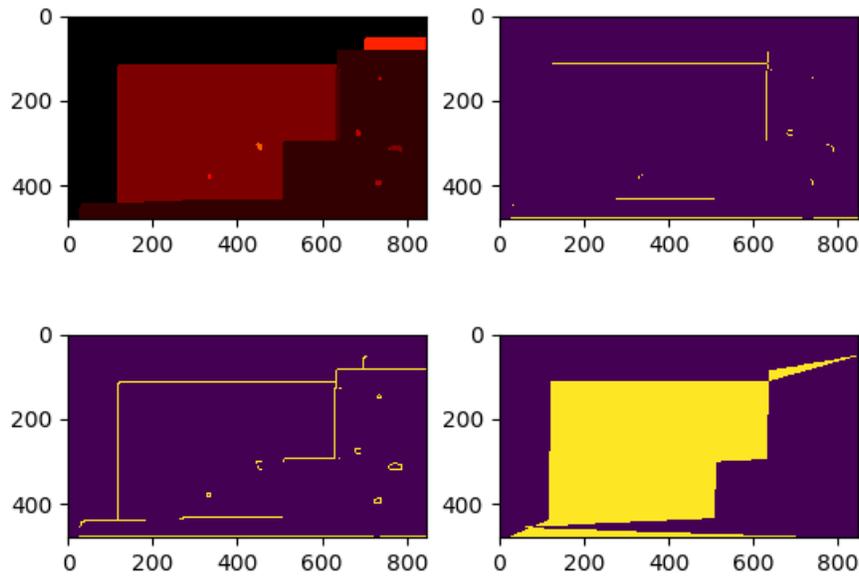


Figure 9: Contours and depth map for exposure below 2000

The contour formation for exposure between 2000-16000 is shown in figure 10.

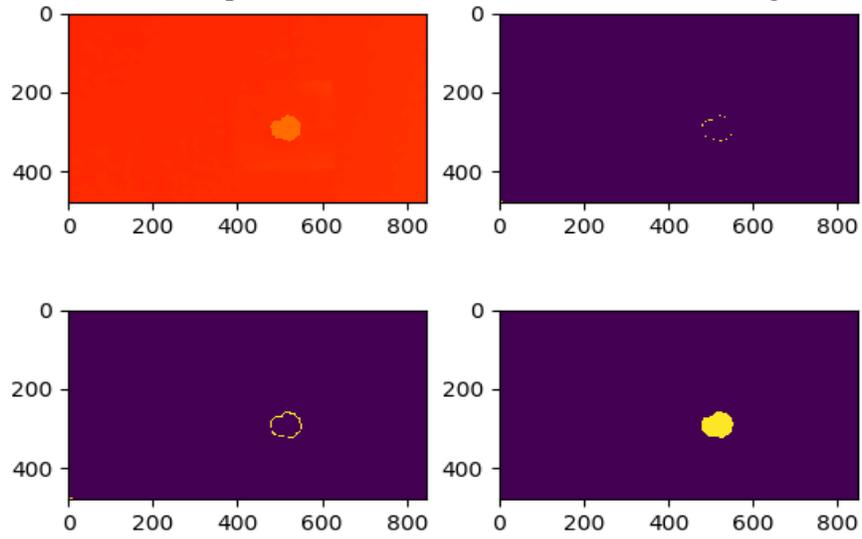


Figure 10: Contours and depth map for exposure between 2000-16000

The contour formation for exposure beyond 16000 is plotted in figure 11. This shows that exposure beyond 16000 may not be suitable for thickness estimation, especially in bright areas.

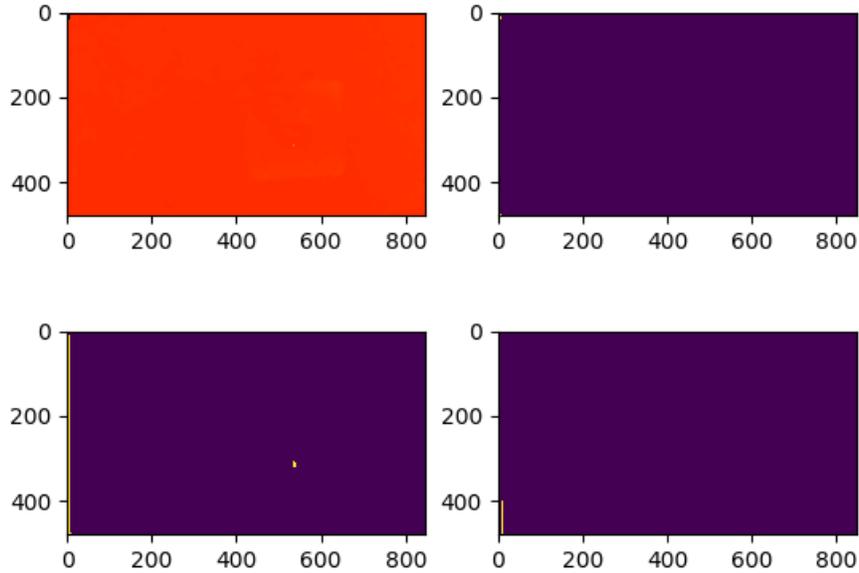
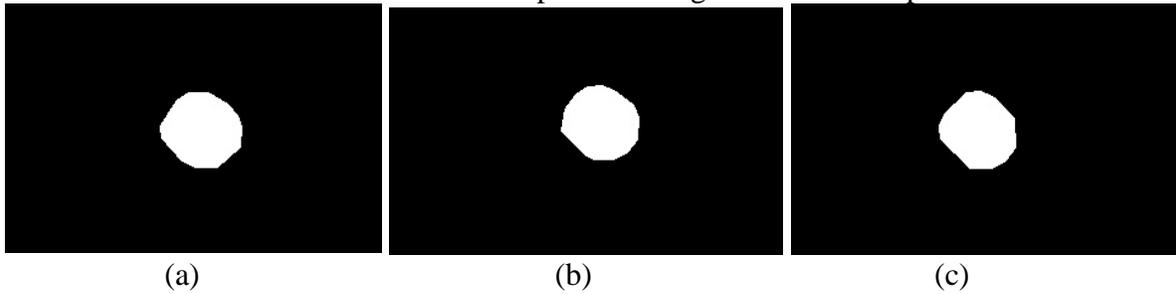


Figure 11: Contours and depth map for exposure beyond 16000

III. Error in the thickness estimation with respect to change in DS second peak threshold



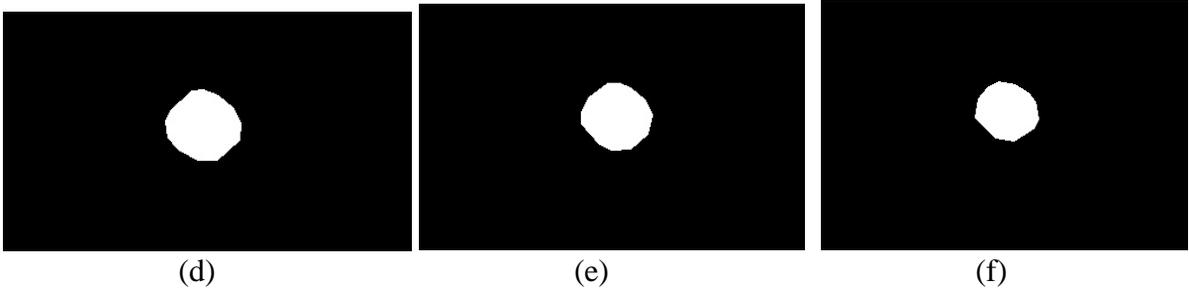


Figure 12: Contour formation for DS second peak threshold: (a) 0 (b) 100 (c) 200 (d) 300 (e) 400 (f) 500

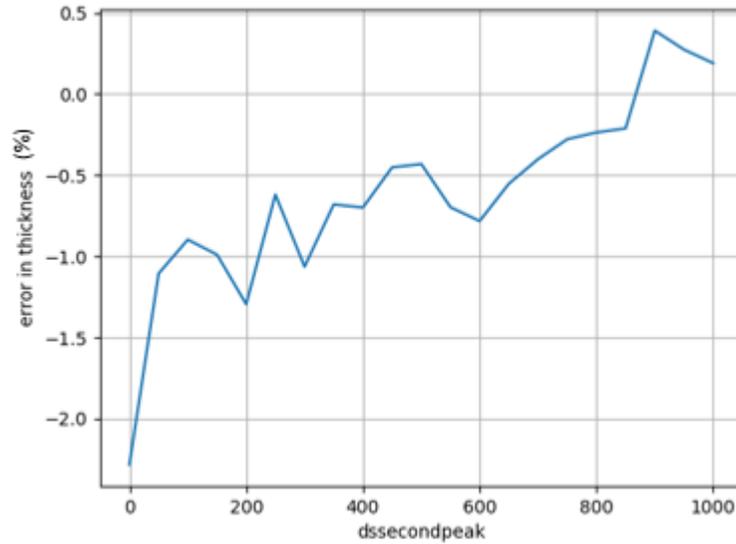
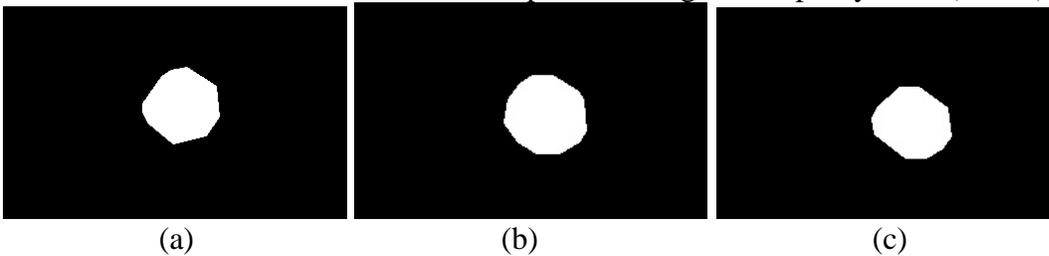


Figure 13: Plot of DS second peak threshold versus error in thickness estimation

As shown in figure 13, the thickness estimation is accurate in the range from 0 to 1000 under the resolution of 848x480. This is also confirmed by the visualizations of depth map in figure 12. Unlike exposure and gain, which are heavily influenced by factors such as brightness of the scene, the thickness error is very low through the entire range of the second peak threshold. This is because the second peak threshold only influences the fill rate of the depth map, that is, the number of holes, which affects area computation more than thickness. Another observation is that the minimum error in figure 13 does not correspond to that observed in figure 15. This is because there are variations in external lighting conditions due to these experiments being conducted in the home environment with loose lighting controls in contrast to the controls in the lab. In the future, an algorithm will be proposed to automatically recommend parameters based on certain quality metrics using unsupervised learning.

IV. Error in the thickness estimation with respect to change in Disparity Shift (Pixels)



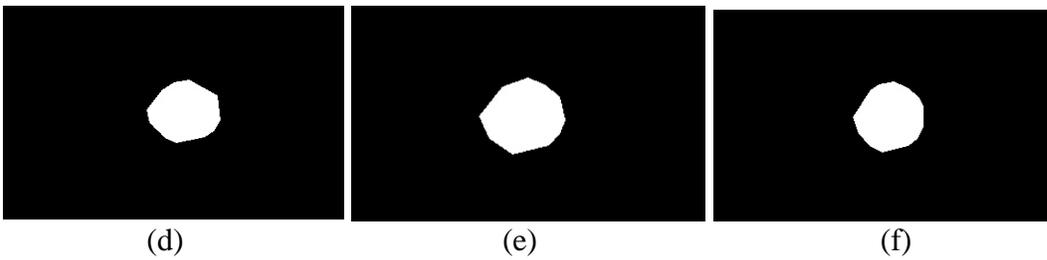


Figure 14: Contour formation for Disparity shift: (a) 0 (b)50 (c)100 (d)150 (e) 200 (f)250

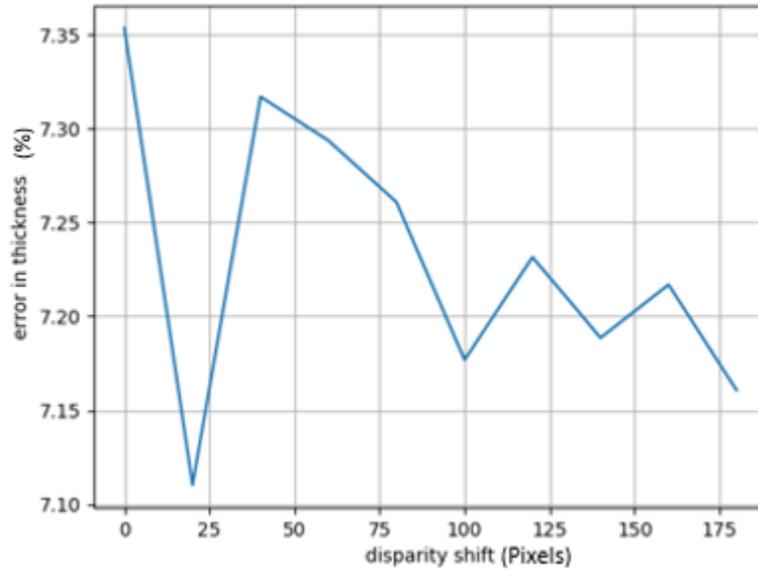


Figure 15: Plot of Disparity Shift versus error in thickness estimation

As shown in figures 14 and 15, the thickness estimation is good in the entire range of 0 to 150 pixels under the resolution of 848x480. However, this setting may cause problems if the camera is located far from the pipe surface, in which case a setting of zero is more appropriate. The closer the camera gets to the surface, the higher the value of the disparity shift needs to be.

