A convolutional approach to quality monitoring for laser welding

ECE697 Summer2021 Finalized Project Proposal

Yeongeun Kim

Introduction and background

In recent years, the market for electric vehicles has grown up at impressive rates. Accordingly, the demand for high-quality electric motors also continued to rise. To improve the efficiency of the motor, the electric vehicle manufacturers had to reduce the copper loss, which accounts for the biggest loss of the motor. Therefore, the hairpin winding techniques replaced the existing winding techniques, which were round wire winding. However, the hairpin winding leads to a high amount of contact points, and more careful concentration was required. Therefore, laser welding techniques were introduced, and auto inspection for the quality of laser welding has been required.

The present laser welding process consists of two steps; pre-welding and post-welding. Before welding, the detector should find each surface of hairpins apart from the background. Also, it should find the center of the surface, which would be the target point for the laser beam. After welding, the inspector classifies all the classes of welding; correctly welded and incorrectly welded states. The general features of incorrectly welding are over-welded, insufficient-welded hairpins. Based on the features from quality deviations, it should filter the incorrectly welded hairpins from the correctly welded ones.

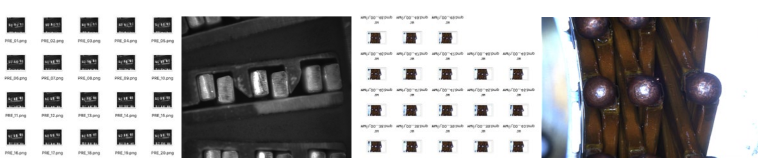
Problem statement

There has been a difference between a rule-based algorithm and a machine learning algorithm in history. For the rule-based algorithm, they do not require a massive training corpus but cannot learn itself. In a while, the machine learning algorithm can learn itself, and with a dataset, fast development is possible.

In the case of laser welding, a rule-based programmed inspection for laser-welding will be able to recognize a state as correctly-welded. There can be different aspects of the state that help it identify it as correctly-welded; it can be the size of the cooper. However, sometimes, the same image can very well be considered as incorrectly-welded state. This project proposes a stable quality monitoring system for the laser welding process using the appropriate machine learning technique. That is, the variation of the conditional environment should be always considered.

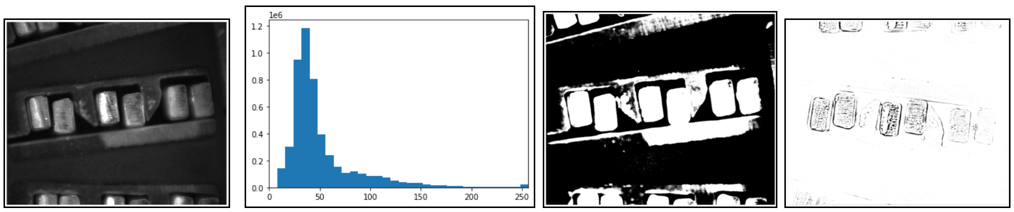
Data analysis

A laser welding data file has two data files, which are pre-welding and post-welding files. The FLIR camera of the laser facility recorded the pre-welding images. The original data file has 20 grayscale images. Each image shows 3 surfaces of hairpins. The D902 camera of an interval inspector recorded the post-welding images. The original data file has 18 RGB images. Each image shows 3 post-welded hairpins.



< Left. Pre_welding_Right. Post_welding>

For the pre-welding process, the surfaces of the hairpins should be detected. Following this, the center of each surface should be computed.
First of all, the grayscale image should check whether there is any high-frequency noise or not. This makes the contour detection process more accurate. Following this, the image should be binarized. Showing the histogram of the image, the distribution of intensity can be checked. Also, the hard thresholding or soft thresholding makes the image extremely binarized. Finally, edge detection can enhance the contour of the object. Therefore, the surfaces of the hairpins can be detected. With this result of pre-processing, the center of the object also can be calculated. The concept of image moments would be used to find the centroid of the object.



