https://dx.doi.org/10.1016/j.asoc.2021.107115

Ball Bonding Inspections using a Conjoint Framework with Machine Learning and Human Judgement

Kit Yan Chan, Ka Fai Cedric Yiu, Hak-Keung Lam, Bert Wei Wong,

Abstract

Ball bonding inspections with human vision are essential in manufacturing processes of semiconductors devices and integrated circuits (ICs). The inspections are an intensive task which involves human labors to detect poor bonds. Prolonged visual inspections cause poor inspection integrity due to eye-fatigue. However, inspections nowadays are mostly conducted manually by humans which cannot satisfy the demanding productions. Motivated by the extraordinary performance of machine learning for manufacturing inspections, a detection framework integrated with machine learning and human judgement is proposed to aid bonding inspections based on visual images. The detection framework is incorporated with the convolution neural network (CNN), support vector machine (SVM) and circle hough transform algorithm (CHT); human judgement is only used when the detection uncertainty is below the threshold. The novel machine learning integration is proposed on the detection framework to improve the generalization capabilities. The CNN architecture is redeveloped by incorporating with the SVM which is generally more effective than the fully connected network in the classical CNN. Also a novel training function is proposed based on the CHT to ensure the inspection reliability; the function not only takes into account real image captures, but also locates important features using pattern analysis of the ball bondings. Experimental results show that significantly better classifications can be achieved by the proposed framework

Kit Yan Chan and Bert Wei Wong are with the Department of Electrical and Computer Engineering, Curtin University, Perth,WA 6102, Australia (e-mails: kit.chan@curtin.edu.au; Wei.Wong@curtin.edu.au);

Ka Fai Cedric Yiu is with the Department of Applied Mathematics The Hong Kong Polytechnic University, Hong Kong (e-mail: macyiu@polyu.edu.hk);

Hak-Keung Lam is with the Department of Informatics, King's College London, 30 Aldwych, London, WC2B 4BG, U.K. (e-mail: hak-keung.lam@kcl.ac.uk).

Manuscript received xxx x, 2020.

compared with the classical CNN and other commonly used classifiers. Only the machine learning determinations below the threshold are reassessed by human judgements.

Index Terms

Machine learning, human judgement, threshold detection, manufacturing of electronic products, manufacturing inspection, ball bonding.

I. Introduction

Modern semiconductor device assembly and interconnection technologies, known as electronic packaging, provide housing and interconnection of integrated circuits to manufacture electronic products. Electronic packaging, such as die attach, is the heart of electronic products involved with microcontrollers, light-emitting diode (LED) products, microchips and integrated circuits. Die attach is a cost-effective and flexible interconnect technology to develop electrical interconnections between electronic systems in semiconductor products. Interconnections can be used for system levels of printed circuit boards such as providing connections to small electronic devices. Since electronic appliances require increased reliability and performance at smaller sizes, increasingly precise of ball bondings is essential to connect many small components onto a small printed circuit board. Inspection of ball bonding conditions is critical since the bondings with poor conditions cannot be reproducible. The inspection attempts to detect whether the ball bonding conditions are satisfactory.

Assume that a ball bonding machine produces 20 bonds per seconds and we have 100 machines in the manufacturing line [1]. When all machines are operated for 24 hours, 172,800,000 bonds are produced everyday and all bonds are required to be inspected. However, the inspections are mostly conducted manually by humans with the aid of microscopes [2]. As human inspections are time consuming and expensive, only a small sample is taken from a manufacturing batch. Also, the inspections are prone to error due to eye fatigue after prolonged working hours. The inspections are unreliable due to susceptible errors which reduce manufacturing efficiency. As human inspections cannot satisfy the massive productions of electronic appliances, commercial vision inspection systems are developed based on 2-D images captured for ball bonding processes [3]. The vision inspection system first locates bonding ball positions in order to trace and inspect

the ball bonding conditions. To detect bonding balls, the pattern matching method is used by comparing the captured image patterns with a set of collected bonding samples [3]. However, long computational time and an extra database is required to track and locate image data.