V. Error in the thickness estimation with respect to change in DS neighbor threshold

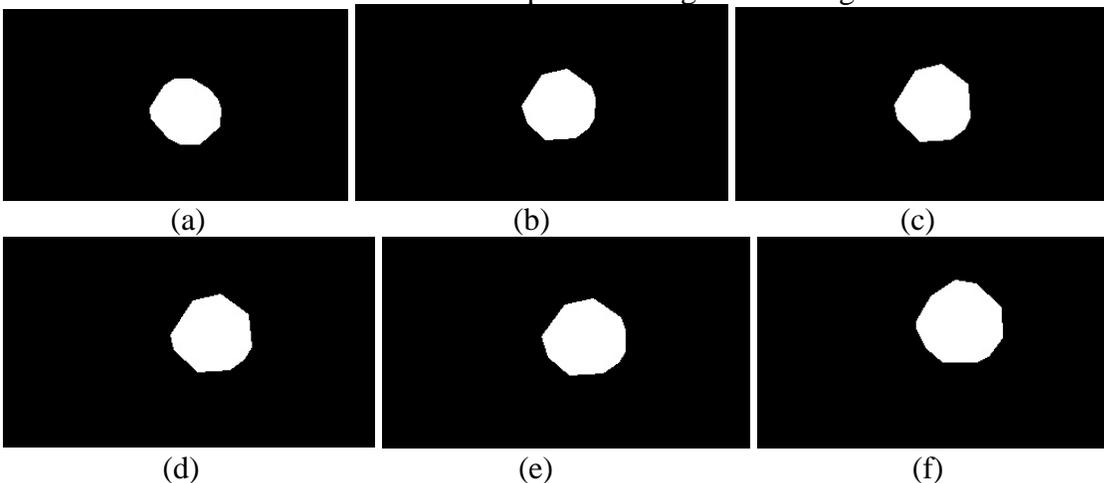


Figure 16: Contour formation for DS neighbor threshold: (a) 0 (b) 50 (c) 100 (d) 150 (e) 200 (f) 250

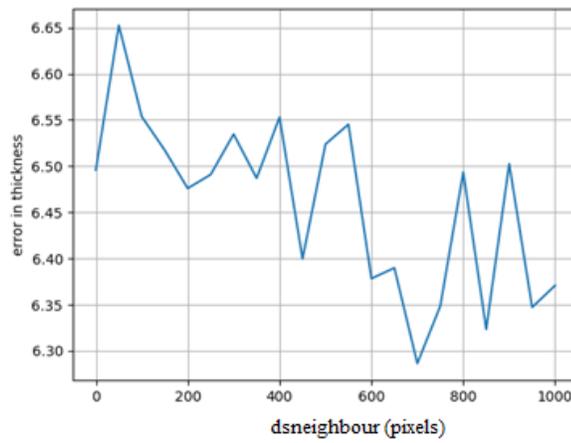


Figure 17: Plot of DS neighbor threshold versus error in thickness

As shown in figures 16 and 17, the optimal range of DS neighbor threshold for thickness estimation is from 0 to 1000 under the resolution of 848x480.

VI. Error in the thickness estimation with respect to change in laser power

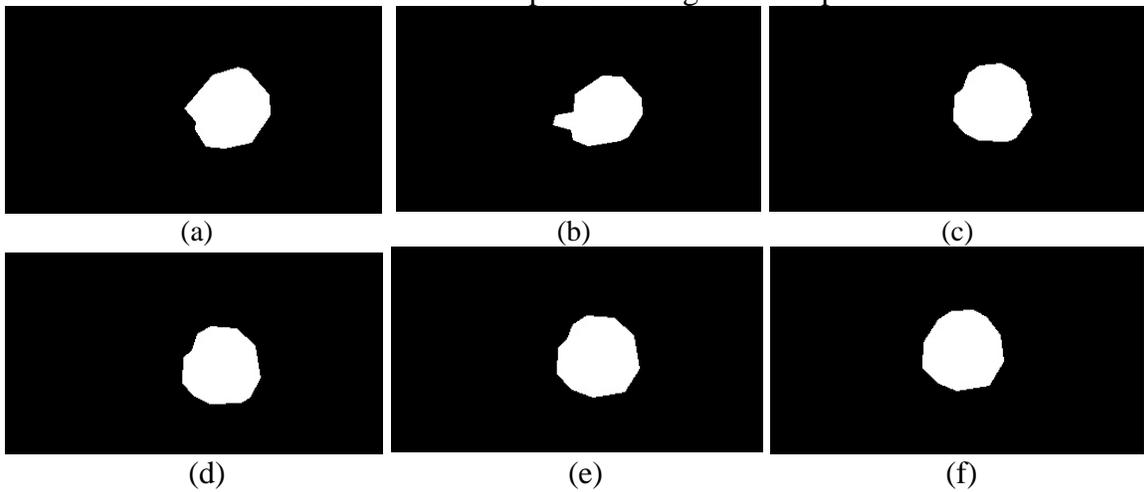


Figure 18: Contour formation for laser power (W): (a) 0 (b) 50 (c) 100 (d) 150 (e) 200 (f) 250

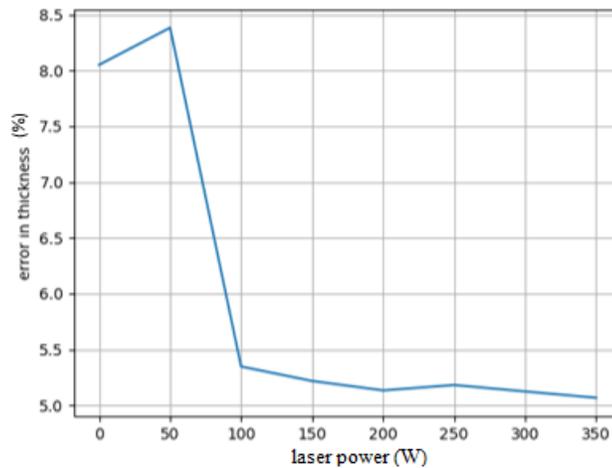


Figure 19: Plot of Laser versus error in thickness

As shown in figure 18 and 19, the change in laser power does not seem to have a major impact on the depth map quality and thickness estimation accuracy.

VII. Error in the thickness estimation with respect to change in resolution

Table 1: Error in thickness estimation versus Resolution

Resolution	Error in thickness	Pixel length in mm
640x360	-11%	0.5963
848x480	-7.5%	0.4472
1280x720	-8.02%	0.3003

Table 1 shows that the resolution has an insignificant effect on the thickness estimation of the object used for experiment but it affects the pixel length. It should be noted that the thickness of the object is 18 mm, which is above the minimum depth detectable by the camera. Therefore, changing the resolution does not have a major effect on the thickness. In addition, in the resolution of 1280x720, the disparity shift needs to be above 20 to get proper information of the object depth.

VIII. Error in the thickness estimation of an object of small thickness

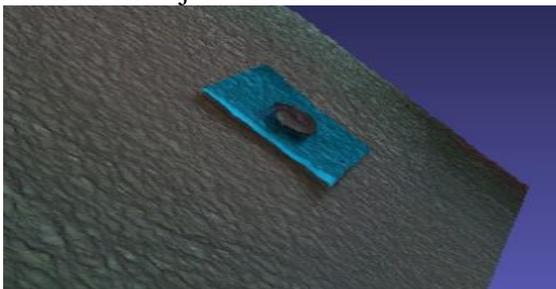


Figure 20: 3-D visualization of a 5 mm thickness coin

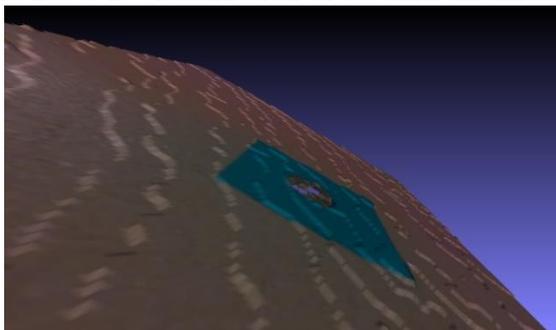


Figure 21: 3-D visualization of a coin of a thickness of 1.75 mm

Figure 20 shows the 3D reconstruction of the scene for the object of 5-mm thickness and Figure 21 shows the 3D reconstruction of the scene for the object of 1.75-mm thickness. The surface reconstruction is obtained by using the point cloud generated by the depth frame aligned with the RGB camera frame. Then, the RGB texture is applied to the point cloud instead of the depth texture. The 5-mm thick object was created by stacking multiple coins on top of each other for this experiment. For thickness estimation of an object of 5-mm thickness and 24-mm diameter, the depth visualization is adjusted according to the distance of the object from the camera. Then contour analysis was performed on the raw depth map as shown in Figure 22. The error was found to be around 22% under the resolution of 848x480. The error in the thickness of an object of 24-mm diameter and 1.74-mm thickness was found to be 18.85%. The results are shown in Figure 23.

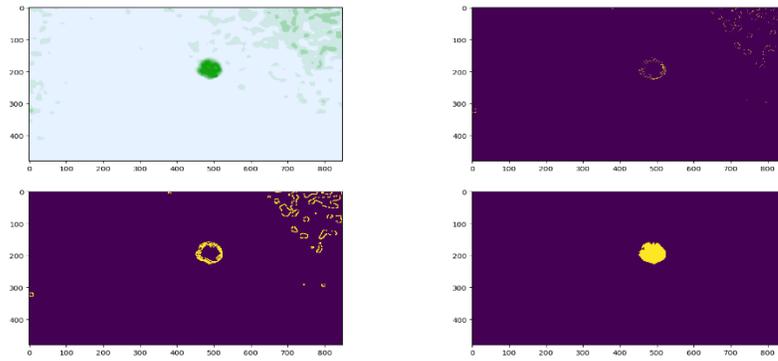


Figure 22: Contour extraction from the depth map for the object of 5 mm thickness

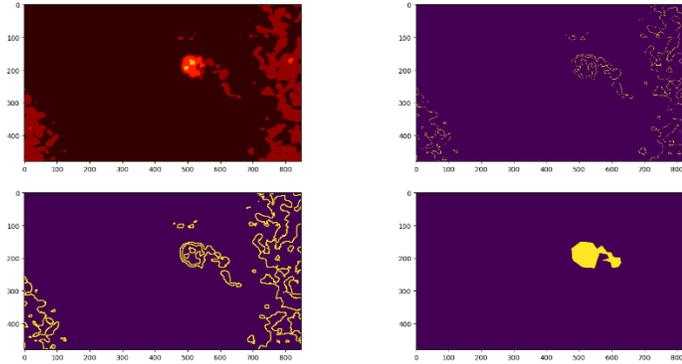


Figure 23: Contour extraction from the depth map for the object of 1.74 mm thickness

IX. Error in the thickness estimation of an object of a small area

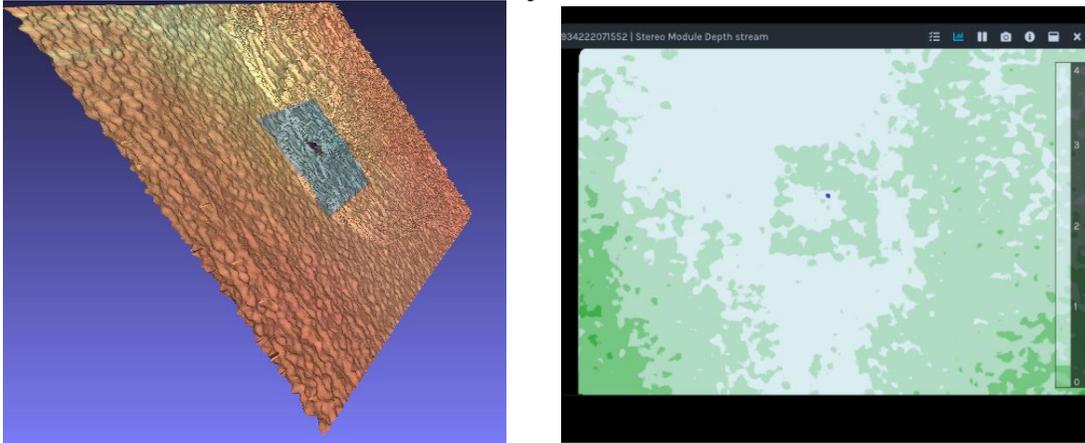


Figure 24: 3-D RGB and depth map for the object of 8-mm diameter

An object of 15-mm thickness and 8-mm diameter was difficult to process with our contour algorithm, so we chose to get the depth information from the Intel RealSense viewer which is used to visualize the raw depth map directly from the Camera as shown in Figure 24. In this case, we are able to accurately estimate the thickness. It should be mentioned that the manual measurement of the depth using the RealSense viewer meant that the number of significant figures that the depth could be computed to was at the millimeter scale. Secondly, it is important to note that the shape of the object was not captured, and only one sample that indicated a deformation in the surface corresponding to the location of the object was taken.

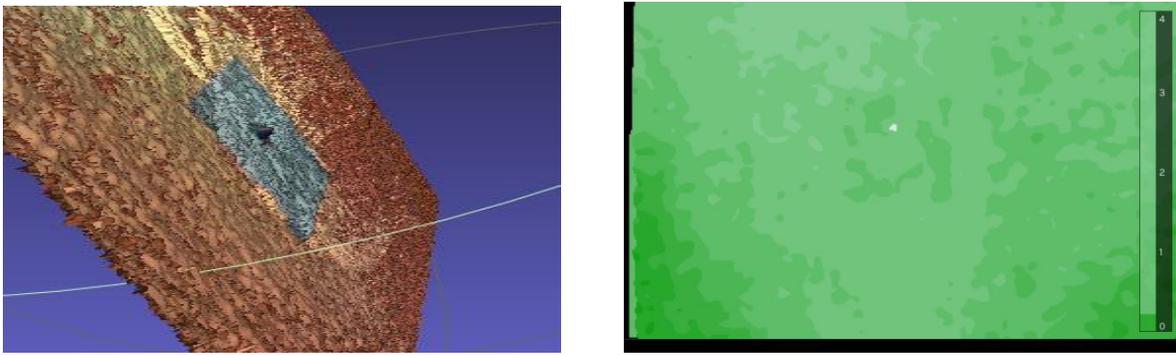


Figure 25: 3-D RGB and depth map for the object of 7.5-mm diameter

An object of 21-mm thickness and 7.5-mm diameter was difficult to process with our contour algorithm, so we chose to get the depth information from the Intel RealSense viewer which is used to visualize the raw depth map directly from the Camera as shown in Figure 25. The error was found to be -9.1%.

