

Non-Contact Extensometer Deformation Detection via Deep Learning and Edge Feature Analysis

Qinghua Xu^a, Xiaodong Wang^{a,1}, Fei Yan^a, and Zhiqiang Zeng^a

^a*College of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China*

Abstract. As one of the basic tests in the mechanics of industrial metal materials, the tensile test is widely used to assess the properties of these materials. By analyzing the data obtained from tensile tests, we can determine the metal materials' tensile strength, elongation, yield strength, et. al. The extensometer is a common instrument in tensile tests and is used to measure the deformation between two points of samples. Traditional extensometers usually require the manual setting of the sensor, which results in poor adjustability, inaccuracy, and incompatibility with specific experimental environments. To address these issues, we design a novel deformation detection framework for non-contact visual extensometer. In this framework, we detect horizontal deformation of materials by extracting and filtering edge features. Besides, a deep learning model is trained to detect vertical deformation between two points of the metal products. We conduct several tensile tests on a non-contact extensometer with our proposed framework. The test results prove that our framework is effective and stable.

Keywords. Deformation detection, non-contact extensometer, edge feature, deep learning.

1. Introduction

The physical life of mankind is filled with various industrial materials, such as metallic and non-metallic materials, composite materials, et al. These materials are the physical basis for human survival and development. To ensure that the materials people use in their lives are reliable and safe. It is essential to test materials and find out their properties [9]. The tensile test [5][6] is one of the most common and important methods for testing the mechanical properties of materials. Through the tensile test, we can know the basic mechanical properties of materials such as flexural strength, elongation strength, tensile strength, section shrinkage, et al. These data are very important for material selection [8], product design, product quality control, and safety assessment. How to achieve efficient and intelligent tests for large numbers of products in industrial applications [18] is still a challenge.

The extensometer [1][2] is an instrument for measuring the deformation between two points of samples and is also a commonly-used tool in the tensile test. Typically, extensometers can be classified into two types, namely contact extensometer and non-contact extensometer. Traditional extensometers usually contain three parts: sensor,

¹ Corresponding Author: Xiaodong Wang, E-mail: xdwangsj@xmut.edu.cn.

amplifier, and recorder. The sensors are connected to the product needed to be tested, they convert the deformation of the product into electrical or acoustic signals which are amplified by the amplifier. Finally, the recorder saves or displays these signals. However, this contact extensometer requires manual setting and unsetting of sensors, which is not only time-consuming but also leads to inaccurate results. Furthermore, this type of extensometer has a fixed calibration distance and measuring range, which may result in poor material compatibility. Most contact extensometers cannot deal with the problem of breaking up products during tests.

To improve the shortcomings of the contact extensometer, the non-contact visual extensometer and its complementary measuring solution were invented. For non-contact visual extensometer, the most important part is to use cameras to capture image data and process these data by advanced machine vision techniques. This extensometer can detect vertical deformation and horizontal deformation simultaneously without manual operation. Its measuring range can be adjusted by the different camera shots with different focal lengths, which makes it adaptable to most sizes of products. This provides the possibility for completely automated tensile tests. However, this extensometer cannot work without the support of deformation detection algorithms. In this paper, we design a novel deformation detection framework for the non-contact visual extensometer. We focus on the deformation of the target product in the vertical direction and the horizontal direction. By detecting edge features to capture the change in the boundaries on either side of the product, horizontal deformation can be further calculated. We design marker points to support detecting vertical deformation. A deep learning-based detecting model is used to detect the marker points' positions in each frame. The vertical deformation of the product can be further obtained by calculating the change in distance between the two marker points. The proposed framework works with a non-contact visual extensometer in tensile tests. The test results proved the effectiveness and stability of our framework.

2. Related Work

2.1. Noise Filtering and Edge Feature Detection

In the tensile test with non-contact extensometers, the deformation process of the sample is started with capturing images by camera. Normally, these original images may contain noises that are not present in real products. These noises may be produced by electric circuits or sensors, which are almost inevitable. They will cause great disruptions to for correct target information extraction. To optimize the original image, we usually conduct a filtering operation at first. The filter-based method is an efficient and convenient way to eliminate noise. This method processes image pixels directly and filters target irrelevant information. Different filters [7][10] have distinctive effects on images with various noise distributions.

