Application of computer vision methods for automated wooden planks length measurement

Lukas Zabulis
Department of Automation
Kaunas University of Technology
Kaunas, Lithuania
lukas.zabulis@ktu.lt

Arūnas Lipnickas
Department of Automation
Kaunas University of Technology
Kaunas, Lithuania
arunas.lipnickas@ktu.lt

Rytis Augustauskas
Department of Automation
Kaunas University of Technology
Kaunas, Lithuania
rytis.augustauskas@ktu.lt

***Abstract*—Automated inspection is a principal component in modern manufacturing aiming for Industry 4.0. With the fast development of digital image processing and machine learning algorithms, quality control is often performed by computer vision systems (CV). Following the presented topic, this paper applies the CV system for non-direct length measuring of objects moving through a conveyor. The approach involves image segmentation by a convolutional neural network and segmented object processing principles. Since high-resolution images were used, segmentation was performed in a patch manner. The algorithm also involves a custom contour analysis and the application of the least-squares method for the midline of wooden plank extraction. The proposed approach is robust to the object's non-alignment and merging factors as well as shape and wooden texture variance. The presented algorithm was tested on a custom wooden planks database gathered in a real EURO pallets production environment and confirmed to work as expected ensuring *±1mm* length measuring accuracy and *0.97s* processing timing meeting the real-time requirement of the specified system.**

Keywords—object segmentation, deep learning, computer vision, digital image processing, non-direct measurement.

I. Introduction

The manufacturing industry has faced growing challenges in recent years. The sector needs to keep up with customer expectations for high-quality, reliable products at an affordable cost. Thankfully, Industry 4.0 with its cutting-edge technologies can alleviate the biggest challenges facing the industry. Along with highly automated manufacturing lines, smart warehouses, and IoT technologies, machine learning, and computer vision (CV) systems come into play. CV, a subset of artificial intelligence, allows computers to take in information from digital images and then make decisions based on that information. When applied correctly, CV allows computers to carry out the kinds of repetitive quality ensuring tasks that skilled workers would otherwise be doing. CV can improve efficiency and attention to detail while freeing up workers' time for other tasks. Moreover, estimation of product quality properties in each step and taking out spoiled items can cut down the overall cost since the fault can be detected in the early stage of production. In this way, more sustainable manufacturing can be achieved while the subsequent operations are not performed and materials are not wasted on already defective products. The vast majority of quality estimation relies on visual inspection and other vision-based parameters (e.g., length, shape, angle). CV quality assessment is the most popular approach for industrial applications [1].

This article explores the application of computer vision methods for object length estimation on a conveyor. The aim was to create a CV system for a EURO pallet manufacturing company with the ability to automatically measure wooden planks, supplied at the early stage of production. EURO pallets, manufactured in the company must meet the standard of the European Pallet Association (EPAL). One of the main focuses regarding this standard is dimensional tolerance. Since current production deals with manual measurement involving human intervention, an automated approach is required for overall company performance and competitiveness.

The paper is organized as followed. Section 2 covers related computer vision applications for quality and measurement evaluation tasks. Section 3 explains the segmentation neural network itself and its configurations used in this paper. Section 4 presents the database creation methodology. Section 5 covers neural network training and the approach behind the plank length measuring algorithm. The results are given in section 6 followed by conclusions in section 7.

II. Related Work

There can be found papers that propose a variety of computer vision applications for automated quality or measurement evaluation. As it can be seen, complex visual patterns or scenery require more sophisticated solutions. For example, researchers in [2] are utilizing lightweight convolutional autoencoder architectures for drilling segmentation in textured wooden furniture boards to estimate drilling position from grayscale images and compare it with a template of furniture part. Another machine learning approach is proposed by Gonzalez et al. [3] where the authors are engaging small convolutional neural network architecture (4 blocks with residual connections) for metal welding defects detection from infrared imaging with a further defective area length estimation. Roberts et al. [4] describe convolutional neural autoencoder solution with densely connected block for crystallographic anomaly detection in steel images and quantification metrics (consisting of diameter, density, etc.) for defects detection from segmented visual context. A computer vision-based approach for cracks (that appeared from fatigue) detection and length estimation in the aluminum alloy is described by Yuan et al. [5]. For quite complex metal surface inspection authors are using MobileNet-based [6] convolutional neural networks for defect detection. From the extracted defective region, the