A summary of the sensitivity analysis conducted is given below:

Parameters	Optimal parameter range	Note
Gain	16-55 dB	Affects mostly the shape of the object beyond 55 dB
Laser power	0-350	
Exposure	2000-18000	Highly dependent on lightning conditions
Disparity Shift	0-125 pixels	Affects mostly shape of the objects beyond 125 pixels
DS neighbor Threshold	0-1000	
DS second peak Threshold	0-600	Affects mostly shape of the objects beyond 600

2.2 Testing of a Lidar Camera for defect detection

2.2.1. Background

Intel RealSense Lidar Camera L515 is a Lidar Camera good for depth sensing in a small form factor. It has an IR Laser, MEMS, IR sensor, RGB camera, display processing ASIC. The MEMS is used to project the IR laser beam over the entire Field-Of-View (FOV). The reflected IR beam is sensed by the IR photodiode sensor which is processed by the display processing ASIC to produce depth points thus giving an accurate distance estimate of the objects in the scene. A point cloud is generated from the combination of depth points. It has an RGB camera to capture images in RGB format and IMU to track its motion and orientation. The architecture is shown in Figure 26.

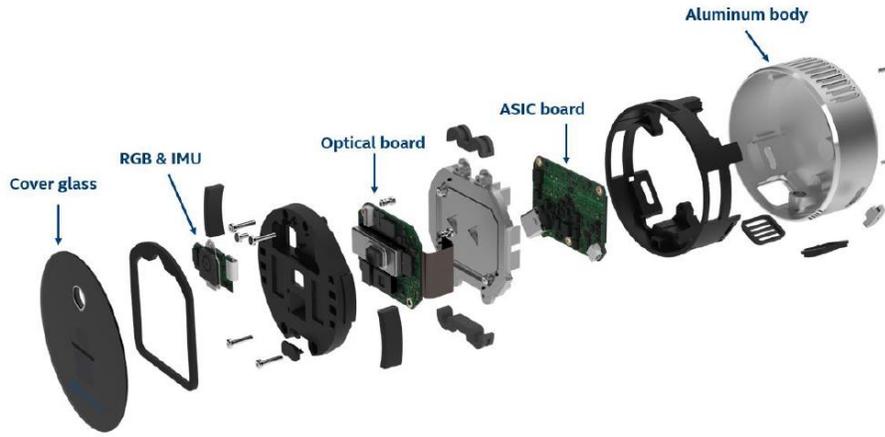


Figure 26: Physical architecture of L515[1]

The main advantage of this camera is that it can capture depth points in a short exposure time of less than 100ns which helps in capturing objects in fast motion without blur. It can capture over 23 million depth points per second with its ASIC. It also consumes less than 3.5 W. Physically, it is very small with a diameter of only 61 mm and 26 mm in height. It weighs only 100g. Thus, it is easy to use for depth sensing and integrating with other modules such as a phone or tablet or a robot.

The camera performs best at 250mm to 9000mm distance (0.25 to 9 m). It can measure 30 FPS Depth at a resolution of 1024x768 and 30 FPS color at a resolution of 1920x1080. It has a FOV of 70°x55°.

2.2.2 Effect of distance on Thickness estimation

This experiment was carried out by placing the Lidar Camera at some distance from the circular object of 18-mm thickness which is attached to a flat wall. Then we calculated the average error in thickness estimation by using the same contour estimation method used before for 10 frames to remove the temporal noise effect from the calculations. All the camera parameters are kept unchanged.

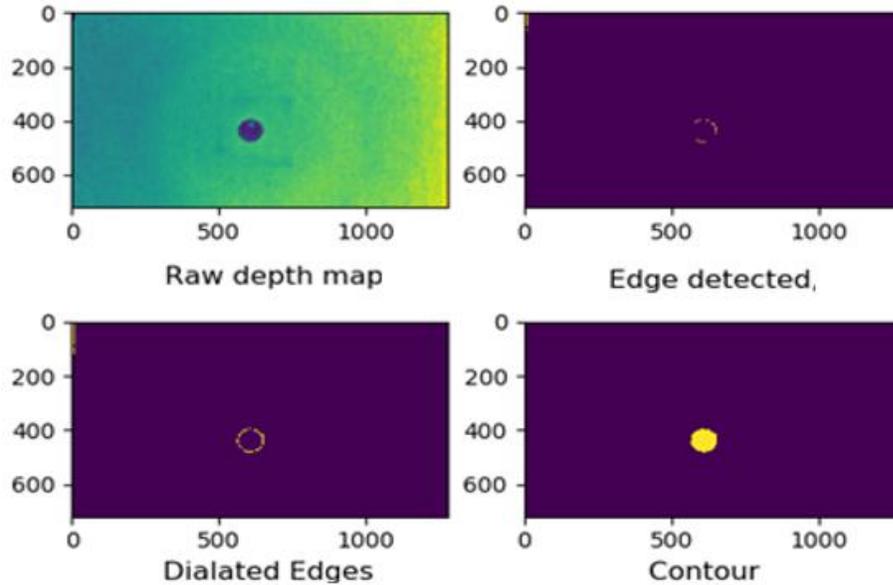


Figure 27: Thickness estimation for the object of thickness 18 mm using Lidar Camera

As shown in Figure 27, the shape of the object can be clearly visualized. But the average error of thickness estimation is given as follows:

Table 2: Error in thickness versus distance from the camera

Distance from the camera (mm)	Average error for 10 frames
311	-10%
297	-20%
249.2	-25%
198	-26.3%
155	-38.91%

Table 2 shows that the Lidar camera gives better thickness estimation when the object is placed at a larger distance from the camera. This analysis was performed at a resolution of 1024x768.

2.2.3. Thickness estimation of a small object

The above analysis was repeated for an object of 50-mm thickness and 24-mm diameter as shown in Figure 28.

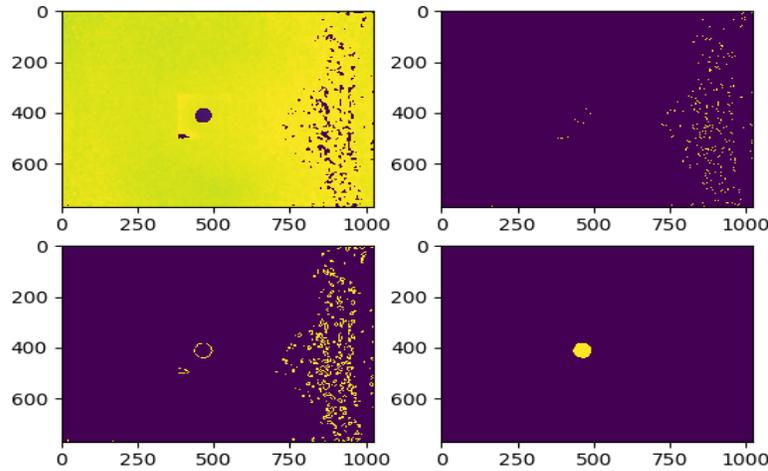


Figure 28: Thickness estimation for the object of thickness 50 mm using Lidar Camera

The actual thickness of the object (purple colored object) being visible in the above image is 50 mm. As we can see from the above image that the shape of the object is a circle which is clearly visible in the depth map. The error of thickness estimation is around 20% at 322 mm camera distance from the background.

2.3 A new representation for the depth stream

Depth maps provide the distance information from the camera to the surface of interest. However, additional geometric cues are embedded implicitly in this representation. [2] uses the depth information to compute additional data for the neural network, which explicitly provides geometry cues. These geometry cues include the Height above ground (H), Horizontal Disparity (H) and Angle with gravity (A). This representation is known as the HHA representation and has been used in [3], [4]. For our problem, we propose an alternative representational scheme, the DNC (Depth-Normal-Curvature) representation. This is a 6-channel representation compressed into 3 channels using 1x1 convolutions, as proposed below in figure 29.

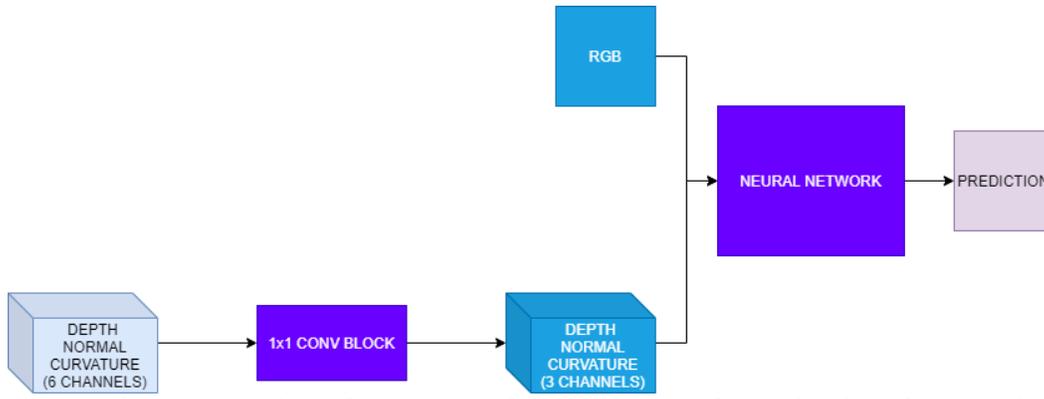


Figure 29: Proposed DNC representation and its transformation into 3 channels

The surface normal is a 3-channel image, one for each component of the normal vector. The curvature consists of the mean and gaussian curvatures, which can be computed from the principal curvatures of a point in a surface.

Curvatures provide a measure of how “curved” a surface is and can therefore be very useful to characterize defects in pipelines. The depth information obtained from the stereo-camera contains the curvature information implicitly. The pipe curvature is modeled by the depth map, and the goal of the investigation is to find out whether transforming this curvature information such that the defect is mapped onto an equivalent flat plate would improve defect localization performance. MSU used a least-squares fit of a cylinder to accomplish this task. But this approach is sensitive to the location of the camera and can introduce errors in the downstream. Instead, if the depth map information is augmented with geometric information, it may help detect weaker defect features.

Computation of surface normals and curvatures:

The depth map consists of data in the form of $Z = h(X, Y)$ where (x, y) belongs to an open set $U \subset R^2$. The surface form of the map is:

$$\eta(X, Y) = (X, Y, h(X, Y)), (X, Y) \in U$$

The normals are computed on this surface using the first derivative information as follows:

The gaussian and mean curvatures K and H of the surface are computed using the fundamental coefficients of a surface. These are obtained by using the first and second derivative information of the surface as follows:

$$D_i \eta = \left[\frac{\partial X_j}{\partial x_i}, \frac{\partial Y}{\partial x_i}, \frac{\partial Z}{\partial x_i} \right]$$

$$D_{ii} \eta = \left[\frac{\partial^2 X_j}{\partial x_i^2}, \frac{\partial^2 Y}{\partial x_i^2}, \frac{\partial^2 Z}{\partial x_i^2} \right]$$

$$D_{ij} \eta = \left[\frac{\partial^2 X_j}{\partial x_i \partial x_j}, \frac{\partial^2 Y}{\partial x_i \partial x_j}, \frac{\partial^2 Z}{\partial x_i \partial x_j} \right]$$

$$n(X, Y, Z) = \frac{D_x \eta \times D_y \eta}{|D_x \eta \times D_y \eta|}$$

$$E = D_x \eta \cdot D_x \eta; F = D_x \eta \cdot D_y \eta; G = D_y \eta \cdot D_y \eta$$

$$L = D_{xx} \eta \cdot n(X, Y, Z); M = D_{xy} \eta \cdot n(X, Y, Z); N = D_{yy} \eta \cdot n(X, Y, Z)$$

$$K = \frac{(LN - M^2)}{EG - F^2}$$

$$H = \frac{LG + NE - 2FM}{2(EG - F^2)}$$

Here E, F, G, L, M, N are the fundamental coefficients (first and second kind) of the surface based on a differential geometric formulation. $n(X, Y, Z)$ is the point cloud normal vector field. H and K are the mean and gaussian curvature scalar fields respectively. η is the point cloud.

Demonstration:

Example 1: Smooth Flat Plate

A smooth flat plate was created with a (1000x1000) grid with $z = 1$ as shown in figure 30. The normals were calculated to be (0,0,1) across the entire grid, which is accurate. The curvatures, both mean and gaussian, are zero as well, which indicates that the computation is correct for the flat plate.