Despite using the pattern matching, circle hough transforms (CHT) can be used to detect bond balls in digital images [4], [5]. In the CHT, the potential centers of circle patterns are nominated, and the patterns with high votes are elected as the existing circles [6]. CHTs have also been used for other electronic packaging inspections such as solder joints [7], [8] and solder pastes [9]–[11]. Although the CHT is simple and involves only a few algorithmic steps, the CHT requires large memory storage since a 4-D parameter space is used to vote circle patterns with different radius [12]. Also, misclassification is likely to be generated when irregular patterns are captured. For example, Fig. 1 shows four ball bonding patterns, where bond balls exist in Fig. 1a and 1b and bond ball is not in Fig. 1c and 1d . The CHT is likely to detect correct circle patterns in Fig. 1a and 1b but it mistakenly detects a circle pattern in Fig. 1d which exists a small hole and is actually a manufacturing defect. A wire is supposed to be bonded in the small hole in Fig. 1d but does not exist. A wrong inspection is caused.

In this paper, a detection framework integrated with machine learning and human judgement is proposed to provide more robust and reliable detections for ball bondings. The detection framework is incorporated with human judgement and machine learning; human judgement is only used when the detection uncertainty generated by the machine learning is below the threshold. The convolutional neural network (CNN) [13] is proposed to be the framework backbone since the CNN has the extraordinary performance for many pattern recognition applications [14] such as solder joint inspection of surface mount [15], quality monitoring of grain products [16], detecting defect in gas pipeline [17], scene understanding [18], crack patterns in tunnels [19], human actions [20] and ECG features [21]. The CNNs use the entire images to perform detections while classical classifiers only use a few image features. Hence, more robust and reliable detections are likely to be performed by the detection framework.

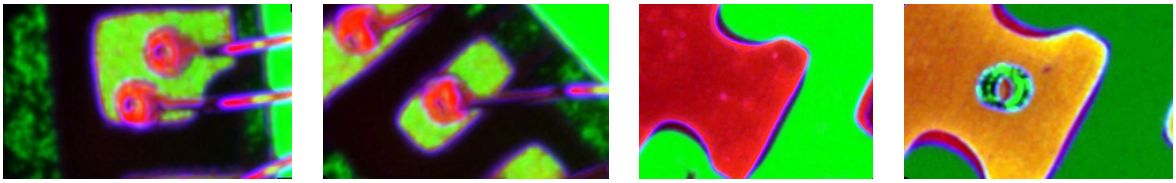
To further improve the generalization capabilities of the proposed detection framework, two strategies are proposed. First, a novel training function is proposed to develop the CNN. Unlike classical CNNs which only optimize the loss function which counts the

correct estimates, the proposed training function is integrated with a penalty term which takes into account the similarities of image features of ball bondings. When images with similar features are given to the CNN but different estimates are generated, some estimates are likely to be incorrect and the CNN penalizes itself through the training. The image features are measured by the CHT which is commonly used for ball bonding inspections [4], [5]. These image features are guaranteed to train the CNN, while the classical training only involves the full image and the image features may not be used for training. Second, the CNN configuration is redeveloped to achieve more robust classifications. Unlike the classical CNN configuration which uses a fully connected neural network in the last layer, the SVM is used in the last layer. The proposed configuration uses the features generated at the last CNN layer as the inputs of the SVM. The SVM is generally more effective for binary classifications or diagnosis compared to the fully connected neural networks or statistical regression models. The classical CNN configuration is ineffective for binary classifications such as handwritten digit recognition [22], target detection at radar images [23], animal classification [24], when comparing to the configuration which uses SVM in the last layer. Since the proposed configuration is incorporated with the SVM, better generalization capabilities are likely to be obtained by the proposed detection framework.

The performance of the proposed detection framework is evaluated by a set of bonding images which are captured by the ball bonding machines. Experimental results shows that the proposed CNN achieved significantly better classifications, compared with the classical CNN and the other commonly used approaches for binary classification including logistic regression, SVM and decision tree. Analytical discussions are given to support why better classification results can be obtained. Also the experimental results show that only the machine learning determinations below the threshold are necessary to be reassessed by human judgements when the proposed detection framework is used.

The main contribution of this article is summarized as follows:

1) A semi-automatical framework is developed to perform die bonding detections. The detections are mostly conducted by the proposed novel CNN in the framework. Only the detections which are below the threshold are reassessed by human judgements. The framework attempts to reduce human resources to perform man-



(a) (b) (c) (d)

Figure 1: IC chip with bonding ball on the base material

ual detection. Also detection accuracies can be improved when this framework is implemented.

2) The detection framework is incorporated with the CNN, SVM and circle hough transform algorithm. The framework attempts to enhance the detection accuracies of the commonly used detection algorithms and the CNN which is effective to perform pattern detection and recognition. Also experimental results demonstrate more accurate detection can be achieved by the proposed framework.