In image analysis, the target that needs to be analyzed is usually not the whole image but some local regions or objects within the image. The first task in many requirements is to identify the target object in the image. Although this may be easy for the human, it is not simple work for machines. Edge feature detection [17] is one of the most important techniques for identifying target contours and extracting the target-background boundary. Image edges are sets of pixels whose properties occur regionally with significant changes. Those properties include continuity of grey gradient direction and continuity of depth.

By extracting these edge features, we can simplify image information and obtain the contour or texture of objects [14], and provide solutions for image segmentation [11].

2.2. Target Detection Based on Deep Learning Networks

Target detection is a challenging problem in the field of computer vision. For traditional machine vision algorithms, it is difficult to locate target objects in images quickly and correctly alone. In recent years, advanced deep learning networks provide solutions for this challenge. As the pioneer of applying deep learning to target detection, R-CNN [4] is well known for its high accuracy, but the two-stage network structure limits its detection speed. Later, the YOLO [12] network was proposed, which significantly improved the speed of target detection while ensuring accuracy. After that, variants derived from the Yolo network have been proposed, such as [13]. Furthermore, these deep learning networks are widely cooperated with traditional machine vision methods to achieve better results in co-segmentation [15] and co-identification [16].

3. Our Framework

In our work, the treated objects are the products obtained by shaping the raw metal materials. The extensometer will stretch the product with a vertical upward force and the product will deform or even break. The main function of our framework is to detect deformation of the product in the vertical direction and horizontal direction. For vertical deformation, we set two marker points on the product at the initial state. By detecting the position of marker points in each frame, we can calculate the vertical deformation of the product. For horizontal deformation, we use the location of the detected marker points to select a suitable sub-region. After that, we detect and extract the edge features of this sub-region, then filter out the boundaries on either side of the product, calculating the width of the product based on these boundaries. The horizontal deformation can be obtained by comparing the width of the product in each frame.

3.1. Marker Points Detection

Many techniques can achieve target detection and tracking. However, most of them only yield rough results, and rely on various features on the object's surface. These surface features can be specific textures, colors, or patterns. For some smooth metallic materials with unclear surface features, it is difficult to obtain sufficient features to support the target detection. To solve this challenge, we create marker points to support target detection. The marker points can be affixed to the product's surface (shown in Fig. 1, which allows it to be detected as a feature on the product surface. There are two types of marker points, one is designed to be a circle containing three rings and the other is a circle containing two rings. These designs are to facilitate distinguishing marker points from various products' surfaces and to improve detection accuracy. We can get the approximate position of the product by detecting the position of these marker points.

In the actual measurement, we set two marker points on the product's surface, the upper marker point and the lower marker point. Then we divide a sub-region according to the obtained marker points' coordinates. Following operations are conducted on this sub-region, which effectively avoids interference from other objects in the image and reduces computation time. Moreover, we can use the change in distance between the

upper marker point and lower marker point to calculate the deformation of this sample in the vertical direction.

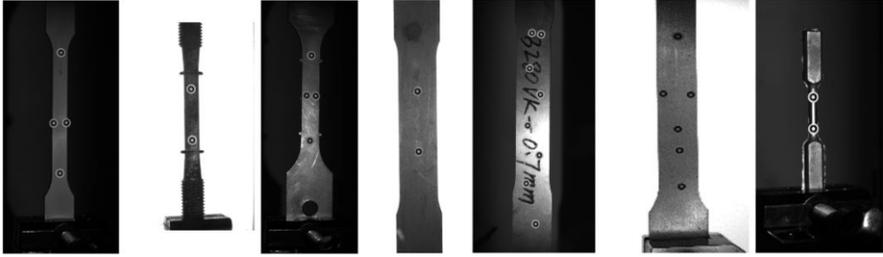


Figure 1. Some products with marker points in our training datasets.