2022 18th Biennial Baltic Electronics Conference (BEC) | 978-1-6654-6295-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/BEC56180.2022.9935598

crack length estimation is done. Al-Hasanat et al. [7] present a deep learning-based solution for object detection and distance from camera estimation. A fully applicable tool for visual context analysis of small living organisms is presented in [8]. The author presents the software which utilizes conventional image processing and ResNet-V2-101 [9] convolutional neural network for small organisms' detection and quantitative analysis. Non-direct mass estimation to improve breeding and adjust feeding is proposed by Zhang et al. [10], where authors are using RGB images of fish to estimate mass. Researchers are utilizing the GrabCut algorithm to segment fish from quite a distinctive pattern and measure the height, width, area, and perimeter of the context. This data is passed to 3-layers fully connected network for mass estimation. A more visually complicated solution for fish length measurement is discussed in [11]. Deep neural network-based detection and segmentation models are used for relevant information extraction and measurements are estimated afterward. Article [12] describes the object measuring technique firstly determining the value of pixels per metric, given a reference object with known measurements (width and height). However, proper measuring relies on easy identification of objects with high contrast between background and objects themselves.

To sum up the research of the above papers, we can notice two main different manners of geometrical or any related parameters evaluation. If the objects in the image are clearly distinguished, then simple image processing techniques help to extract relevant information. Otherwise, more sophisticated (data-driven) models are required for proper object segmentation or localization. However, few papers utilize patch-based image segmentation and analysis of multiple objects per image with merging and non-alignment factors. Thus, we introduce an end-to-end wooden plank length measuring algorithm involving conventional computer vision and deep learning methods. The proposed approach is applied in a real production environment.

III. Deep Neural Network Model

Slightly modified based on U-Net [13] convolutional neural network architecture (Figure 1) selected for this kind of application for its ability to extract high dimensional feature space making U-Net a powerful tool for information extraction in a complicated context. Moreover, the proposed architecture outperforms other segmentation models in terms of processing speed. Authors in [14] compare U-Net along with Mask R-CNN, RefineNet, and SegNet networks. Results showed a significant advantage over the U-Net network since computation is twice as fast as mentioned networks (Table 1). In this case, fast computation does not sacrifice segmentation accuracy drastically.

Table 1. Computation time and number of parameters comparison between different segmentation network architectures [14]

	U-Net	**MRCNN**	**RefineNet**	**Segnet**
Time, ms	466	1245	1143	911
Params, M	8	64	80	29

The architecture consists of two parts: an encoder (left) and a decoder (right). The input of the network is a normalized [0.0,…,1.0] grayscale image of size *128x128*. In this research, we utilized an architectural design of four downscales. In the first stage, *32* feature maps are employed. In the following stages of the encoder, *2x2* MaxPooling filters are used, and the number of feature maps is doubled. Also, we proposed a slight modification to a baseline U-Net introducing varying size convolution filters downscaling from nine to three in each stage of the encoder part. This modification led to increased segmentation accuracy since it is carried out on large-scale objects. Additionally, employing a larger feature kernel size at early stages improves the network's ability to reject image noise. In the decoder, reversed operations are performed. Opposite layers in the encoder and decoder are associated with skipped connections that allow transferring higher-level features from larger dimension layers. In the concatenation layer, the existing feature maps are combined with the corresponding feature maps extracted in the encoder part. The last layer uses a 1×1 convolution filter with a sigmoid activation function which performs as a binary classifier between two classes: background and plank. Pixel values of the segmented image vary in the range [0.0,…,1.0].