II. Ball bonding inspection

Fig. 2 shows the IC chip with ball bondings and it also shows the bonding conditions in the left hand side, where the bond balls are captured by green circles; the ball bondings in good conditions are captured by blue squares; and the poor ball bonding with a short circuit is captured by a red square. In order to trace the wire and inspect the bonding conditions, the computational algorithm first locates positions of bonding balls [3], [25]. The algorithm starts from the located bonding balls to trace the wires. It detects the short circuit between the wire pair which is captured by the red square in the figure. The algorithm also checks whether shift exists in the ball position since slight shift may cause an electric short between two adjacent pads.

To locate the bond balls, the circle hough transform (CHT) can be used [6], where the CHT have also been used in inspections of solder joints [7], [8], solder pastes [9], [10] and wire bonding [4], [5]. When a bond ball with a circle radius is required to be located, the CHT determines the center of a potential circle by "voting" each pixel of the image and then the CHT elects the pixel with the maximum vote as the determined center. When the image, M, is given, the CHT algorithm in Algorithm II, namely $\mathrm{CHT}(M, R, \tau)$, computes the likelihoods h_{R+k} with $k = 0, 1, .., \tau$ such that circle with radius, $R + k$,

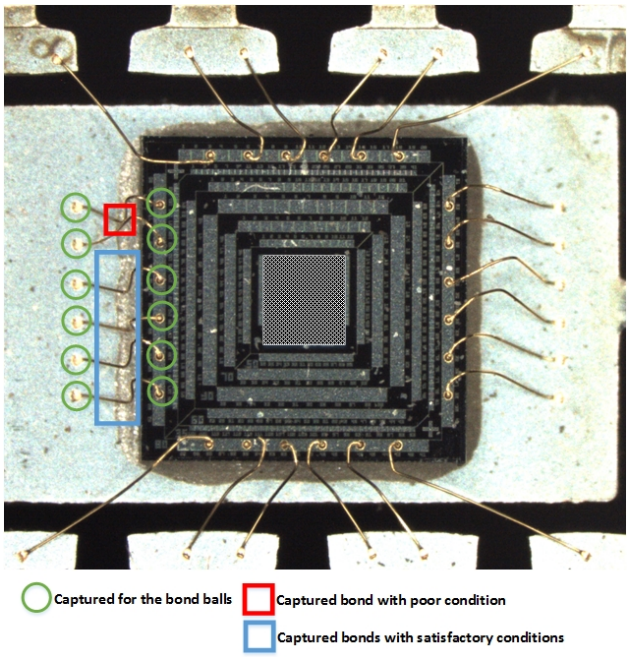


Figure 2: IC chip with ball bondings on the base material

exists in M; R, $R+1$,...and $R+\tau$ are the typical bonding ball radius; All h_{R+k} are determined by the CHT features, which are recently used to determine bold balls [7], [8], [10]; CHT(M, R, τ) attempts to find circle candidates in a digital image where each candidate is formulated by a center, (c_x, c_y), and a particular radius, r, in (1). The circle candidates are produced by "voting" the Hough space and the best candidate is selected with a maximum voting.

$$(x - c_x)^2 + (y - c_y)^2 = r^2 \tag{1}$$

Fig. 3 shows the image patterns for the ball bonding and also their corresponding CHT features. Fig. 3a and 3b show the images with one and two bond balls, and 3c and 3d show their corresponding CHT patterns respectively. Fig. 3c shows clearly that the brightest point is the bond ball center which is given in Fig. 3a. Also Fig. 3d shows two brightest points which are likely to be the two bond ball centers in Fig. 3b. These two examples show that bond ball centers can be correctly detected by the CHT. For the images carrying no bond ball, Fig. 3e and 3f are shown and also their corresponding CHT features are shown in Fig. 3g and 3h. Fig. 3e shows a clear image without a bond

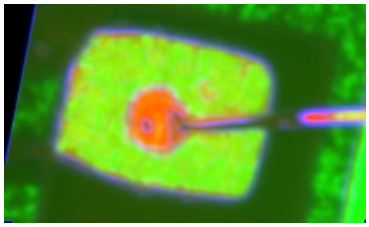
Algorithm 1 Circle hough transform algorithm (CHT)

procedure CHT(M, R, τ) ▷ M is the image
▷ Required radius: R to $R + \tau$
Convert M to gray scale
Detect edges on the gray M by Canny operator
$A[x, y] \leftarrow 0$ ▷ Setting all $A[x, y]$ to 0
▷ (x, y) are the pixels in $M^{(i)}$
for r = R to $(R + \tau)$ **do**
for $\forall(x, y)$ in M **do**
for $t = 0$ to 2π **do** ▷ angle from 0 to 2π
c_x=x–$r * cos(t)$
c_y=y–$r * sin(t)$
▷ polar coordinate for r and t
$A[c_x, c_y] + +$
▷ voting by adding $A[c_x, c_y]$ by 1
$h_{i,r} = \max_{\forall c_x, c_y}(A[c_x, c_y])$ ▷ Maximum value in A
return $(h_{i,R}, h_{i,R+1}, ..., h_{i,R+\tau})$