We use the YOLOX [3] network as the backbone network for the marker points detection. The returned results are the coordinates of the innermost circle center. YOLOX is a light detection network that derives from the YOLO network. The training dataset we use is collected from various metal products. Marker points are affixed to the surface of these products with random positions and random amounts. The same product will be captured from several angles as training images in a stable environment.

3.2. Width Measurement Based on Edge Feature Detection

Intuitively, the horizontal deformation of the product is mainly reflected in the change of the product's width. We can reveal the horizontal deformation of the product by calculating its width in each image. From numerous tensile tests, we found that most products showed the tendency to shrink horizontally at the breaking point. This result in uneven widths at different locations when the product is stretched. However, calculating the width at all locations is complex and meaningless. We use the mean value of the k minimum distances to represent the mean width of the product, and use the minimum of the k minimum distances to represent the minimum width of the product. These k minimum distances are the Euclidean distances of the k sets of points on the boundaries on either side of the product. The processing steps for a single image in the width measurement module are shown in Fig. 2 and the details can be summarized as follows:

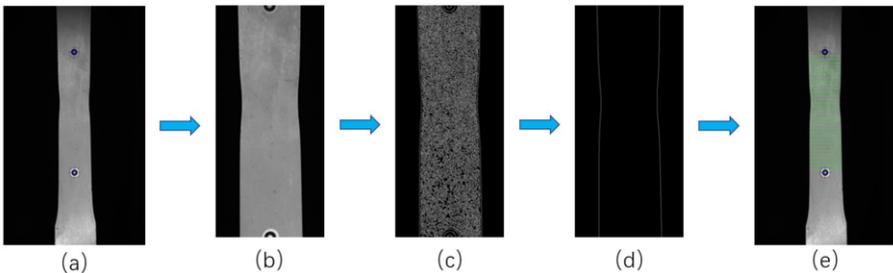


Figure 2. The results of each step for the width measurement. (a) is the original image, (b) is the filtered sub-region, (c) is the extracted edge features from the product, (d) is the selected boundaries, and (e) is the obtained lines (which are marked in green) representing the product width.

- Pre-processing

As shown in Fig. 2(a), we use the coordinates of the detected marker points to intercept a sub-region. The intercepted portion is set to a standard rectangle. We

assume that the coordinates of the upper marker point and the lower marker point are $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$, respectively. Let w be the width of the original image. The left vertex p_e , width w_e , and height h_e of the intercepted rectangle can be denoted as

$$p_e(x_e, y_e) = p_e\left(\frac{x_1+x_2}{2} - \alpha w, y_1\right),$$

$$w_e = 2\alpha w, \quad (1)$$

$$h_e = y_2 - y_1,$$

where α is a parameter which is adjusted according to the size of the product in the image. This sub-region from the original image also contains noise produced by the external environment or the camera. We use the bilateral filter to eliminate the noise, which minimizes the corruption of the original edge information of the sample. The results of this step are shown in Fig. 2(b).

- Two-side Border Recognition

The k sets of points for calculating the product's width are located on the boundaries of either side of the product. Next, we need to identify these two boundaries in the sub-region. As a reminder, we do not need to consider the top and bottom boundaries. This is because the marker points will not be set at the top and bottom of the product. Thus, the top and bottom boundaries of the product will not appear in the sub-region. Since environmental factors (such as illumination and airflow) were preliminarily calibrated at the proper level, we can ensure a significant difference in gray-scale between the sample and the background. In essence, the boundaries on either side that we expect to obtain are part of the edge features. To obtain these edge features, we use the Sobel kernel to calculate the edge gradient and gradient direction for each pixel. Assuming that $G(m, n)$ is the grey gradient of the pixel at m th row and n th column of the input image, $\theta(m, n)$ is its gradient direction, they can be denoted as