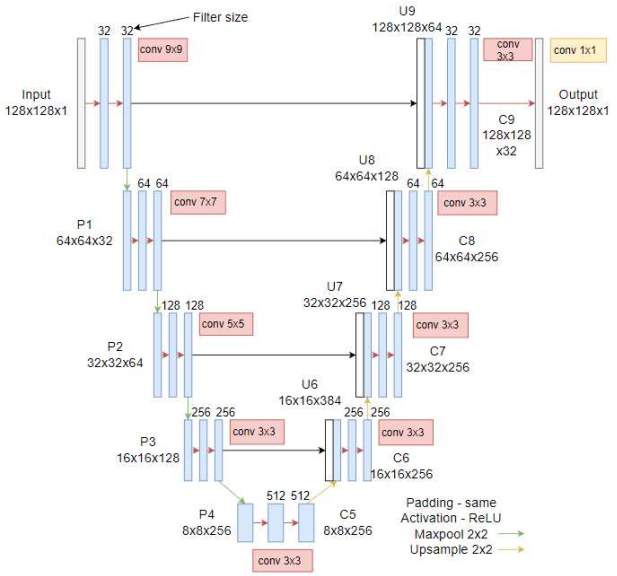

Figure 1. Slightly modified U-Net convolutional neural network architecture

IV. Database

A database of wooden planks pictures was gathered in a real production environment, using the CV system referenced in Figure 2. In production, wooden planks move through a conveyor in a batch of six. By convention, two synchronization functions are applied together to obtain an image. Frame sync is managed by an optical sensor which is triggered by moving objects. TTL signal provided by the rotary encoder manages each exposure of the camera sensor line by line. The Basler's line scan camera raL6144-16gm with NIKON AF Nikkor 24 mm f/2.8D optics was utilized. In addition, an industrial LED (1400 lm) was used for constant object illumination.

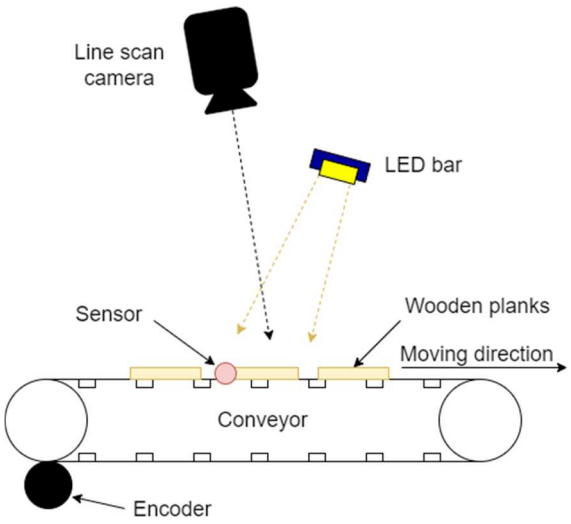

Figure 2. The structure of chain type conveyor CV system and its components

Some important parameters of the image database are given in Table 2.

Table 2. Parameters of image database

Parameter	Description
Number of images	35
Planks per image	6
Type	grayscale
Resolution (*width* x *height*), px	6144 x varying height
Bit depth	8

In order to generate image masks (used for segmentation network training), each plank within the image is approximated by a polygon using the open-source online annotation tool *CVAT*. Once planks and background were separated, individual masks were saved as binary pictures with pixel values *0* or *1*(Figure 3).

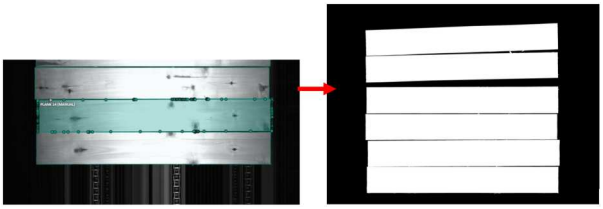

Figure 3. Polygon approximation of wooden planks (left) and saved binary mask (right)

The last stage of data preprocessing includes patch extraction of images and corresponding masks (Figure 4). It is necessary due to high image resolution which is not acceptable for conventional CNN with fixed input size. During experiments, the most effective patch size appeared to be *128x128* considering segmentation performance and accuracy. Patch database parameters are given in Table 3. It was created from 20 images with a wide range of texture variance. The rest 15 images were used for the algorithm (segmentation and length measuring) testing purposes.