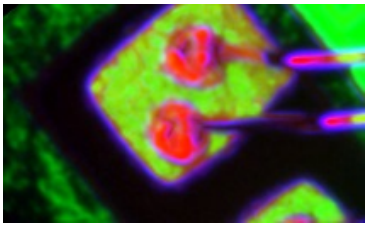
and no bright point exists in the corresponding CHT feature in Fig. 3g. Hence, correct detection can be generated by the CHT. Fig. 3f shows an uncompleted ball bonding. These irregular patterns cause inappropriate CHT analysis, where bright points exist in the corresponding CHT features in Fig. 3h. Therefore, an incorrect detection can be generated when ones use these CHT features for the detection. Despite this limitation, CHT requires large storage for hough features and the computational complexity is high as the transformation requires 4-D parameter spacing [11].

III. Proposed detection framework

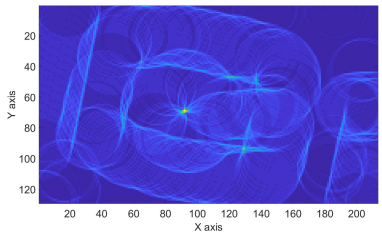
This section discusses the proposed detection framework illustrated in Figure 4, which is incorporated with the CHT, the machine learning algorithm namely convolution neural network (CNN) and the human judgement. In the proposed framework, a set of ball bonding images are captured from the ball bonding machine. Human is employed to classify the images with either bonds or no bond manually. Pattern labels are marked for the images to indicate whether the bond exists or not. Those labeled images are included in the database and are used to trained the CNN. Since the CNN is trained by the human selected data, the CNN simulates the human decision for detecting bonds. Also, the CNN is incorporated with both the SVM and the CHT of which the CHT fully



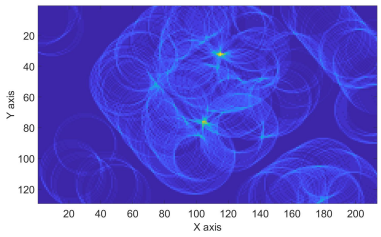
(a) Image with a bond



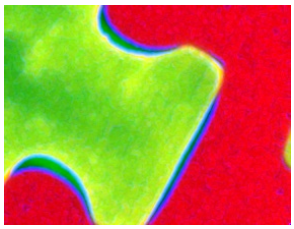
(b) Image with two bonds



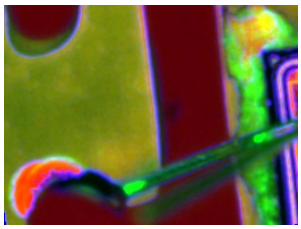
(c) CHT with a bond



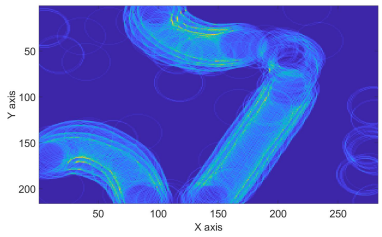
(d) CHT with two bonds



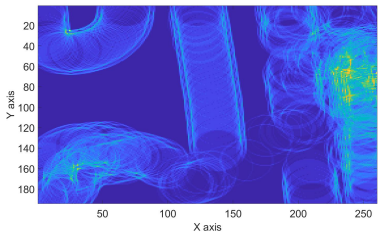
(e) Image with no bond



(f) Image with no bond



(g) CHT with no bond



(h) CHT with no bond

Figure 3: Hough transforms for ball bonding images

depends on the bond features to perform the detection. The incorporation overcomes the limitations of training the CNN that some bonding features are not fully included in the training data, and also the CNN achieves more robust and reliable detections since the SVM is incorporated.