$$G(m, n) = \sqrt{g_x(m, n)^2 + g_y(m, n)^2}, \quad (2)$$

$$\theta(m, n) = \arctan \frac{g_y(m, n)}{g_x(m, n)}, \quad (3)$$

where $g_x(m, n)$ and $g_y(m, n)$ denote the horizontal gradient and vertical gradient of the pixel at m th row and n th column of the input image, respectively. After obtaining all pixels' grey gradients and gradient directions, we filter out the pixels that are not edges and make the width of the edge equal to one pixel. For each pixel, we check if its gradient value is a local maximum in its gradient direction. If it is, we set the grey value to 255, otherwise, we set the grey value to zero. Then, we can obtain a set of edge features containing boundaries on either side of the product, which is shown in Fig. 2(c). It is not difficult to find out that the boundaries on either side of the product run through the sub-region along the Y-axis. The width of the extracted edge is one pixel, which means that

the number of pixels projected on the Y-axis by a single boundary is almost equal to the length of the sub-region on the Y-axis. We can easily extract the boundaries according to the height of the sub-region, the extracted boundaries are shown in Fig. 2(d).

- **Width Calculation**

As we mentioned, it is very difficult and time-consuming to calculate the width of the product at all positions. We pick k groups of points on the left and right boundaries, each group containing two corresponding points from the left and right boundaries respectively. Then, we calculate the distance between these k groups of points to represent the minimum width and average width of the product in the sub-region. Assuming that the two points in each group are $p_l(x_l, y_l)$ and $p_r(x_r, y_r)$, the distance L_k between them can be denoted as

$$L_k = \sqrt{(y_r - y_l)^2 + (x_r - x_l)^2}, s, t, k > 0, \text{ and } k \in \mathcal{Z}, \quad (4)$$

where \mathcal{Z} denotes the set of integers. The distance between each group of points should be the shortest distance from them to each other's boundary. We can divide the boundaries into $k + 1$ equal segments according to their projection in the vertical direction, which gives us k tangent points. However, this method only is satisfied if the product is placed vertically, the distance from one point to another point's boundary in each group is the shortest. In practice operation, it is difficult to place the product vertically every time. To avoid this problem, we choose a boundary as a fixed boundary and obtain its k tangent points. The other boundary is treated as the scanned boundary. For one of the k points on the fixed edge, we search for the point with the shortest distance from it to the scanned edge. After obtaining the k groups of points and the Euclidean distances between them. We will remove those bad groups, whose points have relatively large variances in the Y-axis coordinates. In particular, we calculate the mean variance in the Y-axis coordinates of the points for each group. Those groups whose variance in the Y-axis coordinates is larger than two times the mean variance will be defined as bad groups. For the demonstration, we connect two points in each of the remaining groups with green lines in Fig. 2(e). The mean width of the product is calculated by the mean Euclidean distance of the remaining groups. The minimum width is the minimum Euclidean distance of these remaining groups.

3.3. Calculating Horizontal and Vertical Deformation

Previously, we decomposed the vertical deformation and horizontal deformation into the change in the distance between marker points and the change in the width of the product in each frame. The results of the first frame are taken as the original reference, L_o is the original marker point distance, W_o is the original minimum width. Except for the original reference, the marker point distance and minimum width of the i th frame are denoted as L_i and W_i . Compared with the original reference, the vertical deformation Δ_L and the horizontal deformation Δ_W of i th frame can be written as

$$\Delta_L = |L_o - L_i|, \Delta_W = |W_o - W_i|, s, t, i > 0, \text{ and } i \in \mathcal{Z}, \quad (5)$$

where $i \in \mathcal{Z}$ denotes integers. Even if the product is stationary, the L_i and W_i calculated from each frame are not the same. This is because of the pixel grayscale fluctuations caused by illumination. When the product is stationary, Δ_L and Δ_W will fluctuate within a small range. If the product is being stretched, Δ_L and Δ_W tend to increase until it is broken. According to the changes of Δ_L and Δ_W , we can further analyze the properties of the product such as elastic limit, elongation, tensile strength, et al., can also be assessed.

