Imaging Analysis of Steel Rod Shrapnel Using Artificial Neural Networks

Kittiya Poonsilp

Abstract—**The purpose of this study was to analyse the explosive forensic data contained in digital image. The explosive forensic data used in this study was mainly targeted on the cross-section image of steel rod shrapnel. Digital image processing techniques were used to find the areas and types of the shrapnel. The process began with imaging segmentation to find the boundary box of the shrapnel area and then extracted the areas which contained key features and computed the statistical values of those areas, Finally, the statistical values were administered to the Artificial Neural Network to classify the types of shrapnel. The results showed 71% accuracy which was acceptable since each type of cross-section image had a very slightly different and much different to detect by human eyes.**

Index Terms—**Explosive ordnance disposal, neural network, shrapnel, steel rod.**

I. Introduction

According to the reports and statistics of unrest situations in the 3 southern border provinces of Thailand. It indicated that 90% of the situations was related to the bomb. From the discussion with Explosive Ordinance Disposal (EOD) officer who worked in that area, it was found that all bomb situations used shrapnel made from steel rod because it was easy to buy and had more significant power to destroy objects than any other materials. Incendiaries usually use a several tools to cut a steel rod e.g. chainsaws, pliers etc. The surface of the shrapnel which was cut from different tools had different characteristics. The shrapnel which made from different groups of incendiaries was also different and usually contains a unique strain from the tools.

The operation of the EOD was began with collecting the evidences and shrapnel from the scene. Then took a photograph of cross-section of shrapnel and attached with its data in the form of report paper. If there was a numerous number of situations and data, it would be difficult and take a very long time for human to do manually match the current situation with the past situations, or to compare with historical data. It might be required a microscope to find some small unique features sometime. All of these required a huge amount of effort.

From the limitations above, we proposed the solution to use a digital image processing techniques to automatically extract the features and categorise the type of shrapnel from digital image that contained the cross-section shrapnel. The proposed solution had 3 main steps which were locating a cross-section area, feature extraction, and feeding to neural network to recognise the type of shrapnel. The details of each steps were described in detail in the next topic.

The proposed solution could be used as a prototype for software application to automate image analysis of the shrapnel evidences so that the EOD operation could be done much faster with more accuracy.

II. Related Work

There are a lot of researches on image classification and data analysis [1]-[3] but researches and works on applying image analysis techniques in forensic especially shrapnel image were relatively few. One of the them was [4]. This work aimed to improve the detection depths of the GMR sensor based on shrapnel detectors that could help surgeons to trace the location of shrapnel more accurately in a victim's body during a surgery.

Wei [5] designed a large EOD robot that could replace man to reconnoiter, remove and deal with explosives or other dangerous articles in dangerous environment. The robot was constituted by vehicle body, mechanical hand, remote control trolley, vision system, communications system other ancillary equipments. The large EOD Robot was proved by experiments that had strong capacities of obstacle-crossing and article-grasping. It could be used in wild environment e.g. sand, grass and soft soil, etc.

Chun-yao Xu [6] proposed the virtual reality (VR) system using OpenGL and VC++ technologies to simulate the 3D scenes and environment to be used in EOD training and supposed to support the synchronization of real-time interaction in three-dimensional scene among clients. The system also insulates the explosive parameter from software developers, modelers and other non-authorized personnel.

Sui-ping [7] proposed a method to count an amount of steel rods in real-time image with machine vision based on Learning Vector Quantisation Neural Network (LVQNN). The input vectors for the network were acquired from real-time image. During the training, the samples were divided into two groups: positive and negative samples in which the measured object was exist or not exist, respectively.

Zhang [8] worked on scratch defect detection on a wire rod. Scratch defect was caused from various reasons such as poor quality of raw materials or malfunction of rolling process. The defects were very difficult to accurately detect due to the scale-covered background and uneven illumination. They proposed the computer vision technique to detect scratch defect of a wire rod and achieve good performance.

Moreover, there were some more works that proposed an interesting technique to process and analyse an image. For

Manuscript received June 10, 2019; revised May 29, 2020. This work was supported in part by department of computer science, Suan Sunandha Rajabhat university, Bangkok, Thailand.

Kittiya Poonsilp is with the Department of Computer Science, Suan Sunandha Rajabhat University, Bangkok, Thailand (e-mail: kittiya.po@ssru.ac.th).

doi: 10.18178/ijmlc.2020.10.5.994

example, Jensuriyagul [9] focused on classification method of soft-drink crown-cap types by using Back-Propagation Neural Network (BPNN). The results revealed that an accuracy of the classification of four cap types was approximately 95% while the others were rejected.