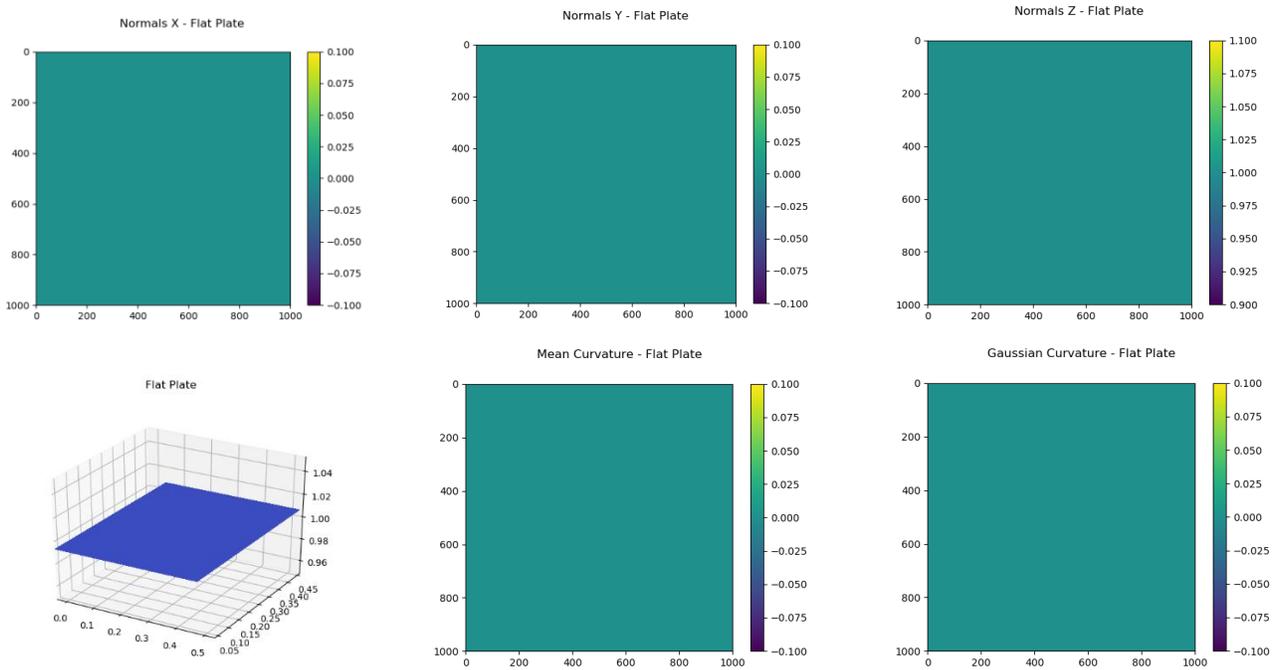
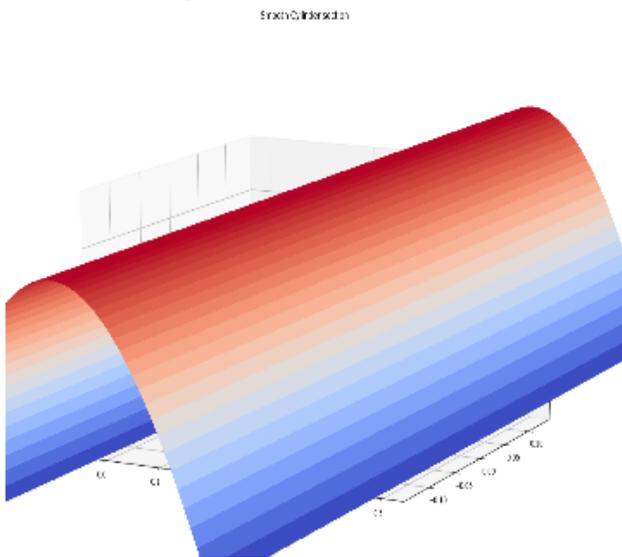


Figure 30: Normals and Curvatures for an idealized flat plate

Example 2: Smooth cylinder section



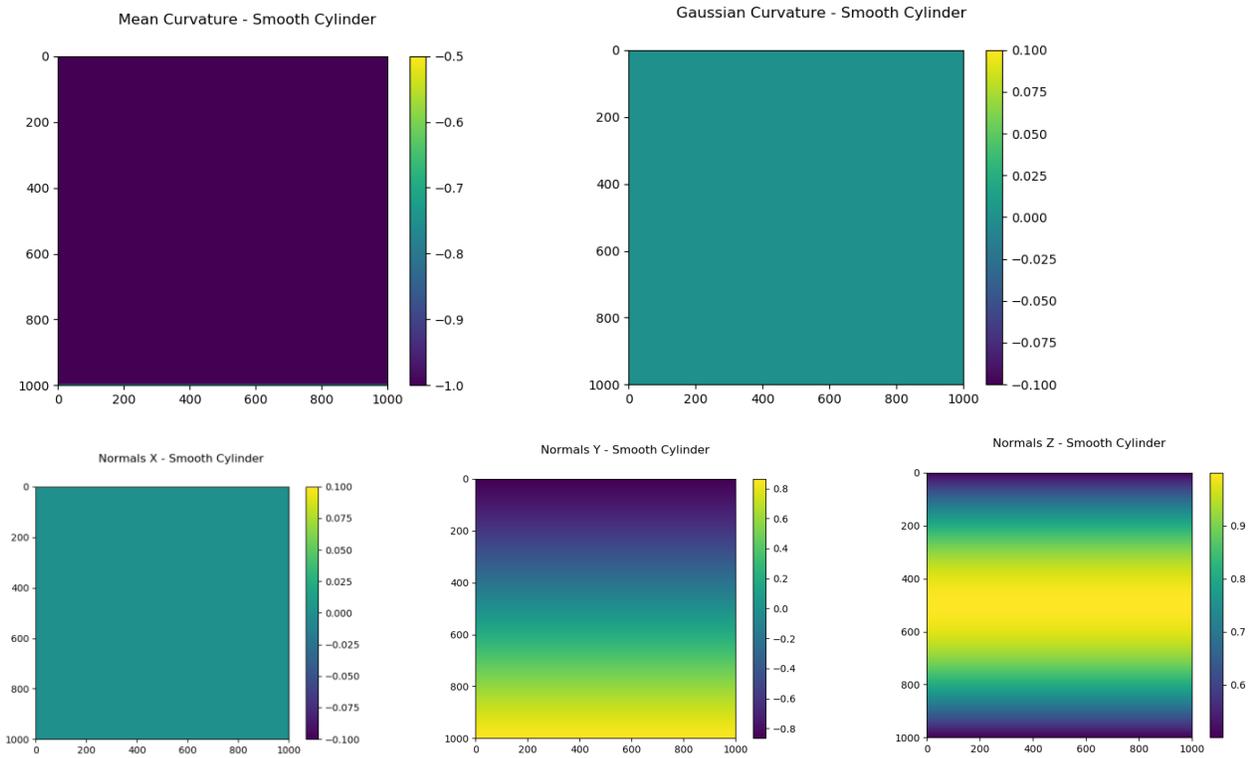


Figure 31: Normals and curvatures for a smooth cylinder section

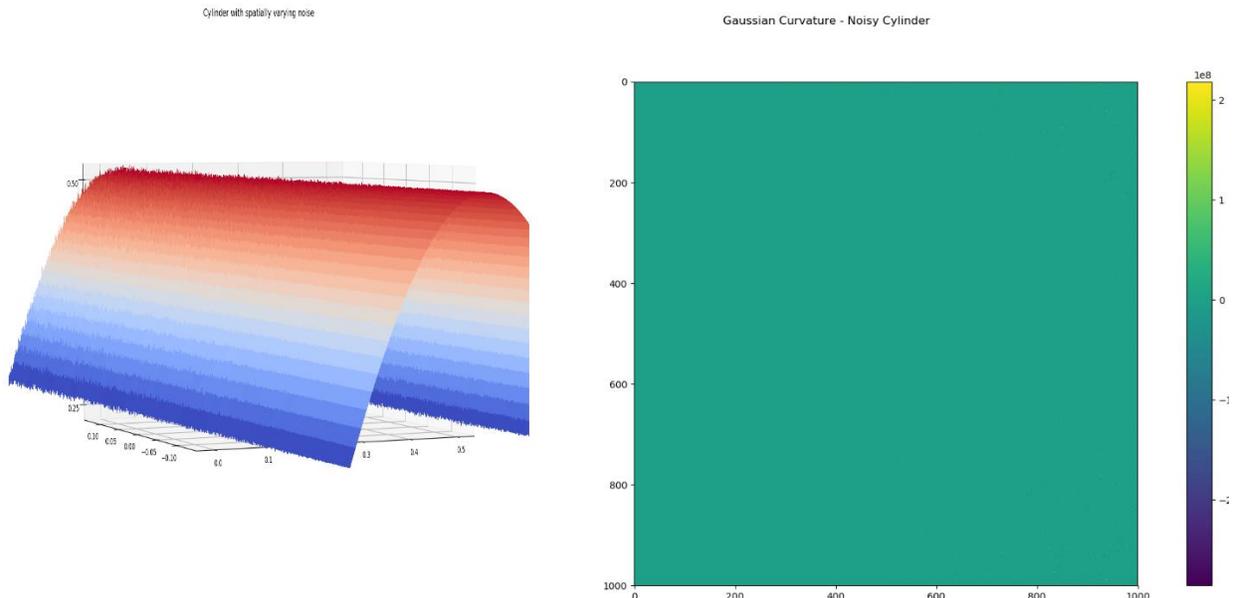
A cylinder section was created using a (1000x1000) grid, with a 120-degree field of view as shown in figure 31. The mean curvature was calculated to be -1, and the gaussian curvature was calculated to be 0. The X-direction normal was found to be zero, with non-zero values for the Y and Z component.

Example 3: Rough cylinder section

A rough cylinder section was simulated by construction of a smooth cylinder and superimposing a spatially varying noise function as follows:

$$Z = Z + N(0, |N(0,0.025) * \sin(X)|)$$

This produces a noisy cylinder with variable variance that is also a function of X, as shown in figure 32. The results show that there are spikes in curvature where there is noise. The X-normal component demonstrates that the noise increases as the X coordinate increases, with the value approaching 0 (similar to the smooth case) as X goes to 0.



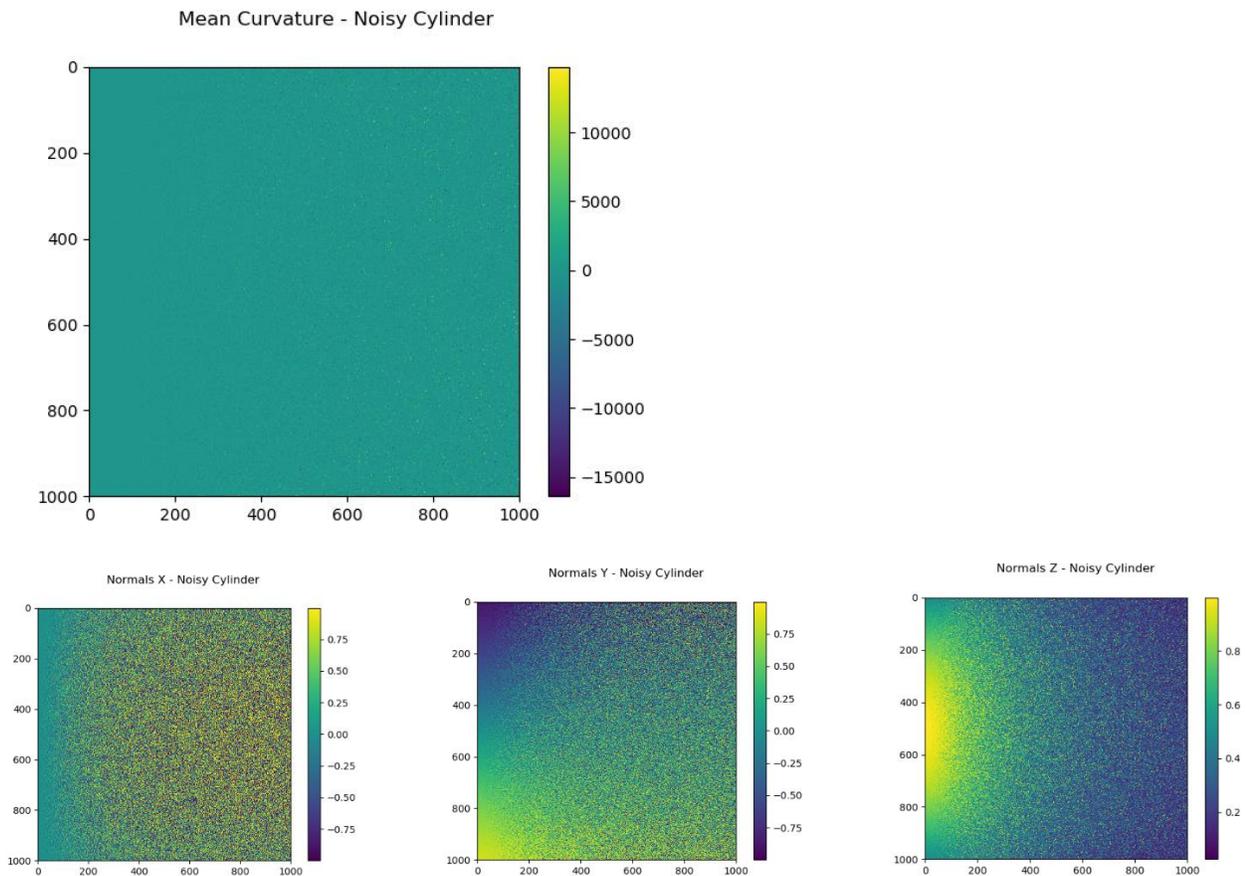


Figure 32: Normals and curvatures for a noisy cylinder section

Future work will consider using real world data from the D435 camera to obtain these representations and validate their usefulness.

2.4 Detection and 3D reconstruction of pitting defects

Until the previous quarter, the reporting only considered protrusion defects on two types of surfaces: Flat and curved. In this quarter, hole defects were explored by drilling a small hole of width 4mm and depth 15mm on a wall. This was a test to simulate pitting. The camera was set to a resolution of (1280x720) and a disparity shift of 120, to ensure the best quality image at this resolution and distance from the surface. Another change made to the settings was to set the depth units to 1e-5 from 1e-3 (m), to better resolve the small indentation in the wall relative to the average depth of the background. The defect was observed and the measured width was 3 mm and the detected depth was 2 mm. The reconstruction of the defect along with the depth map is shown in figure 33. As the defect width is very small, and the matching algorithm cannot determine with confidence the depth of these localized defect regions, which makes accurate depth detection very difficult in this case. Increasing the size of the defect improves accuracy as shown by the sensitivity analysis.