4. Experiments

To verify the stability and accuracy of our framework, we simulated a real industrial environment and conducted experiments. We select a dumbbell-shaped metal product as the experimental object. The size of the input images in the experiments is 2048x1224 including training datasets and test datasets. The deep learning model used in the experiments was trained by a machine with 64G RAM and a 3090ti graphics card on the platform PaddlePaddle. We set the training parameters, i.e., learning rate to 0.008, batchsize to 32, epochs to 400, set the parameters for width measurement, i.e., $k=32, \alpha=0.15$.

4.1. Datasets collection

We fix the dumbbell-shaped metal product on the non-contact extensometer and use a high-speed camera to capture image data. Two static product datasets and one stretched product dataset were collected. For the static product datasets, the product is not stretched. We collect the one (we call it "static dataset one") of these static datasets (Fig. 3(a)) in a black background environment and the other one (we call it "static dataset two") (Fig. 3(b)) in a white background environment, each with 2,000 images. For the stretched dataset (Fig. 3(c)), the product is stretched in a black background until it breaks, with 88 images.

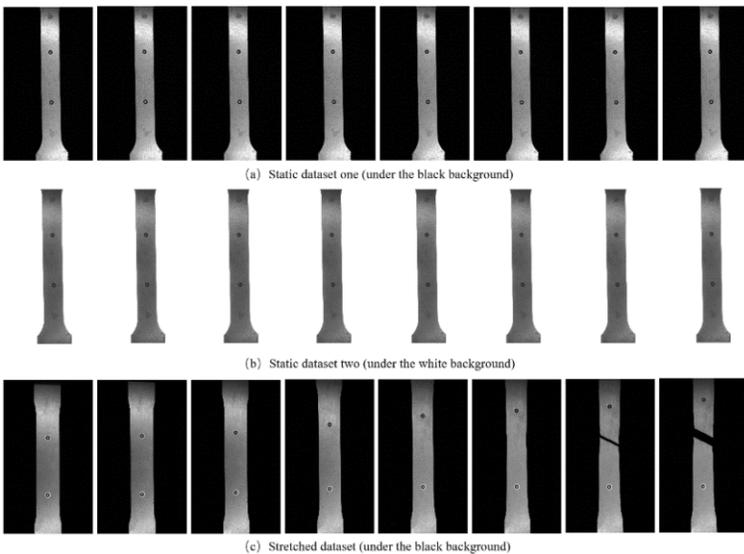


Figure 3. Collected two static datasets and one stretched dataset.

4.2. Stability Experiment

In real-world industrial applications, the environmental factors and the camera itself will cause subtle differences in each original frame. Gray-scale fluctuations and randomly generated noise are inevitable during the continuous original image capture process. To test the stability of our framework with these subtle changes, we conducted stability experiments on two static datasets. The first image of each dataset is treated as the reference. We calculate the vertical deformation and horizontal deformation on the rest images.

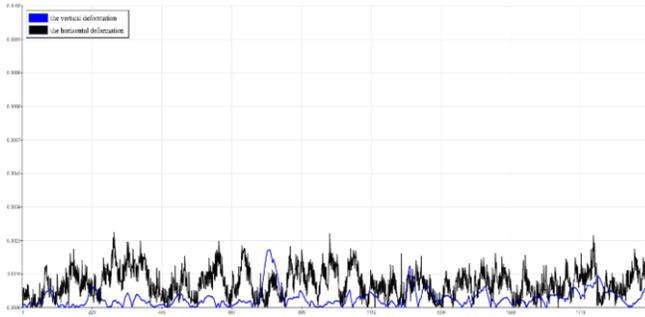


Figure 4. The results of horizontal deformation and vertical deformation in millimeters on static dataset one. The X-axis is the number of frames and the Y-axis is the value of deformation.