After the CNN is trained, the CNN can be used to detect whether a bond exists when a newly image is given. The CNN generates a bond factor namely "$bond\_factor$" which indicates whether a bond exists or not. If "$bond\_factor$" is a positive value, the bond

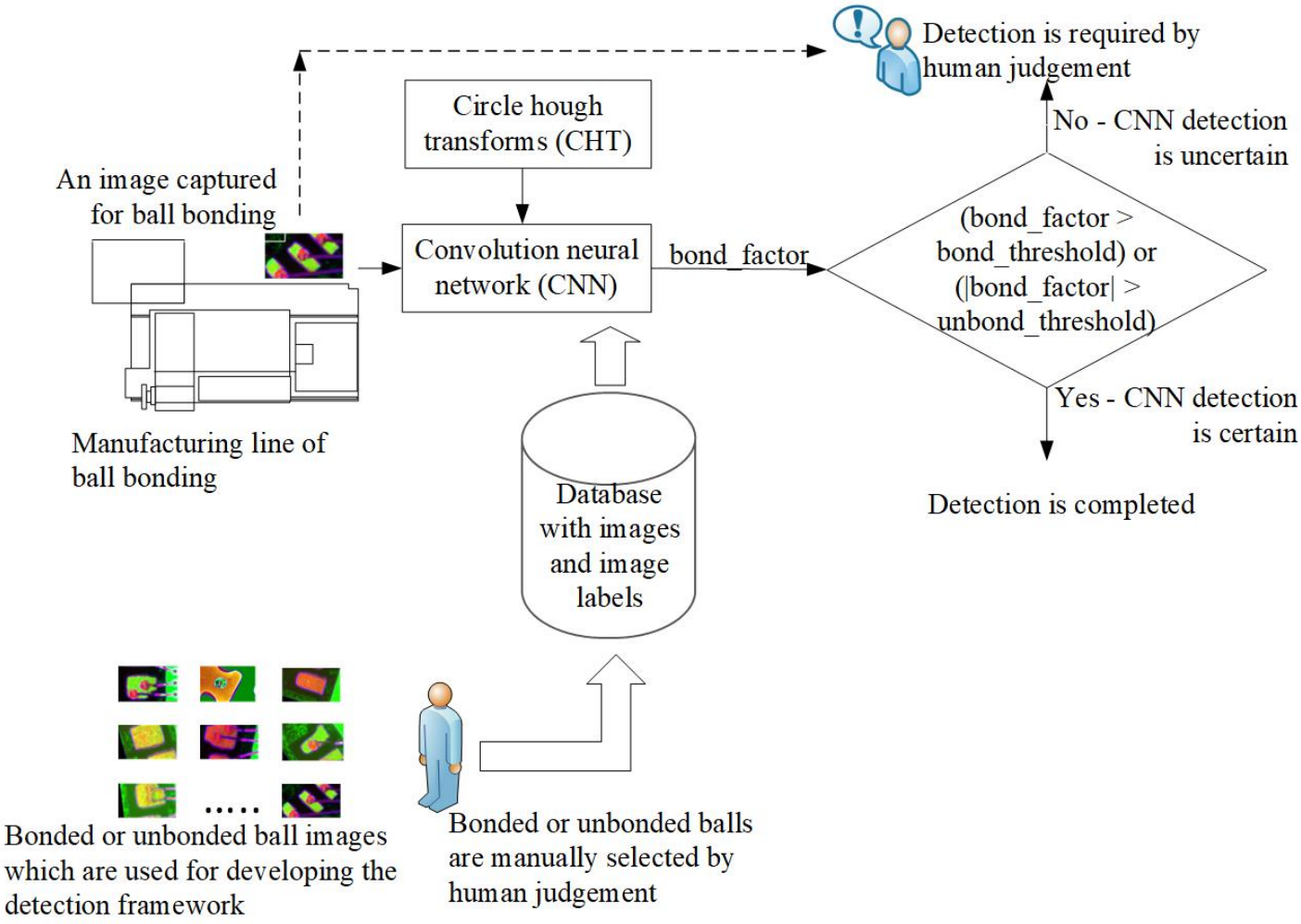


Figure 4: The proposed detection framework

exists; otherwise, if "$bond\_factor$" is negative, the bond does not exist. "$bond\_threshold$" and "$nobond\_threshold$" are the thresholds to indicate whether the CNN determination is certain. If "$bond\_factor$" is larger than "$bond\_threshold$" or "$|bond\_factor|$" is larger than "$nobond\_threshold$", the CNN determination is certain. Otherwise the CNN determination is not certain; Human is employed to perform the final judgement manually. Our experimental results in Section IV show that the absolute values of the bond factors are generally higher than the thresholds, human is barely necessary to perform the final judgements. Hence, this framework is able to save the human resources to manually detect and also enhance the commonly used approach, CHT, which is fully relied on image features to detect.