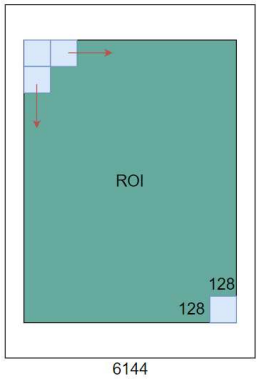

Figure 4. Methodology of patch extraction from the high-resolution images into patches by size 128x128 px

Table 3. Parameters of patch database

Parameter	Description
Number of high-resolution images	20
Patch size, px	128x128
Number of extracted patches	52460
Train-Test-Validation split	0.7/0.2/0.1

V. The Approach

The following part of the article describes object segmentation, and the plank length measuring algorithm itself.

A. Network training, object segmentation

For the U-Net (Figure 1) training *Adam* optimization algorithm was used with a learning rate of 10^{-4} and a batch size of *32*. Train-test-validation split is mentioned in Table 2. *BinaryFocalLoss* was utilized as a loss function. This specific function allows the model to focus on samples that are more difficult to classify, reducing the contribution of easily distinguishable samples. The training was performed in two stages. The first stage of training covers the whole patch database and reached *96.88%* pixel accuracy with the test set. Quality estimation was followed by hard classified patch extraction and next-stage training. This manner led to a pixel accuracy increase of *0.34* percentage points. Segmentation accuracy was determined by pixel accuracy (*PA*) and intersection over union (*IoU*) evaluation metrics (Table 4). The two-stage training took approximately 5 hours using a consumer-grade PC (Intel Core i5-2400 CPU @3.10GHz, SSD disk, 10GB RAM, and CUDA-enabled NVIDIA GeForce GTX 1060 6GB GPU using Tensorflow 2.6 backend).

Table 4. Segmentation results

	PA, %	***IoU***
Whole patch training database	96.88	99.1
Hard to classify patch database	97.22	99.2

The following part describes the high-resolution image segmentation technique. Once the image is captured, auto-cropping takes place to distinguish the region of interest (ROI). Cropping is performed by summing pixel values along

the *x* and *y* axes with the constant stride. Boundaries are determined with the help of pixel sum array derivative peaks exceeding the threshold. The early cropping approach significantly reduces image segmentation time since every second saved counts as a huge improvement in real-time applications. Patch extraction was performed using a sliding window approach without overlapping. Since cropped image size varies, it is necessary to incorporate a black frame ensuring patches cover the whole image area. Each extracted patch is then stacked into an array and held in the computer's memory. At this stage array of patches is fed into the previously described U-Net convolutional neural network with the fixed batch size. After segmentation, the network returns the same size stacked array of patches followed by the reconstruction of a full-sized segmentation mask. Figure 5 shows a visual explanation of the image segmentation algorithm.

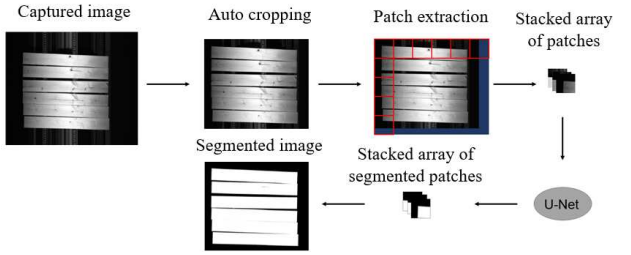

Figure 5. Segmentation algorithm

B. Mask processing and length measuring

After segmentation, mask pixel values vary in *0-1* range, hence thresholding method should be applied to properly distinguish wooden planks from the background. However, even after thresholding random noise occurs inside of object regions. Due to this convention, *floodfill (OpenCV)* method is applied and dark regions are removed within each of the planks. The image sequence in Figure 6 shows each stage of the mask processing technique starting from the original image.