Kample [10] presented a coin recognition method with rotation invariance. The rotation invariance feature was represented by the absolute value of Fourier coefficients from polar image of coins on circles with different radii. Then coins could be distinguished by feeding those features into a multi-layered BP neural network.

III. Shrapnel Cross-Section Detection

This topic focused on locating the cross-section area from the input image. The process could be described in Fig. 1.

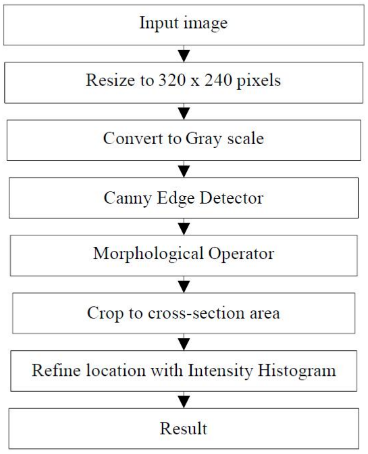

Input image
Resize to 320 x 240 pixels
Convert to Gray scale
Canny Edge Detector
Morphological Operator
Crop to cross-section area
Refine location with Intensity Histogram
Result

Fig. 1. Locating cross-section area process.

Sample image which contained the cross-section of shrapnel was shown in Fig. 2.

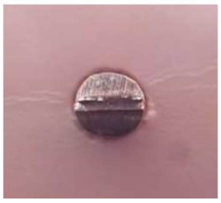


Fig. 2. Sample image.

After receiving image from user. Firstly, resized image to 320 × 240 pixel by using Bicubic Interpolation Technique [11]. After resizing, the search space was significantly reduced and the performance of the subsequent analysis was also improved.

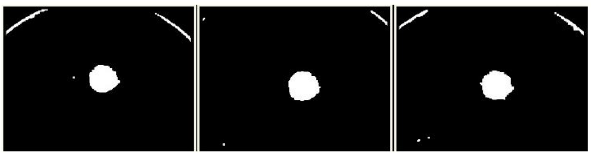


Fig. 3. Image after applied fill holes.

Next, converted the image to grey scale to eliminate color factor that could affect the analysis and then found the edge by using Canny Edge Detector [12]. Detecting the edge was robust for the various exposures. In this step we obtained a binary image, which was further applied to the Morphological Opening algorithm from [13] and [14] to remove some noises.

The image in this step still contained some noises and black holes inside the cross-section area, so we used Fill Holes technique to fill the black holes with white pixels.

After filling the holes, the cross-section area was completely shown in Fig. 3.

From Fig. 3, there were several connected components or blobs in the image. So we selected the most fullness connected component that could be calculated by using (1).

$$\text{Fullness} = \frac{N}{\text{Width} \times \text{Height}} \tag{1}$$

where N is the total number of white pixels in a blob, Width and Height are width and height of blobs, respectively. Since the circle area occured a fullness value more than line and curve, therefore there was a cross-section area as the result.

Once we obtained a blobs of cross-section, the next step was to crop only the cross-section area of the image. Because the current image was a small size image, so we needed to locate and crop the area from the original size image instead to get the most maximum resolution as possible for the next step.

After cropping, the resulted images were shown in Fig. 4.

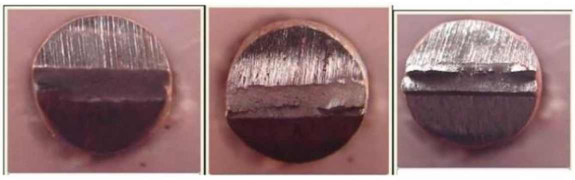


Fig. 4. Image after cropped from the original image.

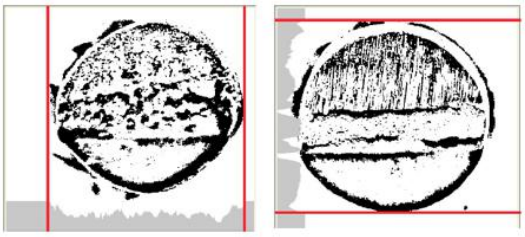


Fig. 5. Image from intensity histogram calculation for both x axis and y axis.

From Fig. 4, the cropped image was not fit to the cross-section area, but still had some space around the cross-section that might be caused by object shadow etc. So the next step was to specify more accurate location for the cross-section.

For more precise locating the cross-section, we used an intensity histogram which required binary image as an input, so it was necessary to convert to binary image by using an Adaptive Threshold. The advantage of adaptive threshold was the threshold value which calculated dynamically according to surrounding pixels. So we obtained good and clear result although there were different exposures or shadows in some parts of an image.