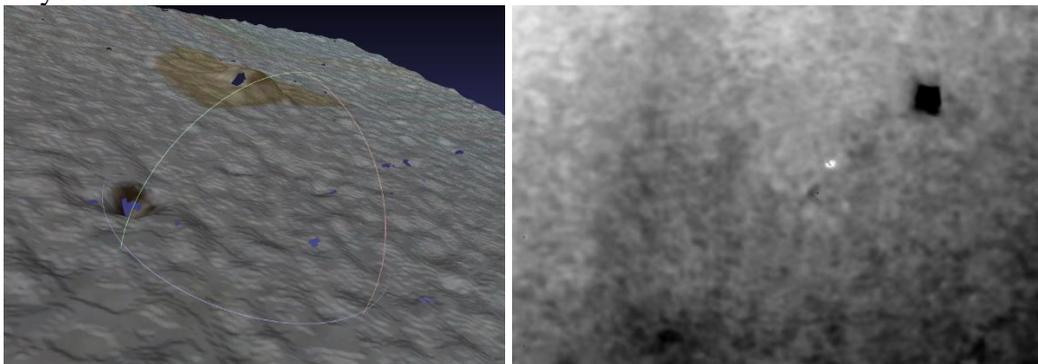


Figure 33: (Left) 3D reconstruction of a hole and a protrusion defect (Right) Depth map of the scene

Additionally, more images were acquired from the D435i system on the pipe sample by drilling into it and causing pitting-like formations on the pipe. These localized defect regions were detected by the camera and the 3D reconstruction demonstrates that the current system can detect these pitting defects, as shown in Table 3. The pitting defects are characterized by calculating their approximate area from the point cloud as described in algorithm 1.

Algorithm 1: Defect Area Estimation

Input: Point-cloud array $[x_i, y_i, z_i]$, ground-truth image [hereby referred to as “mask”]

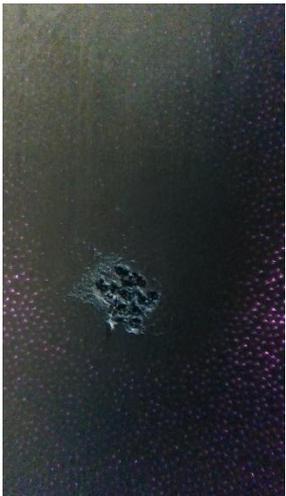
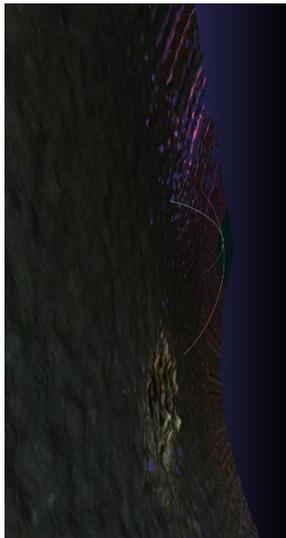
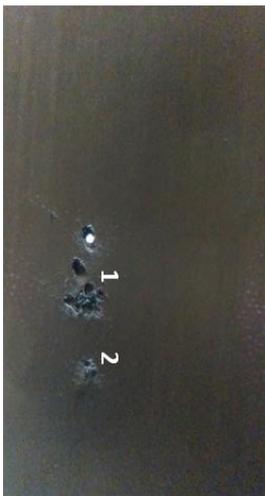
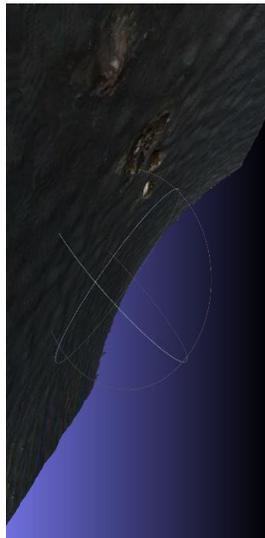
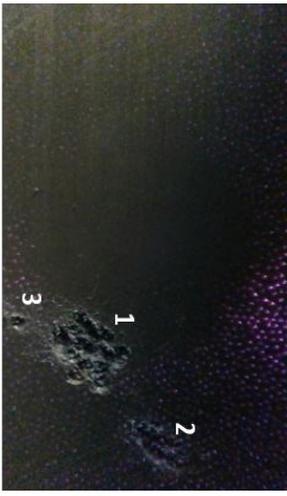
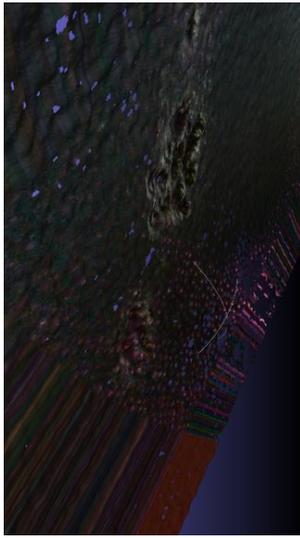
Output: Area

Function calculate_area(pointCloud, mask)

1. Compute contours from mask
2. Sort contours by area
3. Select contour corresponding to pit defect
4. Use contour coordinates to extract corresponding point cloud points
5. Compute convex hull of selected points.
Convex hull orders the points anti-clockwise.
6. Compute area using convex hull

Return area

Table 3: Images and reconstruction samples from the newly acquired pipe sample dataset

Image	Reconstruction	Defect type	Measurement
		Pitting	Area: 2306.5 mm ² Depth: ~1 mm
		Pitting	Pitting - 1: Area: 1433.5 mm ² Depth: ~3 mm Pitting - 2: Area: 356.9 mm ² Depth: ~1 mm
		Pitting	Pitting -1: Area: 2364.7 mm ² Depth: ~1 mm Pitting-2: Area: 1108.8 mm ² Depth: ~2 mm Pitting-3: Area: 188 mm ² Depth: ~1 mm

2.5 Surface roughness profile – Signal vs Noise at high resolution

An interesting observation while capturing the images at the (1280x720) pixel resolution and depth unit $1e-5$ was that the images captured by the depth sensor showed patterns similar to the ones found on the wall as shown in figure 35. To confirm whether this pattern was indeed the surface roughness profile, or it was the sensor noise, a comparative analysis was performed between the rough wall and a baseline smooth wall. Three images were captured using the camera at a distance away from the surface pointing perpendicular to it, approximately with the same camera settings. Aggressive temporal smoothing was performed to average out temporally varying noise signals with large temporal filters. The stabilized image was analyzed using a histogram of depth values. The spatial domain is inspected using the depth histogram, and this would directly provide the distribution of depth information. The smooth image and roughness-level 1 image distributions show minimal differences between the two cases. Therefore, we can conclude that the detection of surface roughness from depth maps at the levels indicated by the middle image in figure 35 will be unreliable. However, at a higher degree of roughness in the millimeter scale and above, it can be detected, as shown by the histogram on the far right in figure 36. Further analysis on whether a noise signature can be extracted for the image in the middle will be done as part of future work.

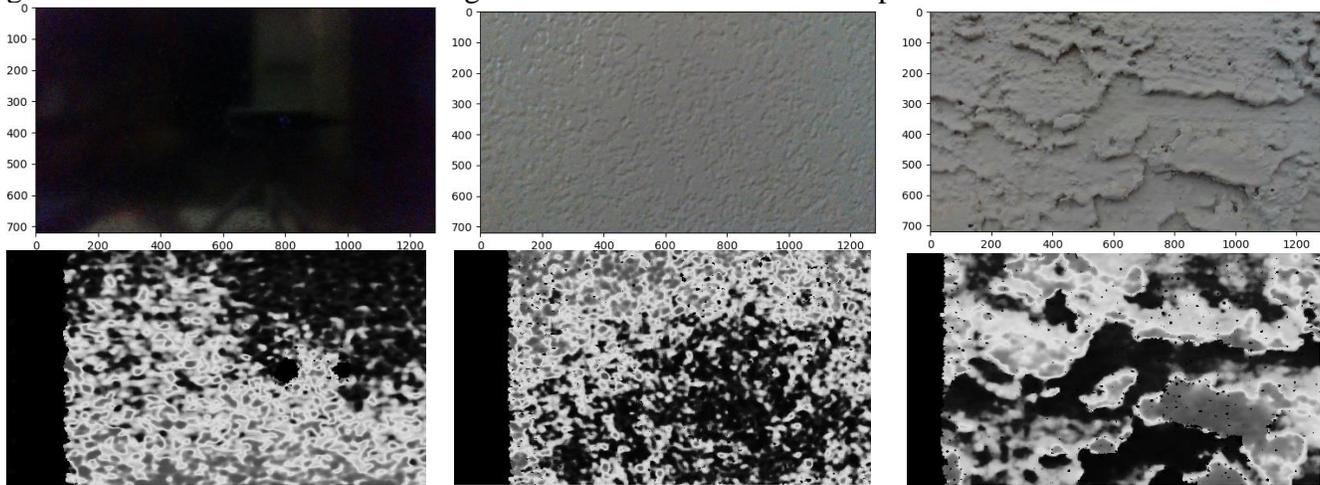


Figure 35: (Left) Smooth Surface (Middle) Roughness Level 1 (Right) Roughness Level 2 : Top row: RGB Images, Bottom Row: Depth maps in white to black scale - Whiter shades are closer. The dramatic contrast occurs because the range has been limited to the minimum and maximum depth values to highlight differences in depth. The camera was placed directly perpendicular to the surface and this was verified by ensuring that the depth did not vary substantially.

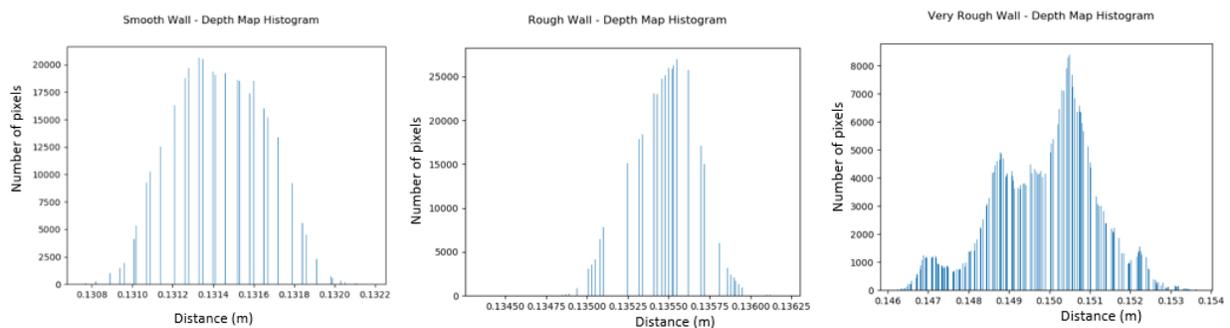


Figure 36: Histogram: (Left) Smooth Surface (Middle) Roughness Level 1 (Right) Roughness Level 2

2.6 Effect of adding two cameras in series

The resolution of the defect was shown to improve with the increase in camera resolution from 848x480 pixels to 1280x720 pixels, along with a higher sensor noise profile, in sections 2.4 and 2.5. To further attempt an improvement in the reconstruction, adding one more camera in series with the original camera was proposed. The D435i system is an “active-stereo” system, that uses a combination of color imaging and an IR dot pattern to perform stereo matching, and the dot patterns provide additional texture onto

texture-less scenes and these patterns are not visible. The color sensor detects these patterns and are substantial only in the case of high laser power. An increase in dot pattern density was expected to improve the matching quality by providing even more density of textures for the matching algorithm, however, no significant differences were observed. Another possible improvement for the matching algorithm could be to reduce the size of the IR dots. This would mean that we can have smaller dots, with a denser pattern. We have not tested this yet, as it would require an external IR dot projector to be installed. This may be incorporated as part of future work.

3. Task 2: Integrating the Data Acquired from the Camera system with the Fully Convolutional Fusion Network

3.1 Neural network architecture

In the previous quarter, the network used for semantic segmentation was a Fully Convolutional neural network with a pretrained VGG backbone, as shown in figure 37.

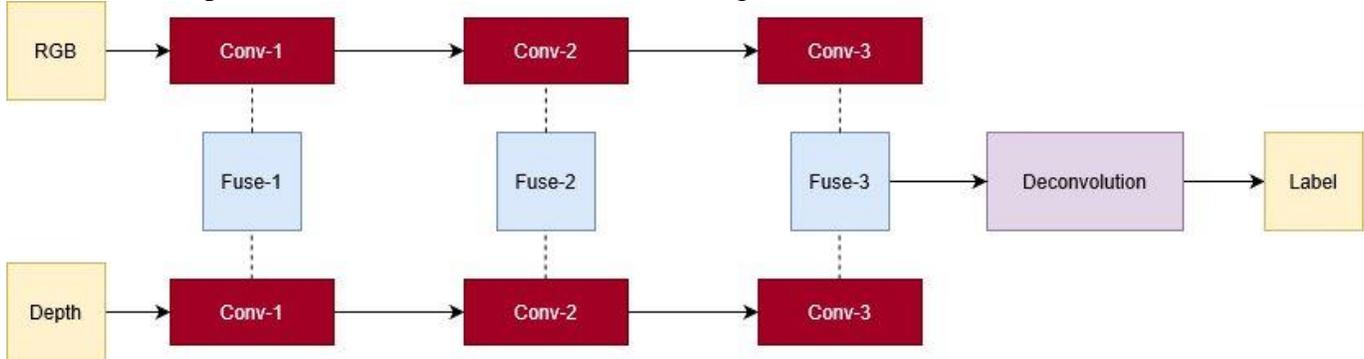


Figure 37: Previous neural network architecture with VGG-16 and Deconvolutional upsampling

In this quarter, the VGG backbone is replaced with a ResNet backbone [5], [6]. The ResNet architecture consists of a modular structure, with the number of parameters being controlled by varying the number of modules, and the structure of each module. ResNet-18 is the smallest available pretrained network, and we use this for our dataset. The ResNet-18 module consists of a dual path encoder architecture, with one path for the depth and the other for the RGB image. This is in keeping with the previous work, where multiple fusion blocks are used at different points of the architecture to ensure that the RGB and depth information are combined. The processing after the ResNet backbone produces an intermediate encoder output. This output is used to compute an attention map. Each fusion block output is then fed to an Atrous Spatial Pyramid Pooling (ASPP) layer, after which the deconvolution layers from the previous quarter are replaced with an up-sampling operation to reduce the number of weights. The difference in adding the deconvolutional layers is discussed as well in the results section. Figure 38 shows a summary of the baseline architecture.