<Left to Right. Denoised image, Histogram, Hard threshold image, Soft threshold image >

For the post-welding process, the images should be classified as the quality of laser welding. Then, the model would be tested with a test dataset.
First of all, the images of the original file should be accumulated into a dataset. Therefore, after loading original images with the assigned path, they should be saved as a type of array. Arbitrarily, the train and test split rate is 75:25. In addition, for the baseline model which is a supervised model, the labels of each result are attached. Based on the size of the cooper, the classes could be classified. Finally, the train and test dataset with feature and label would be completed.
After this data generation, the data should be normalized by dividing by 255 and reshaping to the same shape. Also, the mean and standard deviation of each image would be controlled. Then, the whitening would reduce the noise part of the image.
After the data normalization, the shifting, rotating, and flipping processes for data augmentation can be applied. With this result of pre-processing, the baseline model which is a multi-layered CNN model would use the train dataset and test with the validation dataset.
The structure of the CNN model has three convolutional layers, which are followed by the MaxPool layer. At the last layer, the dropout layer would be added. Using Adam optimizer, the model would be trained with a learning rate of 0.000001. Because the data size is small, the epoch should be as big as it can. Finally, the accuracy and loss would be checked.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	896
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_1 (Conv2D)	(None, 112, 112, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 56, 56, 32)	0
conv2d_2 (Conv2D)	(None, 56, 56, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 28, 28, 64)	0
dropout (Dropout)	(None, 28, 28, 64)	0
flatten (Flatten)	(None, 50176)	0
dense (Dense)	(None, 128)	6422656
dense_1 (Dense)	(None, 2)	258

Total params: 6,451,554
Trainable params: 6,451,554
Non-trainable params: 0

	precision	recall	f1-score	support
normal (Class 0)	1.00	0.33	0.50	3
abnormal (Class 1)	0.60	1.00	0.75	3
accuracy			0.67	6
macro avg	0.80	0.67	0.62	6
weighted avg	0.80	0.67	0.62	6

<Left to Right. CNN structure with parameters, # of correctly & incorrectly welding, classification report>

Tools for project

The initial programming tool was PyCharm Community Edition 2021.1.1 x64. The libraries for the pre-welding process would be OpenCV, scikit-image, numpy, and matplotlib. The OpenCV allowed opening the image and process with it. Especially, the scikit-image had lots of tools for image processing. Image segmentation, feature detection, and filtering were executed by the scikit-image library. The libraries for the post-welding process were not only cv2, os, matplotlib, and seaborn, but also TensorFlow and scikit-learn. As you know, the keras is a deep learning support tool in Python, so by using this open-source software, the CNN model could be structured. However, the programming with PyCharm was time-consuming, so GoogleColaboratory was used. In Colab, they were only 30 seconds taken.

Feasibility test

Based on the above scenario, the output of the image could be achieved as the following pictures. The representative output figures of the pre-welding model are denoised, hard-thresholded, soft-thresholded image, and histogram. Because the main objective of this pre-welding was finding the centroid of the surface of the hairpin, the following step should be completed. Therefore, with this image data and scikit-learn tool, they would be detected. Also, with the transition of the external environment such as light and moving camera, the final model should be trained.

The output figures of the post-welding model are the classification_report and the number of parameters used in the CNN model. The highest accuracy with this model was 0.83 and the lowest accuracy with this model was 0.5 in validation dataset. However, based on more clarified proof, such as how far from the centroid, the labels should be rearranged. Also, the base-line model should be upgraded with transition of parameters.

Reference

[1] Mayr A, Lutz B, Weigelt M, Glabel T, Kibkalt D, Masuch M et al. Evaluation of Machine Learning for Quality Monitoring of Laser Welding Using the Example of the Contacting of Hairpin Windings. In : 8th Unt. Elect. Drives Prod. Conf. (EDPC). IEEE; 2018, p 1-7.

[2] J. Vater, P. Schamberger, A. Knoll, and D. Winkle, "Fault Classification and Correction based on Convolutional Neural Networks exemplified by laser welding of hairpin windings," 2019 9th International Electric Drives Production Conference (EDPC), 2019, pp. 1-8.

[3] Image Classification in Python with Keras | Image Classification (analyticsvidhya.com)

[4] https://www.pyimagesearch.com/2016/02/01/opencv-center-of-contour/

[5] Image Segmentation using Python's scikit-image module. | by Parul Pandey | Towards Data Science

[6] OpenCV 3 Image Noise Reduction : Non-local Means Denoising Algorothm - 2020 (bogotobogo.com)