The binary image was then used to calculate intensity histogram for both vertical and horizontal axis which yielded the graph histogram to show the amount of white pixels on each value of x and y axis as shown in Fig. 5.

The Fig. 5 (left) revealed that the horizontal intensity histogram would reach the maximum value if all pixels in the row were white and the value would decrease if the row got more dark pixels (which was likely to the edge of cross-section). Therefore, when determining the histogram from the left-most position, if the intensity value decreased more than the predefined threshold, that position would be regarded as the left edge of cross-section. This techniques would also be applied to the top, right and bottom edge. Finally, there was a boundary box of cross-section as the final result.

The size of result image might be varied depending on size of cross-section area in the original image. Before processing the subsequent steps, the image needed to be firstly resized to 300 × 300 pixels and applied a histogram equalisation for a better contrast. The result was shown in Fig. 6 and Fig. 7.

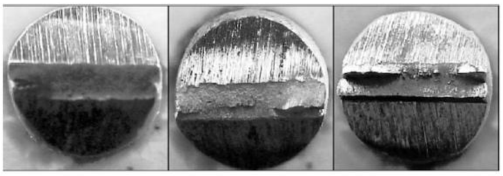


Fig. 6. Image after applied histogram equalisation technique.

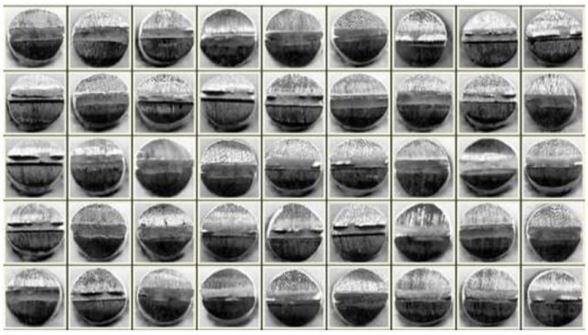


Fig. 7. Image after applied histogram equalisation technique.

IV. Features Extraction

Features extraction consisted of processes as shown in Fig. 8.

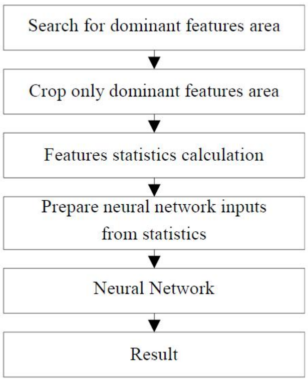

Search for dominant features area
Crop only dominant features area
Features statistics calculation
Prepare neural network inputs from statistics
Neural Network
Result

Fig. 8. Features extraction process.

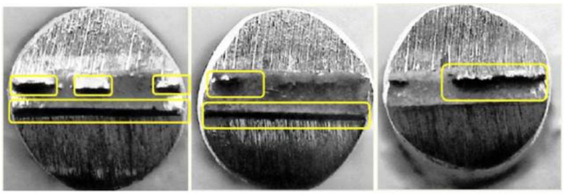


Fig. 9. Crack and shape at the center area of cross-section.

Regarding to the analysis on historical image data, it was found that cross-section normally contained the dominant features at the center area. Center area contained some unique crack or different shape (due to different tools for cutting or different skills) which could be a useful information to categorise the type of cross-section.

From the study of various histogram on cross-section image, it was found that at the dominant features area, histogram significantly changed because dominant area contained more detail, bright and dark pixels (which caused from light reflection and shadow of cracks) than other areas. So we decided to focus on extracting this area.

In order to process an image more robustly, the pre-processing process should be done by two following processes:

1) Reduce grey scale level, from 256 shades of grey to only 5 shades of grey which were dark, dark-grey, grey, light-grey and white by using (2).

$$f(x) = \begin{cases} 0, & 0 \leq x \leq 50 \\ 1, & 50 < x \leq 100 \\ 2, & 100 < x \leq 150 \\ 3, & 150 < x \leq 200 \\ 4, & 200 < x \leq 255 \end{cases} \tag{2}$$

2) Create circular mask, since the cross-section normally is in circular shape, then used circular mask to process only the pixels in the mask, which could be illustrated in Fig. 10.

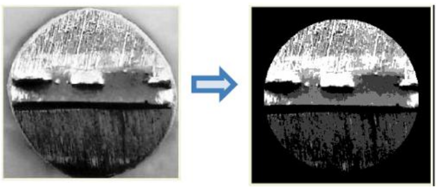


Fig. 10. Image after applied a circular mask.