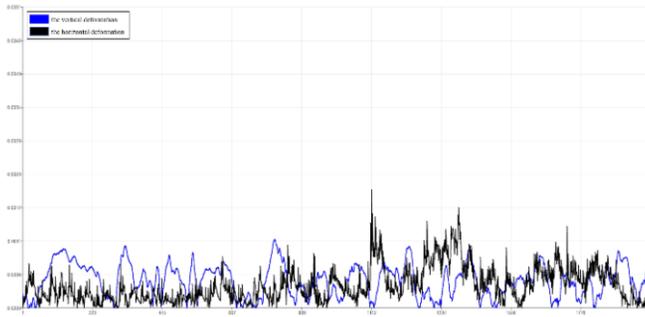


Figure 5. The results of horizontal deformation and vertical deformation in millimeters on static dataset two. The X-axis is the number of frames and the Y-axis is the value of deformation.

Fig. 4 and Fig. 5 show the results for static dataset one and static dataset two (under the light background) respectively. The black curve represents the horizontal deformation and the blue curve represents the vertical deformation of the product in each image except the first image. All results are based on the comparison with the reference image. In Fig. 4, the black curve fluctuates more sharply than the blue curve. In Fig. 5, the fluctuations of the two curves are relatively close to each other, but the black curve fluctuates greater in the short term. This indicates that our framework performs more stably under light backgrounds. It is also evident from the maximum deformation values, which are over 0.0023 millimeters in Fig. 5 but below 0.0023 millimeters in Fig. 4. The reason is probably that the Gray-scale fluctuates greater in the black backgrounds, which leads to unstable edge recognition in the width measurements stage.

4.3. Effectiveness Experiment

The main purpose of our proposed framework is to measure the deformation when the product is being stretched. To check the effectiveness of our framework, the tensile test is essential. We conduct the tensile test on the stretched dataset. In this experiment, we also take the results of the first image as the reference and calculate the vertical deformation and horizontal deformation of the products in the remaining images. We select some processing results of frames and exhibit them in Fig. 6. The vertical deformation and horizontal deformation of the product are shown in Fig. 7.

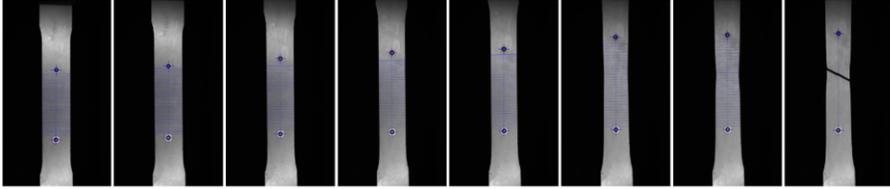


Figure 6. Processing results of some frames by our framework.

From Fig. 6, we can clearly see that the product is stretched until it breaks. Those blue solid lines in the horizontal direction represent the 32 groups of the product width between the marked points. The bold blue line in each image represents the smallest width. If all widths are equal then the first one will be marked as bold blue. The blue dotted lines in the horizontal direction represent the distance between the marked points. Following the product being stretched, the curves in Fig. 7 show an upward trend, and the bold minimum width is also moving to the breaking area. The horizontal deformation detection is terminated if the sample is broken, which is why the black curve disappears halfway through. The blue curve rises for a while and then stops changing, this is because the extensometer still works after the product breaks. We measure the minimum width and the distance between the marker points on the broken product using a ruler with 0.01-millimeter precision. The final deformation in Fig. 7 is compared with the actual measurement results. The comparison results confirm that our framework can meet the precision requirements of at least 0.01-millimeter.

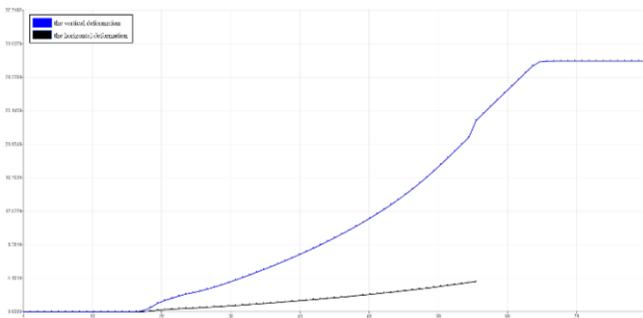


Figure 7. The results of horizontal deformation and vertical deformation in millimeters on the stretched dataset. The X-axis is the number of frames and the Y-axis is the value of deformation.