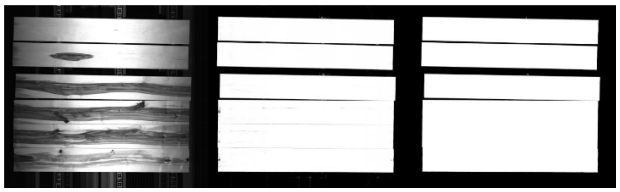

Figure 6. Mask processing using threshold and floodfill methods. Original image (left), thresholded segmentation mask (middle), processed mask (right)

Once the desired mask is obtained, *findContours (OpenCV)* algorithm is used to extract the curves joining all the continuous points along the object's boundary. Each contour point has its coordinates. As an additional feature, each contour is validated considering minimum perimeter, leaving only significant contours. For proper plank length measuring it is necessary to properly estimate the object's midpoints. Contour inspection by vertical lines with the constant stride appeared to be quite a reliable approach for midpoints estimation. Since uneven distribution and rotation of planks occur, some of them remain merged after segmentation (Figure 7, first contour from the bottom). As long as the plank's width remains constant, the potential number of them can be easily estimated within the merged contour. This is done by defining lower and upper bound constants. If one object is detected, then (1) condition should be met.

$$a < l < b \tag{1}$$

where a – minimum object width, b – maximum object width, l – contour object width in a specified region. If (2) condition is satisfied, then the contour region contains more than one object that needs to be examined.

$$l > b \tag{2}$$

Hence, the number of objects within a contour region can be calculated by dividing the contour object width in a specified region by the minimum object width as follows:

$$n = \frac{l}{a} \tag{3}$$

where n – number of objects within a contour region. If one object is detected, then the midpoint y coordinate is calculated by dividing l by *2* (x coordinate remains constant). In the case of merged objects, midpoints are estimated by dividing l by *2n*. Figure 7 shows described midpoints extraction procedure.

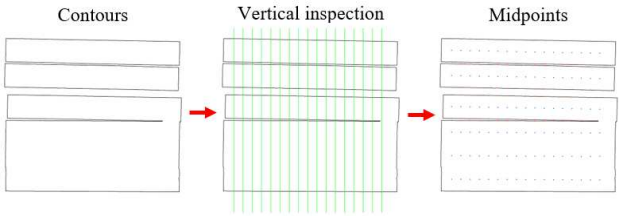

Figure 7. Planks midpoints extraction from contours

After the midpoints estimation, the algorithm proceeds to fit midlines using the least-squares method. LSM finds a polynomial (line in this case) with the help of summed squared error minimization between fitted line points and approximated points:

$$E = \sum_{j=0}^{k} (p(x_j) - y_j))^2 \tag{4}$$

where E – error, k – number of points, $p(x_j)$ – fitted line points, y_j – approximated points. Finally, the object's length is measured using Euclidean distance:

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \tag{5}$$

where (p_1, p_2), (q_1, q_2) – coordinates of the contour points along the midline. Distance in pixels can be easily converted to any of the desired units, given pixel per unit ratio. Due to contour line fluctuations, multiple measurements parallel to the midline were carried out and an average was considered as a final measurement. Figure 8 shows the final measurement of the planks in millimeters.