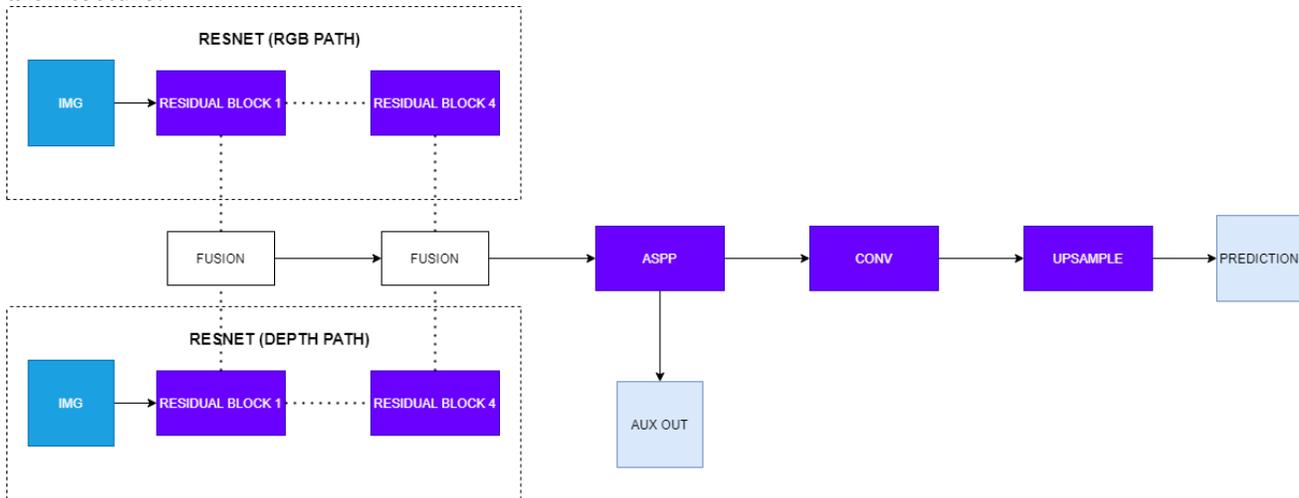


Figure 38: Baseline architecture with a ResNet backbone for this quarter’s experiments

Resnet Architecture

Neural networks can be considered universal function approximators. Given enough parameters and data, a neural network can, in theory, fit any continuous function in R^n . In practice, a single layer network overfits to the training data, and this led to the need for deeper networks. Compared to earlier CNNs which had less than 10 layers in total, deeper networks such as the VGG network architecture resulted in better performance. However, this improvement in performance reached its limits when the number of layers reached the order of 100, where the well-studied vanishing gradient problem comes into the picture. This is a performance reduction caused by information-loss, rather than overfitting. Backpropagation does not handle very deep networks well, and a solution was proposed by He et al. in [5], [6]. Neural networks approximate functions based on the training data. Deeper networks with more layers can approximate a different class of functions as compared to shallow networks. A key problem in designing deep neural networks is that one does not know how many layers are needed to get to the right function class. Adding new layers changes the representations learned in an unpredictable fashion. Residual networks work by adding a so-called “skip connection” from one layer to another, deeper layer, as shown in figure 39 (a), resulting in a nested representation of functions. Each residual block performs an “identity mapping”. The output of a layer L_k is fed into the input of layer L_{k+i} as an additive component to the output from layer L_{k+i-1} , as shown in figure 39 (b) Further discussion on identity mappings, residual networks and their mathematical properties are described in [5].

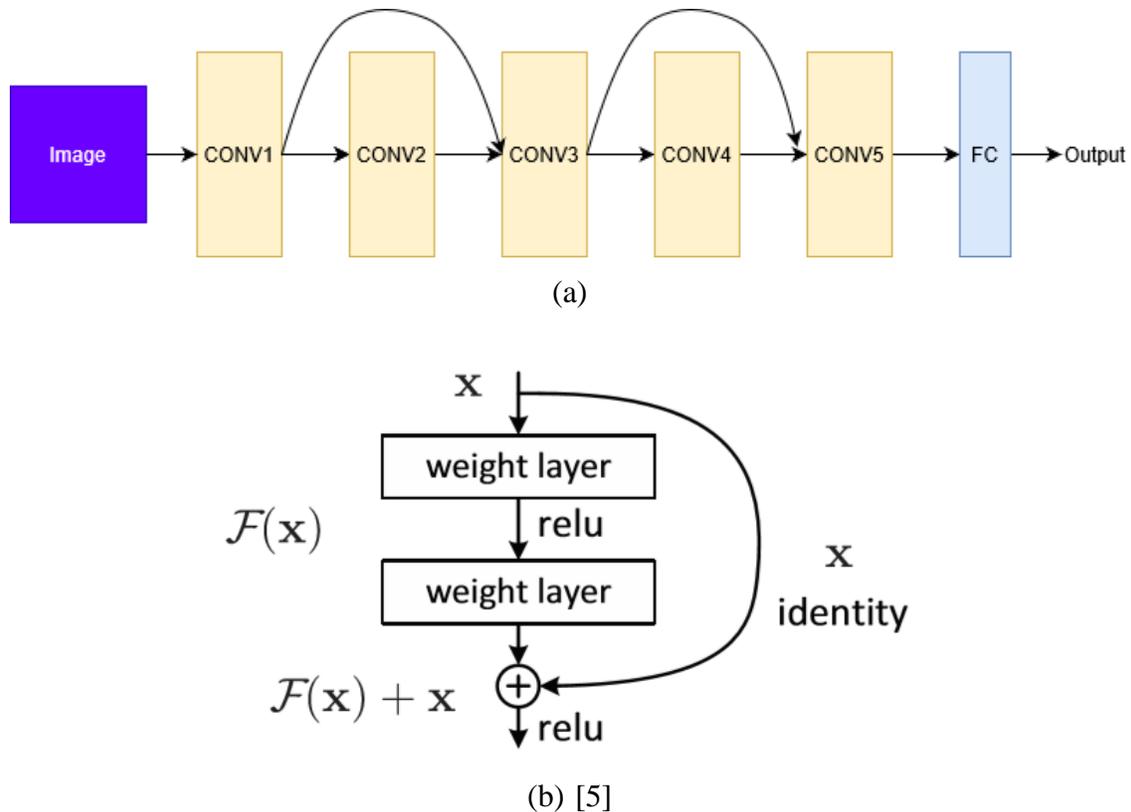


Figure 39: Residual network learning structure: (a) General structure of ResNet blocks (b) Residual connection and Identity mapping in a ResNet block

Atrous Spatial Pyramid Pooling (ASPP)

The repeated application of convolution and pooling layers in a deep neural network reduces the dimensionality of the input data. Very deep networks often end up with a low dimensional representation that does not contain any fine-grained information. One way to preserve fine grained information in deeper layers is to have a convolution layer that does not dramatically reduce the size of the features. Having larger filters will introduce more parameters, so this will not be efficient for very deep networks. The solution proposed by [7] is the ASPP layer, consisting of “dilated convolutions”. Each filter has a size determined

by the “rate” parameter and the kernel size. The rate parameter tells us about the internal padding of the filter and increasing the rate parameter will result in a larger filter with more internal zero padding. This not only helps improve the size of the feature maps, but also enables capturing features at different fields of view (FOV). Highly accurate localization is achieved using small FOV and spatial context is improved with larger FOV. The ASPP formulation is shown in figure 40.

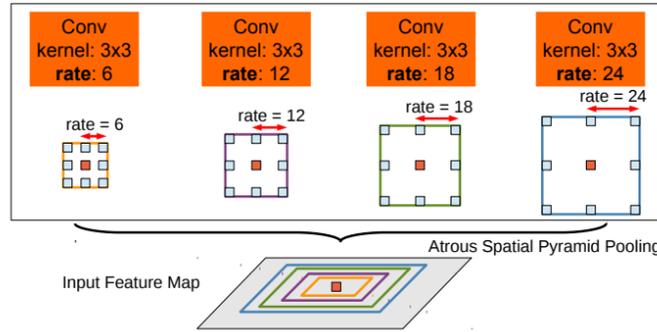


Figure 40: [4] ASPP Layer visualized with various filter sizes: The ASPP block has a pyramidal architecture as recommended in [7]

Fusion Block – Attention layers

The fusion block was modified from the previous quarter (the older fusion block is shown in figure 41(a)) to incorporate an additional feature fusion element to see whether this impacts performance on our dataset as shown in figure 41(b). This block was taken from the paper [4]. Attention is a concept used frequently in the computer vision literature and can be useful to ensure that the network learns to highlight the regions of interest better based on the input data and the ground truth. The first sub-module of the fusion block is called the “Feature Separation Module”, which [4] demonstrates empirically that it suppresses noisy depth features. The second sub-module is the weighted non-linear fusion block, which is essentially the same module proposed in the previous quarter. The modification made in this quarter is to make this sub-module a convex combination of RGB and depth features by constraining the weights of the respective streams to sum to 1 and to always be positive. The combined fusion block is shown in figure 41(b).

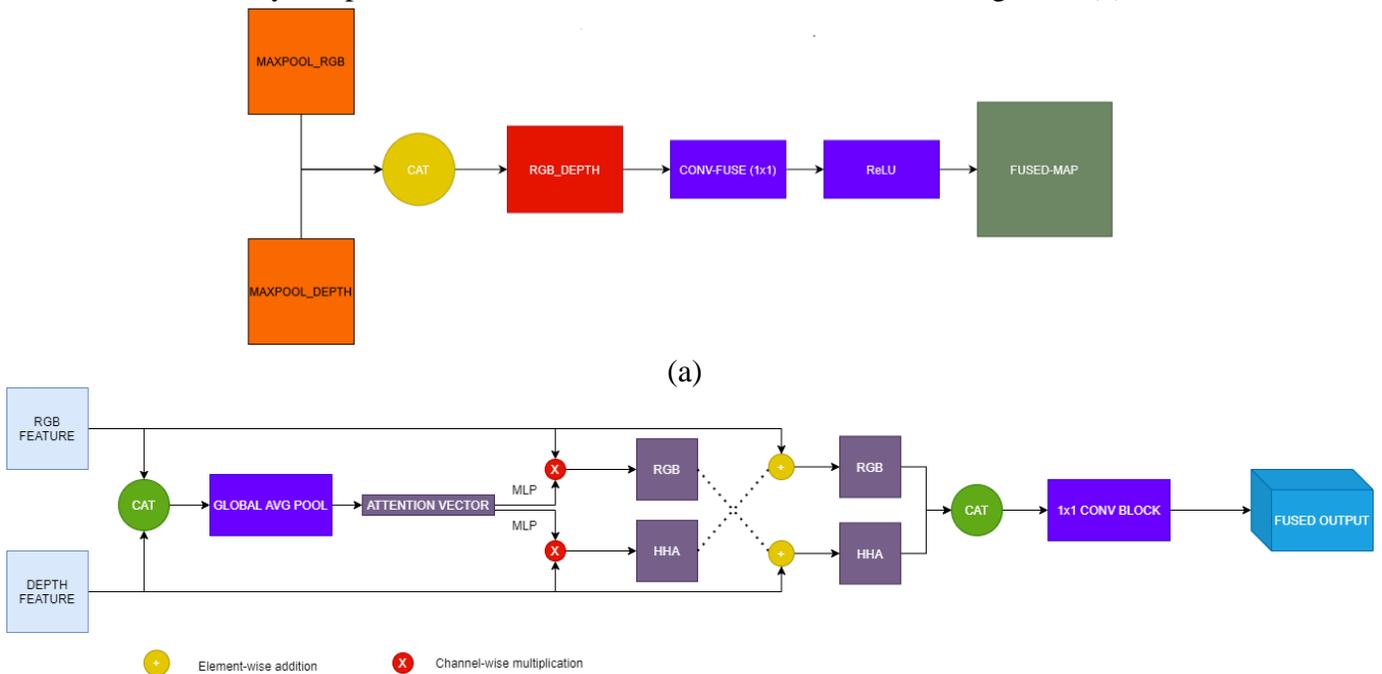


Figure 41: (a) Old fusion block (b) Fusion block proposed in this quarter

Learning rate scheduling policy

In previous quarters, we had used a linearly reducing learning rate policy for the neural network. This meant that the learning rate was reduced uniformly by a certain amount after a certain number of epochs. As [7] have shown, using a poly learning rate policy as shown below yields small improvements to the segmentation performance.