5. Conclusion

To overcome the shortcomings of traditional contact extensometers. We design a deformation detection framework for the non-contact visual extensometer, and design

marker points to support detecting deformation. The proposed framework utilizes edge features and marker points to measure the horizontal deformation and vertical deformation of industrial metal materials. A target detection model based on deep learning is trained to detect the locations of marker points. We collected the datasets in a simulated real-world industrial environment and conducted the tensile tests. The test results on all datasets proved that our framework is stable and effective.

Acknowledgement

This paper was supported by National Natural Science Foundation of China (Grant Nos. 61871464, U1805264), National Natural Science Foundation of Fujian Province (Grant Nos. 2021J011186, 2020J01266), the "Climbing" Program of XMUT (Grant No. XPDKT20031), Scientific Research Fund of Fujian Provincial Education Department (Grant No. JAT200486), the University Industry Research Fund of Xiamen (Grant No. 2022CXY0416).

References

- [1] Corominas, J., et al., Measurement of landslide displacements using a wire extensometer, *Engineering Geology* **55.3**(2000):149-166.
- [2] Gamez-Perez, J., et al, Use of extensometers on essential work of fracture (EWF) tests, *Polymer Testing* **27.4**(2008):491-497.
- [3] Ge, Z., et al, YOLOX: Exceeding YOLO Series in 2021, *ArXiv abs* **2107.08430** (2021).
- [4] Girshick, R., et al, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *IEEE Computer Society*, (2014):580-587.
- [5] Hart, E. W., Theory of the tensile test, *Acta Metallurgica* **15.2**(1967):351-355.
- [6] Huh, H., J. H. Lim, and S. H. Park, High speed tensile test of steel sheets for the stress-strain curve at the intermediate strain rate, *International Journal of Automotive Technology* **10.2**(2009):195-204
- [7] Ito, K., and K. Q. Xiong, Gaussian filters for nonlinear filtering problems, *IEEE Transactions on Automatic Control* **45.5**(2000):910-927.
- [8] Kang, Jk., A method for optimal material selection aided with decision making theory, *Materials & Design* (2000):199-206.
- [9] Rothe, Marti Jill., Assessment of the accuracy of material safety data sheets, *American Journal of Contact Dermatitis* **7.3**(1996):195-195.
- [10] Lukac, R., Adaptive vector median filtering, *Pattern Recognition Letters* **24.12**(2003):1889-1899.
- [11] Malik, J., Belongie, S., Leung, T., et al, Contour and Texture Analysis for Image Segmentation, *International Journal of Computer Vision* (2001):7-27.
- [12] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., You Only Look Once: Unified, Real-Time Object Detection, *2016 IEEE Conference on Computer Vision and Pattern Recognition* (2016):779-788.
- [13] Redmon, J., and A. Farhadi, YOLO9000: Better, Faster, Stronger, *2017 IEEE Conference on Computer Vision and Pattern Recognition* (2017):6517-6525.
- [14] Rosenfeld, A., and M. Thurston, Edge and Curve Detection for Visual Scene Analysis, *IEEE Transactions on Computers* **20.5**(1971):562-569.
- [15] Vicente, S., C. Rother, and V. Kolmogorov, Object cosegmentation, *IEEE Computer Vision and Pattern Recognition* (2011):2217-2224.
- [16] You, Yu-Li and Mostafa Kaveh, A regularization approach to joint blur identification and image restoration, *IEEE transactions on image processing* **5.3**(1996):416-28.
- [17] Yu-he, Li., Summary of image edge detection, *Optical Technique* (2005).
- [18] Yang, Y., Zhuang, Y. & Pan, Y. Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. *Front Inform Technol Electron Eng* **22**(2021):1551–1558.