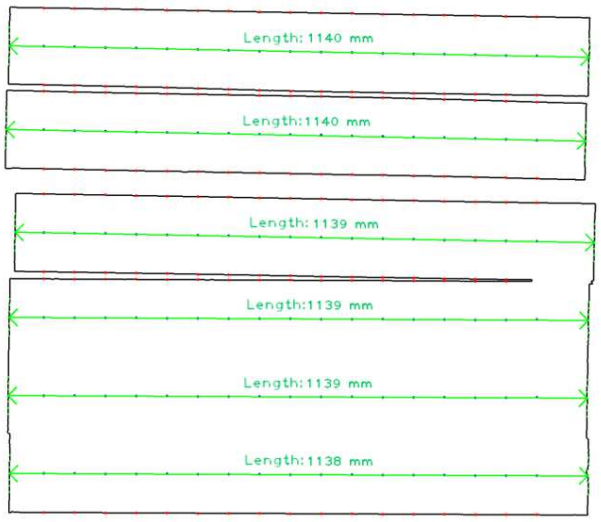

Figure 8. Contours measured

Evaluation of measurement accuracy was performed by comparing measurements of manually annotated images and actual images using the same algorithm. Results showed an average of ±3px fluctuations between the same measurements. The overall error of measurements considering the actual pixel-millimeter ratio is:

$$e = 3a = 3 * 0.288 = 0.864 \approx 1mm \qquad (6)$$

where e – error, a – pixel-millimeter ratio.

To sum up, the full algorithm flow diagram, from the image capture to the final measurement is given in Figure 9. Some of the measured planks are shown in Figure 10-12.