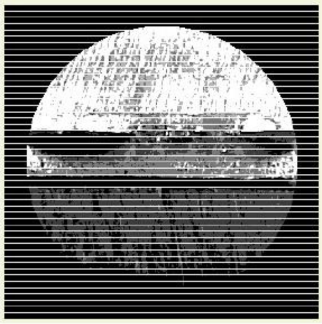


Fig. 11. 50 rows in image.

For the statistics calculation of given cross-section, the object called *Receptive Field* was designed to use. Each receptive field had responsibility to calculate statistics values over the certain part of image.

The whole image was divided into 50 rows horizontally and used 50 receptive fields to calculate the statistics by using 1 receptive field per 1 row.

On each receptive field, it calculated 3 values which were mode of grey level, white pixel ratio and black pixel ratio in the following:

1) Mode of grey level. For example, if the most pixels have a grey level = 3 in a receptive field, the Mode value = 3.

2) White ratio is the amount of white pixel per total pixels in a receptive field.

3) Black ratio is the amount of black pixel per total pixels in a receptive field.

The statistic values were shown in the Fig. 12. The leftmost band was Mode, the center was white ratio, and the rightmost band was black ratio.

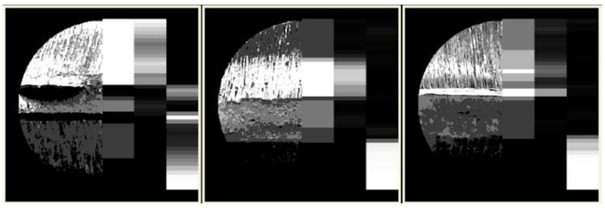


Fig. 12. Histogram of cross-section statistics values.

From Fig. 12, the histogram values significantly changed in the center of the image. The total differential could be calculated by using (3)(4)(5)(6), as

$$diff_{total}(i) = (a \times diff_{mode}(i)) + (b \times diff_{white}(i)) + (c \times diff_{black}(i)) \quad (3)$$

$$diff_{mode}(i) = MAX\begin{cases} |(mode(i) - mode(i-1)| \\ |\, mode(i) - mode(i+1)| \end{cases} \quad (4)$$

$$diff_{white}(i) = MAX\begin{cases} |(white(i) - white(i-1)| \\ |\, white(i) - white(i+1)| \end{cases} \quad (5)$$

$$diff_{black}(i) = MAX\begin{cases} |(black(i) - black(i-1)| \\ |\, black(i) - black(i+1)| \end{cases} \quad (6)$$

where i is the number of receptive field

$mode(i)$ is mode calculated by receptive field i

$white(i)$ is white ratio calculated by receptive field i

$black(i)$ is black ratio calculated by receptive field i

a, b, c are coefficient values of differential, which is specified to 2.0, 1.0 and 1.0, respectively.

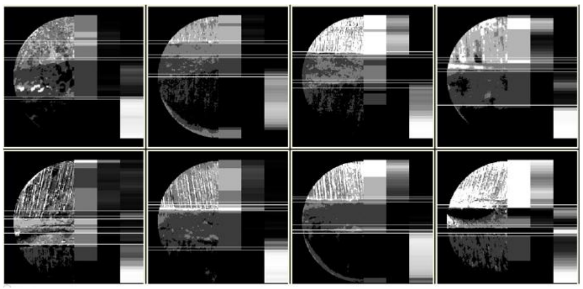


Fig. 13. Position of 6 receptive fields contained maximum diff total value.

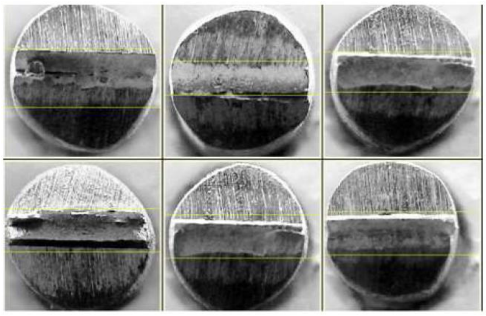


Fig. 14. Boundary of dominant features area.

After getting all of the $diff_{total}$ values, the next step was to find the 6 receptive fields which had a maximum value. The result was shown in the Fig. 13, the white horizontal line showed the position of the 6 receptive fields.

From the 6 receptive fields, the top most and the bottom most receptive fields represented the top and bottom boundary of dominant features area, respectively as illustrated in the Fig. 14.

Once knowing the boundary, then cropped to extract the only dominant features area. The result image might be varied in size, the results after resized to 300 × 100 pixels were shown in Fig. 15.