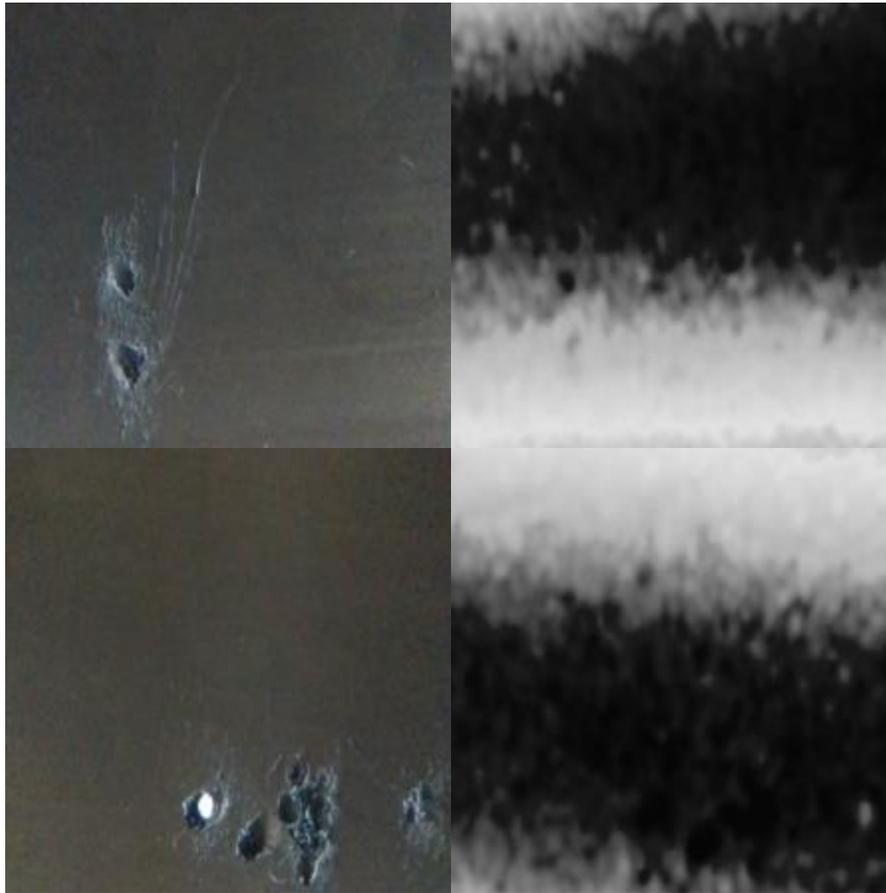
$$LR = \left(1 - \frac{i}{n}\right)^\alpha$$

Here, n is the total number of iterations, i is the current iteration, and α is the learning rate power, set to 0.9. A lower value of alpha results in a less steep curve.

3.2 Results and discussion

Dataset:

The training set consists of the new images acquired from the pipe sample approximating pitting defects. The depth map settings were modified in order to be able to detect these localized defects and approximate the pitting depth. Some image samples are shown in figure 42.



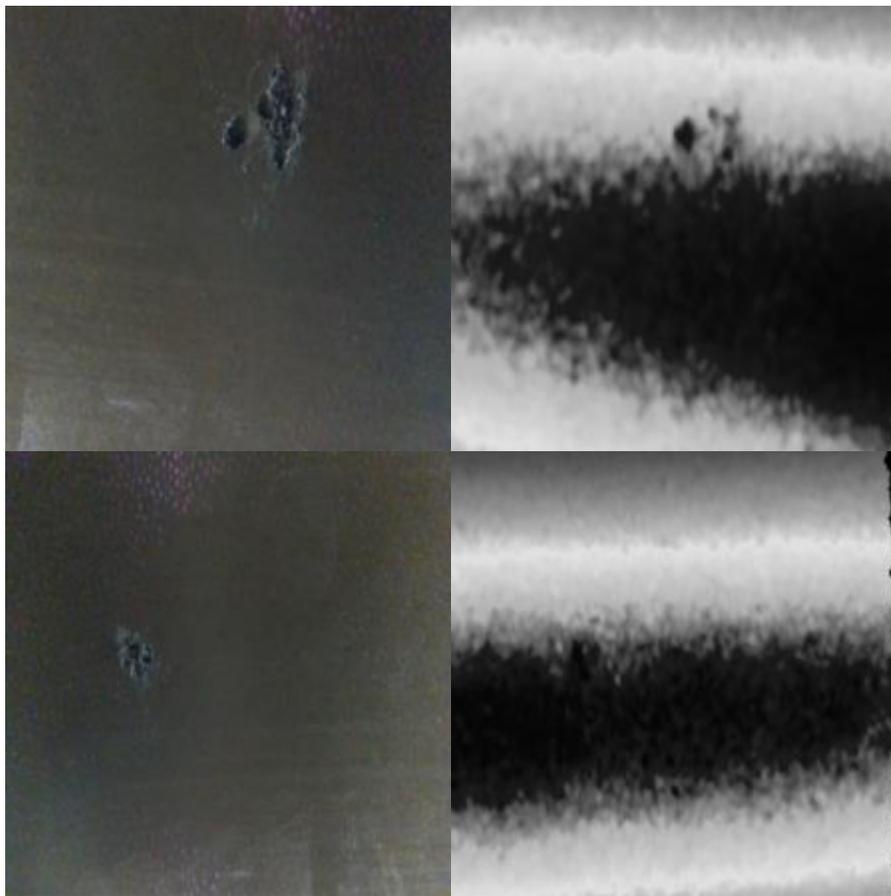


Figure 42: RGB and Depth map pairs for the training set

VGG-16 with new RGB-D Fusion Block

The first experiment involved using the VGG-16 network with the non-linear fusion block with the Feature Separation Part (FSP) preceding the NL-fusion block. The network used the binary cross entropy loss for training and the labels were one-hot encoded for comparison in accordance with that format. Two dataset sizes were tested, one with 200 samples and the other with 500 samples, to test the effect of increased dataset size on performance metrics. The performance metrics used in these studies are the loss, and the mean intersection over union (mIU) score. The mIU computes the degree of overlap between the ground truth segmentation and the predicted segmentation, and ranges from 0 to 1 with the best value being 1.0:

$$mIU = \frac{|A \cap B|}{|A \cup B|}$$

The one with 500 samples converged faster in around 50 epochs with an mIU of 0.985 and the one with 200 samples reached an mIU of 0.965 but took twice as many epochs to settle the loss function to a stable value, as shown in figure 43 (a). Removing the FSP layer produced negligible reductions in performance as shown in figure 43 (b).

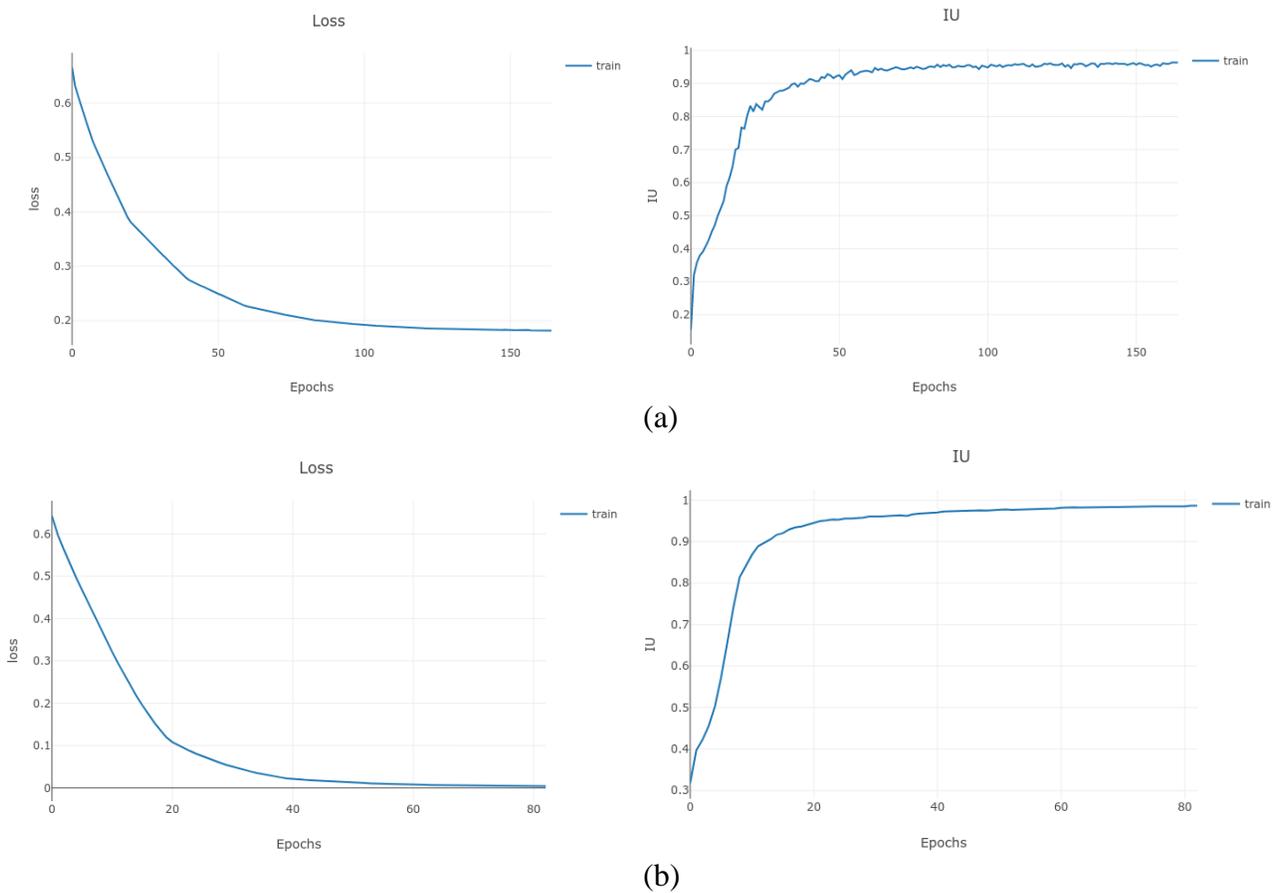


Figure 43: Plotting the train loss and IU for: (a) VGG-16 with 200 training samples (b) VGG-16 with 500 training samples

ResNet-101 with ASPP and Bi-Directional Cross-Modality Fusion [4]

The baseline case involves using a ResNet-101 network with ASPP attached at the final layer of the encoder module as shown in figure 38. There is no deconvolutional upsampling layering, instead replacing them with direct interpolation back to the original size. The loss function used was a composite cross-entropy loss:

$$L(y, Y) = L_{pred}(y_{pred}, Y) + \lambda * L_{aux}(y_{aux}, Y)$$

L_{aux} – loss over the encoder output (A low resolution representation) weighted by $\lambda = 0.2$

The mIU for this experiment was 0.68 and reached an accuracy of 0.72, as shown in figure 44.

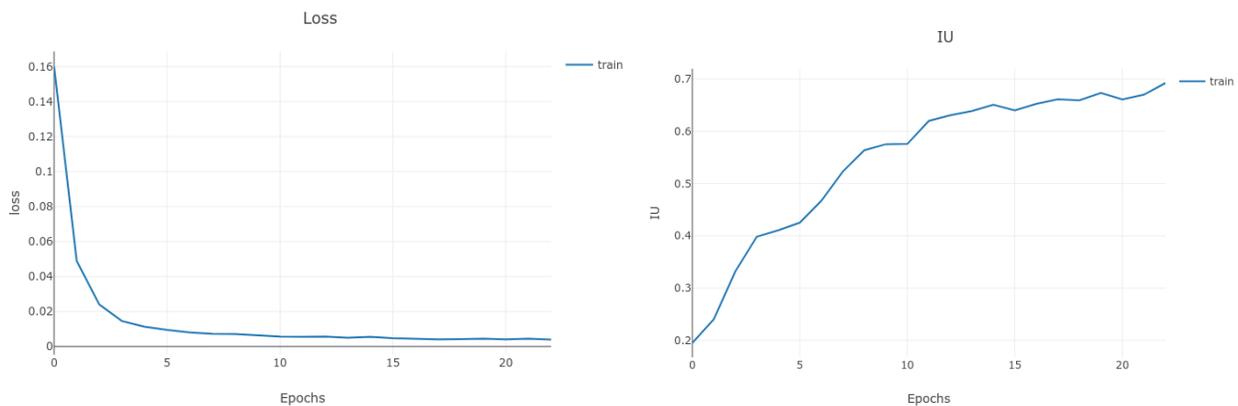


Figure 44: ResNet-101 baseline results plotting the train loss and IU

ResNet-18 with ASPP and Bi-Directional Cross-Modality Fusion

Another set of experiments was conducted with a lightweight ResNet-18 network. The ASPP layer and the

fusion block were kept unchanged from the previous subsection but the backbone was changed from ResNet-101 to ResNet-18. The loss functions were varied as well, with the cross-entropy composite loss as described in the earlier section being used as the baseline and a new loss composite loss function was defined as follows:

$$L(y, Y) = L_{pred}(y_{pred}, Y) + \alpha * L_{IU}(y_{pred}, Y) + \lambda * L_{aux}(y_{aux}, Y)$$

L_{IU} – Lovasz softmax loss, α – lovasz factor

The Lovasz softmax loss is the same as that described in Q3, but instead of using the loss in isolation, the loss was used as a composite with a variable weight α that depends on the loss L_{pred} . The value of α increases as L_{pred} reduces, leading to the use of L_{IU} to fine-tune the loss function after L_{pred} converges. The results indicate that adding the Lovasz loss improves the baseline IU of 0.56 with only cross entropy to approximately 0.6 after adding Lovasz loss, despite an increase in time to converge, as shown in figures 45 (a) and (b) respectively.