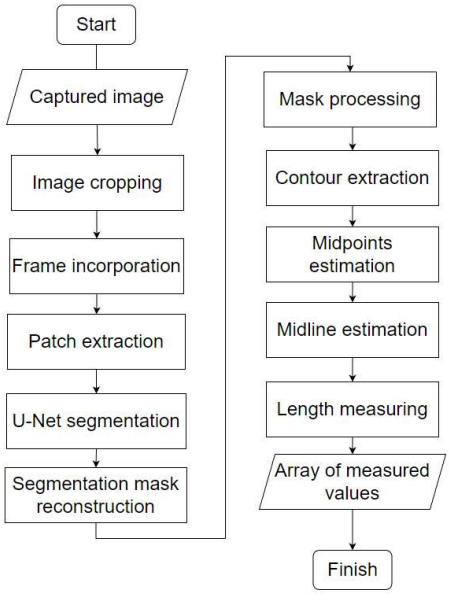

Figure 9. Full algorithm flow diagram

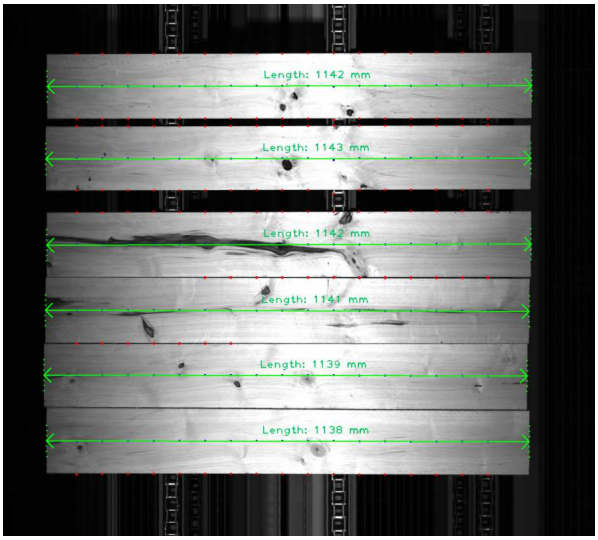

Figure 10. Picture of measured planks No.1

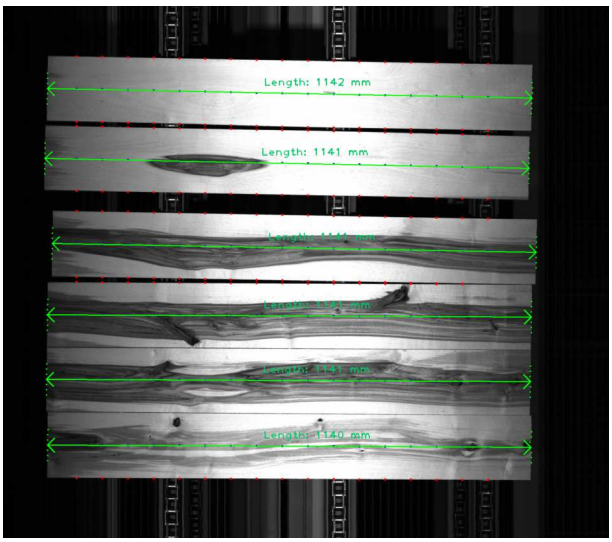

Figure 11. Picture of measured planks No.2

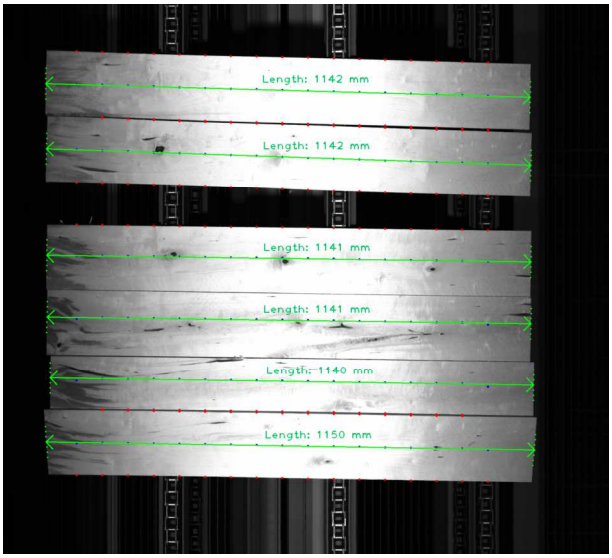

Figure 12. Pictures of measured planks No.3

VI. Results

On purpose to test the algorithm, *204* planks (34 images) were measured. In production, the acceptable range of plank length varies between *1137-1143 mm*. Once measured, 188 planks met the specified range requirement. Figure 13 shows the planks length distribution diagram. All the measurements are made with a possible error of *±1mm*.

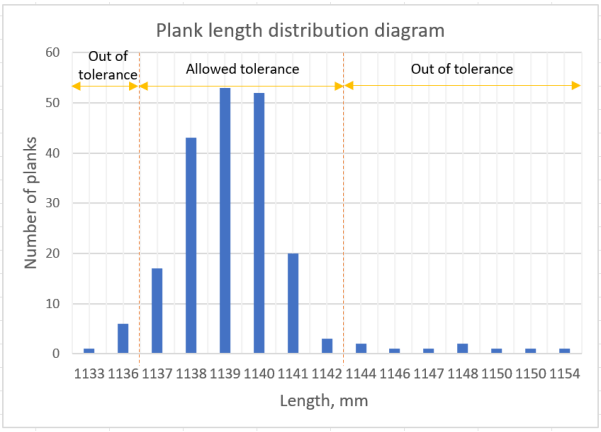

Figure 13. Plank length distribution diagram with specified tolerance bounds

During tests, the average cycle time of the algorithm appeared to be *0.97s* using the hardware configuration mentioned in section 5. As presented in Figure 14, segmentation is the largest compute-time contributor (*0.76s*) compared to mask processing and measuring time (*0.2s*). To cut the overall compute time, it is possible to reduce image size while keeping the aspect ratio, simultaneously reducing the area of segmentation. However, the drawback of this approach is measuring accuracy reduction. Another straightforward thing to increase the model's performance can be calculation precision reduction by neural network quantization from mixed precision (float32) to 16 bits (float16). This way, the model inference can be accelerated in recent GPUs or dedicated hardware.