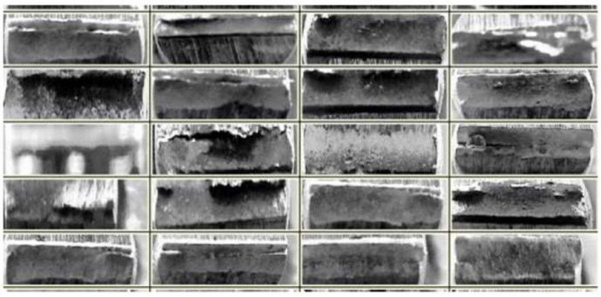


Fig. 15. Dominant features area cropped from the original image.

The third major step was to categorize the type of cross-section using Artificial Neural Network. Since the result was an image having 300 × 100 pixel, if the image was directly feeded to the neural network, the network would obtain 30,000 input nodes and 15,000 hidden nodes, 45,000 nodes in total which were too large number of nodes, causing a very slow and inefficient in training.

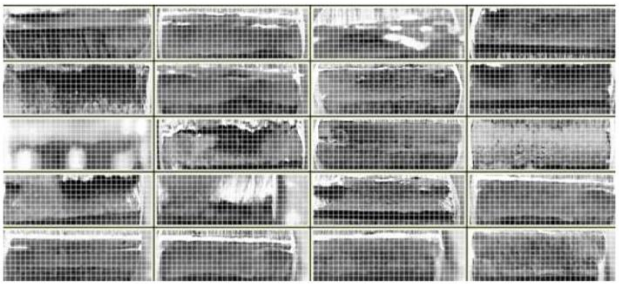


Fig. 16. Horizontal and vertical receptive fields.

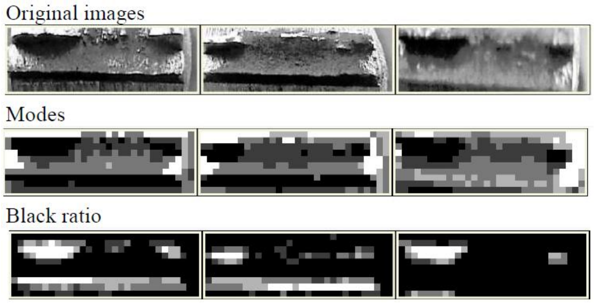


Fig. 17. Image from receptive fields.

In order to make a neural network to be more efficient, we had to reduce an input and keep only important features. Hence, the subsequent pre-processing technique was introduced. The pre-processing also use a receptive fields but in this step, 30 horizontal receptive fields and 10 vertical receptive fields were used. Each receptive fields had a responsible for 10 × 10 pixels as shown in Fig. 16.

After calculating of receptive fields as same as previous steps to get mode, white ratio and black ratio. We used only mode and black ratio which were necessary for neural network to determine important features e.g. crack shape. The result was shown in Fig. 17.

The result of this step was mode and black ratio which were ready for feeding to neural network in the next topic.

V. Cross-Section Classification using Neural Network

The neural network from [15] and [16] proposed in this paper was the feed forward multi layer network containing 3 layers which were input layer, hidden layer and output layer. The input came from 30 × 10 = 300 receptive fields, each of field contained 2 values: mode and black ratio, hence the total number of input nodes would be 300 × 2 = 600 nodes. For the mode value, the range [0-4] was converted to the range [-1, 1]. For the black ratio value, the range was [0, 1]. These values were feeded to input node.

For the hidden layer, the number of nodes was half of input node which was 300 nodes in total. Output layer contained 7 nodes. Each output node represented the features which could be recognised from the input as shown in Table I.

TABLE I: The Meaning of Each Node in Output Layer

Node	Recognized Features	Value	
		Yes	No
1	Plain cross-section, no any detail	1	0
2	Two-tone cross-section, no any detail	1	0
3	Crack at the left	1	0
4	Crack at the center	1	0
5	Crack at the right	1	0
6	Swell indentation	1	0
7	Other	1	0

From the Table I, the recognised features could be illustrated in Fig. 18.

The output values (3, 4, 5, 6) from neural network could be combined together to make a total possibilities $2^4 = 16$ possibilities of crack that could be detected by the system. If include (1) and (2), the system might classify up to 18 different types of cross-section. The neural network structure was illustrated as shown in Fig. 19 where:

Mode #1 is Mode value from Reception Field #1.

Black #1 is Black ratio value from Reception Field #1.

Mode #2 is Mode value from Reception Field #2.