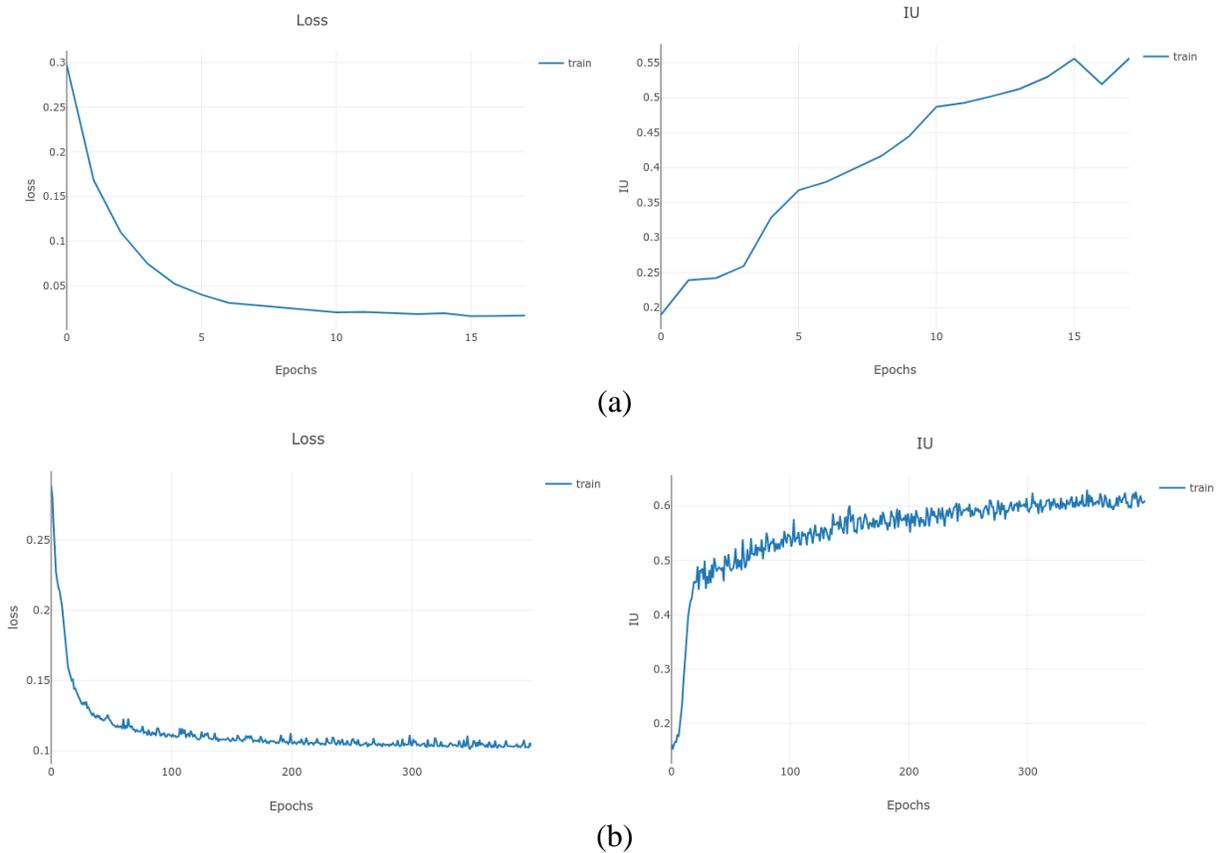


Figure 45: Plotting the train loss and IU for: (a) ResNet-18 with only cross entropy loss (b) ResNet-18 with cross entropy and Lovasz softmax loss

ResNet-18 with ASPP and Transposed Convolutional Decoder

The next set of experiments involved investigating the effect of adding deconvolutional layers to upsample the image instead of using interpolation. Another addition to the network was to add the ASPP block to the lower-level feature output in the first ResNet block. The effects were not supported with improved results, and both the mIU and the accuracy dropped precipitously. This is an unexpected result, as adding deconvolutional layers provides a degree of flexibility to the upsampling process. Long-range contextual dependencies are not important to this dataset so ASPP layers do not play an important role here. However, more experiments with a larger dataset with more varied classes in various spatial locations in the image will demonstrate whether the ASPP layer is indeed effective or has only negligible contributions. The results for this set of experiments are shown in figure 46.

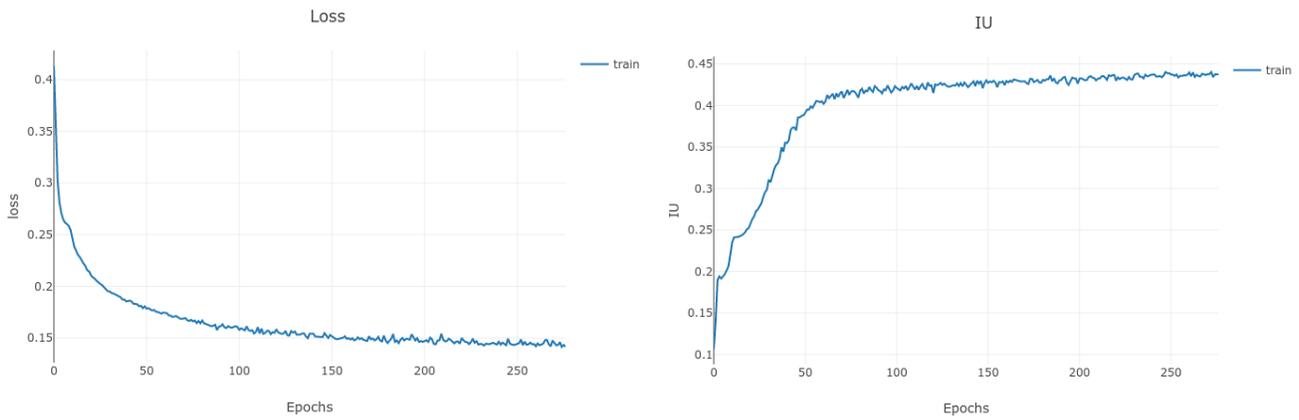


Figure 46: Plotting the train loss and IU for the case where a deconvolution layer is added to the ResNet-18 network

Measurement Results with the best network

The best performance was obtained with the VGG-16 network with the FSP layer preceding the non-linear weighted combination. A hold-out image is an unseen image which is out of sample from the augmented training set. We use a hold-out image for our tests in this case as the number of samples in our dataset is limited. The measurement was conducted with two images, one from within the sample, but with no augmentation to make it look slightly different from all the augmented samples in the training set, and the second image was the hold-out sample. The results are visualized in figure 47 and tabulated in tables 4 and 5. The image on the top row of figure 47 shows the hold-out example. The hold-out sample contained a pitting defect and two cracking defects. The network was unable to detect the cracking defect but segmented the pitting defect out. The within sample result shows that all the defects were detected, as shown in the bottom row of figure 47.

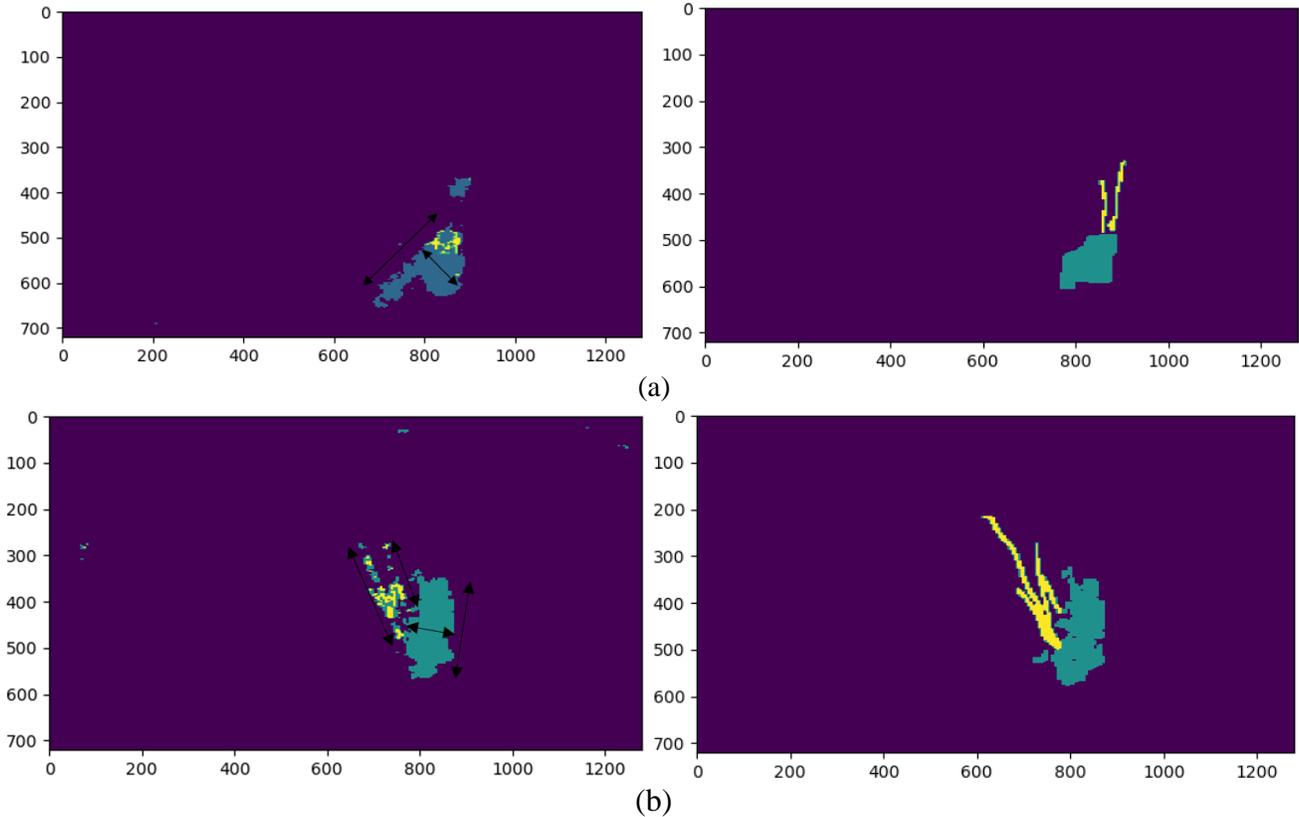


Figure 47: Visualization of measurement: (a) (Left) Prediction and (Right) ground truth for hold-out sample; (b) (Left) Prediction and (Right) ground truth for sample within the train distribution

Table 4: Measurement results for the hold-out case

Image	Defect Type	Measurement	Estimated (mm/mm ²)	Actual (mm/mm ²)	Error (%)
	Pitting	Area	1330	835	37.35
		Depth	1	~3	~200

Table 5: Measurement results for the case within training distribution

Image	Defect Type	Measurement	Estimated (mm/mm ²)	Actual (mm/mm ²)	Error(%)
	Pitting	Area	2603	2375	8.76
		Depth	0.001	~2	~100
	Cracking-1 Root Middle	Length	71.9	78	8.48
		Width	2	1	~100
	Cracking - 2 Root Left	Length	34.7	37	6.62
		Width	3	1	~200
	Cracking - 3 Root Right	Length	50	56	12
		Width	3	1	~200

4. Summary and Future Work

4.1 Task 1

- **Summary of the sensitivity analysis**

Depth camera D435 is suitable in analyzing the depth of even small objects. The Lidar camera only works well for larger camera-to-object distance which is not suitable for our application if the camera is facing directly to the pipe wall. However, if the camera is facing along the longitudinal direction of the pipeline, the Lidar camera can be potentially used for pre-screening of pipeline defects. Detailed results of the sensitivity analysis have been reported above.

- **New representation proposed for depth stream:** A new curvature and normal vector-based representation for the depth stream was proposed. This is termed the DNC representation (Depth-Normal-Curvature). The technique to compute these metrics from the depth was presented using concepts in differential geometry of surfaces and is demonstrated on sample surface data. In the next quarter this will be extended to real pipe data.
- **Pitting Defects images:** Pitting defect images on both flat walls and pipes were procured using the camera system. Welding defects will be captured in the next quarter if metallic pipes are available.

- **Adding a camera in series and IR dot sizes:** The effect of adding one additional camera in series with the first camera was studied and found to have minimal effects on surface reconstruction. However, it is hypothesized that the reduction of IR dot pattern size will improve reconstruction results. This will be tested in future quarters once we procure an external IR dot projector.
- **Noise analysis – Preliminary work:** The depth maps procured for pitting on the wall showed interesting patterns similar to that seen on the wall. To test whether this was indeed the surface roughness or the noise, analysis was done on walls with varying degrees of roughness and results were reported. In the next quarter, focus will be on identifying if there is a noise signature that can be procured from the depth information to detect the presence of surface roughness or if the signal is too weak to extract any signature from.

4.2 Task 2

In this quarter, several modifications were made to the network architecture with the newly acquired pitting data and evaluated if there was any benefit. Lovasz softmax loss along with cross entropy loss used in combination improved mean IU scores by approximately 3-5 points. However, the best performing network for the dataset currently procured was still the VGG-16 architecture backbone. An addition was made to this architecture in the fusion layer using a method that suppresses noisy depth features. In the next quarter, features produced by these layers will be visualized to observe if there is any visual difference in various fusion blocks that might explain whether the design of the fusion block has a major impact on performance. As more data is acquired, the test set results and measurement accuracies will become more reliable.

References

- [1] Intel RealSense, “L515 Datasheet.” [Online]. Available: <https://dev.intelrealsense.com/docs/lidar-camera-l515-datasheet>.
- [2] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, doi: 10.1007/978-3-319-10584-0_23.
- [3] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, doi: 10.1109/CVPR.2015.7298965.
- [4] X. Chen *et al.*, “Bi-directional Cross-Modality Feature Propagation with Separation-and-Aggregation Gate for RGB-D Semantic Segmentation,” *arXiv*. 2020, doi: 10.1007/978-3-030-58621-8_33.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, doi: 10.1007/978-3-319-46493-0_38.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, doi: 10.1109/CVPR.2016.90.
- [7] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “DeepLabv3+,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.