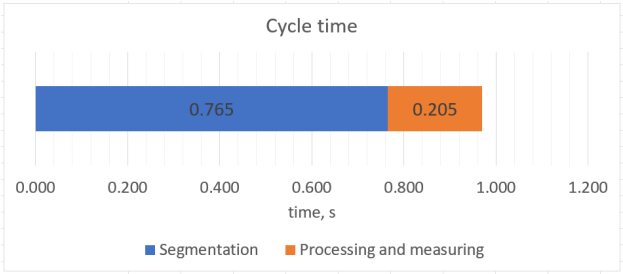

Figure 14. Cycle time diagram

VII. Conclusion

In general, the proposed computer vision measuring technique may be used for any precision demanding applications to estimate geometrical parameters of rectangular objects. Proper segmentation helps to accurately distinguish objects with random texture variance and a non-static background. Moreover, patch manner segmentation can successfully deal with high-resolution images. As for measuring itself, estimating midpoints and fitting a midline based on them seems to be a quite reliable approach even under the object's non-alignment factor.

References

[1] T. Czimmermann *et al.*, "Visual-based defect detection and classification approaches for industrial applications—A SURVEY," *Sensors (Switzerland)*, vol. 20, no. 5, pp. 1–25, 2020, doi: 10.3390/s20051459.

[2] R. Augustauskas, A. Lipnickas, and T. Surgailis, "Segmentation of Drilled Holes in Texture Wooden Furniture Panels Using Deep Neural Network," *Sensors*, vol. 21, no. 11, p. 3633, May 2021, doi: 10.3390/s21113633.

[3] C. Gonzalez-Val, A. Pallas, V. Panadeiro, and A. Rodriguez, "A convolutional approach to quality monitoring for laser manufacturing," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 789–795, Mar. 2020, doi: 10.1007/s10845-019-01495-8.

[4] G. Roberts, S. Y. Haile, R. Sainju, D. J. Edwards, B. Hutchinson, and Y. Zhu, "Deep Learning for Semantic Segmentation of Defects in Advanced STEM Images of Steels," *Scientific Reports*, vol. 9, no. 1, pp. 1–12, 2019, doi: 10.1038/s41598-019-49105-0.

[5] Y. Yuan *et al.*, "Crack Length Measurement Using Convolutional Neural Networks and Image Processing," *Sensors*, vol. 21, no. 17, p. 5894, Sep. 2021, doi: 10.3390/s21175894.

[6] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, doi: arXiv:1704.04861.

[7] M. N. Alhasanat, M. H. Alsafasfeh, A. E. Alhasanat, and S. G. Althunibat, "RetinaNet-Based Approach for Object Detection and Distance Estimation in an Image," *International Journal on Communications Antenna and Propagation (IRECAP)*, vol. 11, no. 1, p. 19, Feb. 2021, doi: 10.15866/irecap.v11i1.19341.

[8] S.-K. Jung, "AniLength: GUI-based automatic worm length measurement software using image processing and deep neural network," *SoftwareX*, vol. 15, p. 100795, Jul. 2021, doi: 10.1016/j.softx.2021.100795.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[10] L. Zhang, J. Wang, and Q. Duan, "Estimation for fish mass using image analysis and neural network," *Computers and Electronics in Agriculture*, vol. 173, p. 105439, Jun. 2020, doi: 10.1016/j.compag.2020.105439.

[11] J. Mei, J.-N. Hwang, S. Romain, C. Rose, B. Moore, and K. Magrane, "Absolute 3d Pose Estimation and Length Measurement of Severely Deformed Fish from Monocular Videos in Longline Fishing," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 2175–2179. doi: 10.1109/ICASSP39728.2021.9414803.

[12] T. Dhikhi, A. N. Suhas, G. R. Reddy, and K. C. Vardhan, "Measuring size of an object using computer vision," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6 Special Issue 4, pp. 424–426, Apr. 2019, doi: 10.35940/ijitee.F1086.0486S419.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[14] R. Decelle and E. Jalilian, "Neural Networks for Cross-Section Segmentation in Raw Images of Log Ends," in *4th International Conference on Image Processing, Applications and Systems, IPAS 2020*, Dec. 2020, pp. 131–137. doi: 10.1109/IPAS50080.2020.9334960.