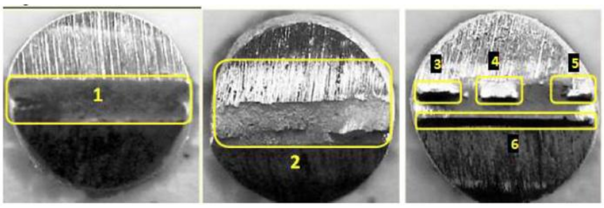


Fig. 18. Features and corresponding output value.

In each node, it calculated summation from all input and used sigmoid function as activation function.

The result from output nodes was interpreted to a type of cross-section. For example, if value from Output #3 and Output #4 were closed to 1 and other output values were closed to 0, it was interpreted that the cross-section contains two cracks which were left and center crack. If Output #5 and Output #6 were closed to 1 and other outputs were closed to 0, it was interpreted that the cross-section was swell and contained only one crack in the right.

For the training part, we used *Back Propagation* as the training method which was the most popular method to train the multi-layer neural network.

After getting the result from the neural network, the Table II was used to interpret the result to the corresponding cross-section type.

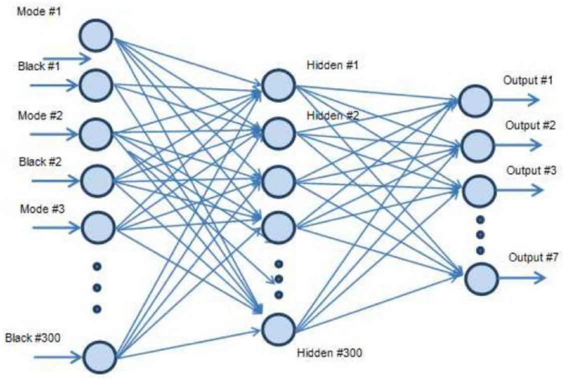


Fig. 19. The structure of the neural network.

TABLE II: Result Translation Table

Output #1	Output #2	Output #3	Output #4	Output #5	Output #6	Output #7	Type
1	0	0	0	0	0	0	A
0	1	0	0	0	0	0	B
0	0	0	0	0	1	0	C
0	0	0	0	1	0	0	D
0	0	0	0	1	1	0	E
0	0	0	1	0	0	0	F
0	0	0	1	0	1	0	G
0	0	0	1	1	0	0	H
0	0	0	1	1	1	0	I
0	0	1	0	0	0	0	J
0	0	1	0	0	1	0	K
0	0	1	0	1	0	0	L
0	0	1	0	1	1	0	M
0	0	1	1	0	0	0	N
0	0	1	1	0	1	0	O
0	0	1	1	1	0	0	P
0	0	1	1	1	1	0	Q
0	0	0	0	0	0	1	R

VI. Result

In this experiment, there were 205 sample images, having a resolution of 2592 × 1944 pixels. The experimental results were shown in the following:

A. Locating a Cross-Section Area

Result from locating a cross-section area was shown in Table III.

TABLE III: Result from Locating Cross-Section Area

	Result		Total
	Correct	Fault	
Images	202	3	205
Percent (%)	98.5	1.5	100

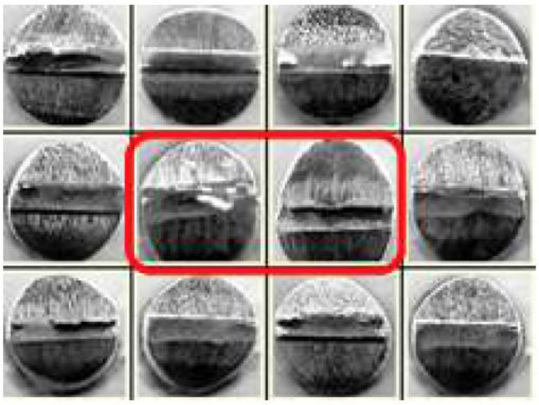


Fig. 20. Incorrect result area.

The result of images was shown in Fig. 20, the red box indicated the incorrect area.

B. Feature Extraction

This section evaluated the boundary box of dominant features area. The boundary box contained the dominant features area at least 80%.

TABLE IV: RESULT FROM DOMINANT FEATURES AREA EXTRACTION

	Result		Total
	Correct	Fault	
Images	179	26	205
Percent (%)	87.3	12.7	100

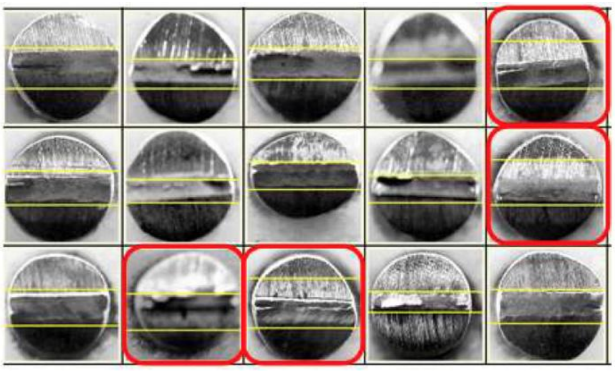


Fig. 21. Incorrect cases from dominant features area extraction process.

The incorrect cases were shown in red box in Fig. 21.

C. Neural Network

All of the sample images were divided into 2 sets which were 100 images for training and 100 images for testing. For the first iteration of training, we used only 20 images to quickly adjust weights for each node and then used the remaining images to refine the values until mean square error was less than the threshold value.

The mean square error from each training iteration was illustrated in Fig. 22.

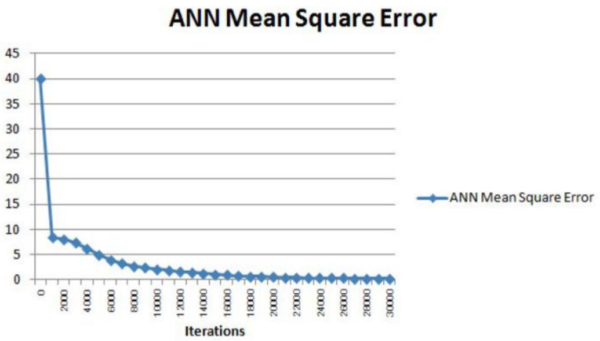


Fig. 22. Mean square error from each training iteration.

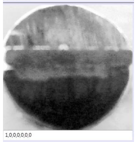
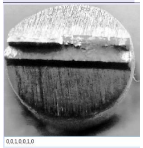
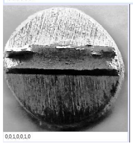
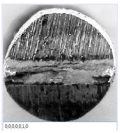
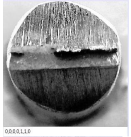
The first 1,000 iterations had a big impact to weight, the weight was quickly adjusted which decreased the mean square error from 40 to 8.4. After the first 1,000 iterations, the error was gradually decrease. After 16,000 iterations, the error was below to 1.0 and stable. Until 30,000 iterations, the error was end at 0.18.

The neural network yielded the classification result at 71%, the correct classification was 71 images from 100 images. Incorrect classification was 29 images as shown in Table V. We analysed the result for both cases in Table VI.

TABLE V: NEURAL NETWORK CLASSIFICATION RESULT

	Result		Total
	Correct	Fault	
Images	71	29	100
Percent (%)	71	29	100

TABLE VI: ANALYZE THE RESULT FOR BOTH CASES

Image	Result from ANN	Result	Analysis
	1,0,0,0,0,0,0 Plain, no crack	Correct	This type yielded most accurate result at 90% due to its simple characteristics.
	1,0,0,0,0,0,0 Plain, no crack	Correct	Can be classified correctly even though the image was blur due to its simple characteristics
	0,0,1,0,0,1,0 Swell with left crack	Correct	The dominant features area was not center but located above the center, anyway the result was correct.
	0,0,1,0,0,1,0 Swell, left crack	Correct	Contain very small crack, anyway the neural network can classify correctly.
	0,0,1,1,1,1,0 Swell, Left, Center and right crack	Correct	The crack in this image seems to be ambiguous. Left crack was extended to almost the center. The neural network reported it as left, center and right crack.
	0,0,0,0,0,1,0 Plain, Swell	Fault	Plain was correct but not swell. The dark dot along the top and bottom edge might lead to misunderstanding to neural network.
	0,0,0,0,0,1,0 Plain, Swell	Fault	Swell was correct but plain was wrong since it contained a very small crack. This small features was difficult to detect.
	0,0,0,0,0,1,0 Plain, Swell	Fault	This was totally fault, very small crack was hard to detect and shadow in the bottom can be ambiguous with swell.
	0,0,0,0,1,1,0 Swell, right crack	Fault	The crack was located in the right and extended to the center, so the result should include the center crack. And also not swell
	0,0,1,0,0,1,0 Swell, Left crack	Fault	From the image, cracks were located on both left and right. But the right one was very small and hard to detect.

VII. CONCLUSION

This paper proposed the algorithm to analyse and classify the cross-section of shrapnel by using image processing techniques and artificial neural network.

According to the experimental data, the cross-section area detection yielded the accurate result at 98.05% because the background was in plain color and the foreground was clear and easy to detect.

For the dominant features area detection, the accuracy was dropped to 87.3% because the detail in this area was quite small and more difficult to detect. The classification by neural network yielded the 71% accuracy which was the total accuracy of this system.

This research could be used as a prototype for software application to automate the image analysing for finding the correlation between group of shrapnel evidences or shrapnel evidence and agitator which could help the EOD to save time and effort in their work.

Conflict of Interest

The author declares no conflict of interest.

Author Contributions

Kittiya conducted the research, analyzed the data, wrote the paper and approved the final version.

References

[1] C. Liangboonprakong and O. Sornil, "Classification of malware families based on N-grams sequential pattern features," in *Proc. 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, Melbourne, VIC, 2013, pp. 777-782.

[2] S. Areerachakul and P. Junsawang, "Rainfall-Runoff relationship for streamflow discharge forecasting by ANN modelling," in *Proc. World Congress on Sustainable Technologies (WCST-2014)*, London, 2014, pp. 27-30.

[3] P. Jomsri, "Book recommendation system for digital library based on user profiles by using association rule," in *Proc. the International Conference on the Innovative Computing Technology (INTECH 2014)*, Luton, 2014, pp. 130-134.

[4] M. Sakthivel, B. George, M. Sivaprakasam, and V. J. Kumar, "Locating metal shrapnel in human body using GMR sensors," in *Proc. IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2014.

[5] B. Y. Wei, J. Y. Gao, J. G. Zhu, and K. J. Li, "Design of a large explosive ordnance disposal robot," in *Proc. Second International Conference on Intelligent Computation Technology and Automation*, 2009.

[6] C. Xu, J. Song, M. Chen, and L. Yu, "Network-based desktop virtual reality system for EOD simulation training," *Information and Business Intelligence. Communications in Computer and Information Science*, vol. 267, Springer, Berlin, Heidelberg, 2012.

[7] S.-P. Qi, H.-J. Zhang, X.-J. Li, and H.-L. Zhou, "Counting steel rods online using LQV neural network in real-time images," in *Proc. IEEE International Conference on Information Acquisition*, 2006.

[8] J. Zhang, Y. L. Cui, G. N. He, C. M. Luo, and G. J. Miao, "A new image processing enabled approach for detection of scratch defects for wire-type objects," in *Proc. 3rd International Conference on Information Science and Control Engineering (ICISCE)*, 2016.

[9] P. Jensuriyagul, S. Sreemuangklang, A. Leelasantitham, C. Nudol, D. Banthao, and N. Ritnoom, "A classifiable method of soft-drink crown-cap types using back propagation neural network," in *Proc. International PSU Engineering Conference*, IPEC 5, May 2010.

[10] M. Kample, "Image-based retrieval and identification of ancient coins," *IEEE Trans. Intelligent System*, vol. 24, issue 2, pp. 26-34, Mar./Apr. 2009.

[11] R. Key, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 29, issue 6 pp. 1153-1160, Dec. 1981.

[12] G. Jie and L. Ning, "An improved adaptive threshold canny edge detection algorithm," in *Proc. Rec. 2010 IEEE Int. Conf. Computer Science and Electronics Engineering*, pp. 164-168.

[13] K.-H. Lee, A. Morales, and S.-J. Ko, "Adaptive basis matrix for the morphological function processing opening and closing," *IEEE Trans. Image Processing*, vol. 6, issue 5, pp. 769-774, 1997.

[14] H. Heijmans, M. Buckley, and H. Talbot, "Path based morphological openings," in *Proc. Conf. Rec.2004 IEEE Int. Conf. Image Processing*, vol. 5, pp. 3085-3088.

[15] D.-X. Zhang, C. Liu, Z.-Q. Wang, and N.-B. Liu, "A new fast learning algorithm for multi-layer feedforward neural networks," in *Proc. Conf. Rec. 2006 IEEE Int. Conf. Machine Learning and Cybernetics*, pp. 2928-2934.

[16] J. H. Cao, Z. B. Wei, T. Huang, and X. W. Xiong, "Analysis of feature extraction criterion function maximum in nonlinear multi-layer feedforward for pattern recognition," in *Proc. Conf. Rec. 2010 IEEE Int. Conf. Intelligent Computation Technology and Automation (ICICTA)*, pp. 655-658.

Kittiya Poonsilp receives M.Sc degree in computer science from Walailak University, Thailand in 2006. Her B.Sc degree in computer science from Suan Sunandha Rajabhat university, Thailand in 2001.

She is a lecturer, researcher and currently the head of the Computer Science Department, Suan Sunandha Rajabhat university, Thailand.

She is interested in several research topics including image processing, computer vision, machine learning, deep learning, knowledge graph and semantic network.