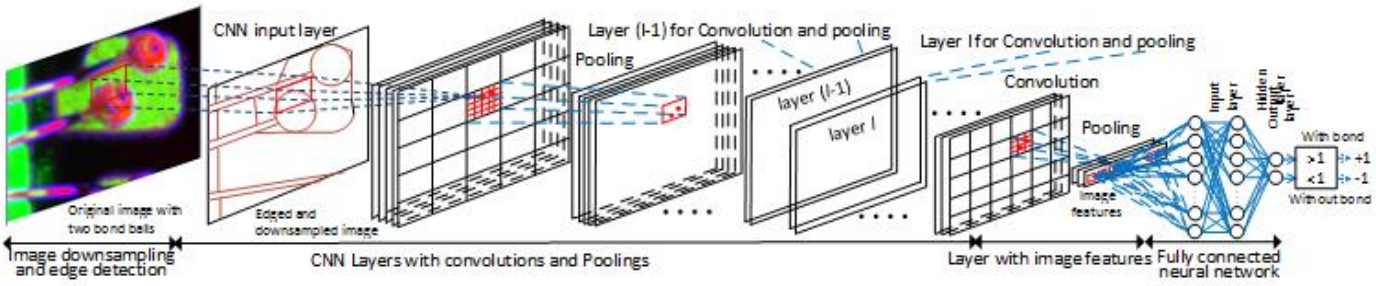


Figure 5: CNN for bonding ball detection

Section III-A discusses the CNN backbone for ball bonding detection. In Sections III-B and III-C, two strategies are proposed to improve the generalization capabilities. Section III-B discusses the proposed training function which takes into account the similarities between image patterns. Based on the similarities, more information of bond patterns can be used to train the CNN and better generalization capabilities are more likely to be obtained. Section III-C discusses the proposed CNN architecture which is integrated with the SVM. The proposed architecture attempts to further improve the generalization capability.

A. CNN backbone for ball bonding detection

For bonding ball detections, the CNN in Fig. 5is used as a classifier $f^{CNN} : M \rightarrow \hat{d}$, where M is the image pattern given to the CNN and $\hat{d}$ is the predicted pattern label which is either $+1$ or -1. If $\hat{d} = +1$, a bonding ball is detected in M. Otherwise, if $\hat{d} = -1$, no bonding ball is detected. In f^{CNN}, M is preprocessed by downsampling from a high resolution to low resolution and also the edge detection is performed on the downsampled M. The number of pixels of the preprocessed image is less than that of the original image. Also the edges attempt to cover the important image features. Hence, much less memory is required to develop the CNN. Based on the experiments, we found that there is no significant difference when using the preprocessed or the original images to train the CNN. After preprocessed the M, f^{CNN} has an alternate pattern transformation which consists of convolution and pooling.

The convolution of layer l is computed as,

$$s_j^l = \varphi\left(\sum_{\text{all}i} s_i^{l-1} * w_{ij}^l + b_j^l\right) \tag{2}$$

where " $*$ " denotes the convolutional operation; s_i^{l-1} and s_j^l are the i^{th} and j^{th} neurons at layers $l-1$ and l respectively; s_i^{l-1} and s_j^l are the square matrices which cover the square regions in layers $l-1$ and l; w_{ij}^l are the kernel weights between s_j^l and s_i^{l-1}; b_j^l is the bias for s_j^l, and $\varphi(x)$ is the activation function. As each s_j^l at layer l is convolved from all s_i^{l-1} in layer $l-1$, the convolutional layer can be used to detect local combinations of features from the previous layer. $\varphi(x)$ can learn nonlinear combinations of all s_i^{l-1} in the previous layer. In f^{CNN}, $\varphi(x)$ is the sigmoid transformation of the maximum sum of x.

Following the convolution, pooling is performed in each convoluted nervous, s_j^l. Here max-pooling is adopted, since it eliminates nonmaximal values in the convolution layer in order to reduce computation. Also it generates translation invariance and captures the more robust features for all s_j^l. Based on s_j^l, the max-pooling operation is computed as

$$\text{pooling}(s_j^l(m)) = \max_{x \in s_j^l(m)} (x) \tag{3}$$

where x is an element within a region of $s_j^l(m)$. The processes of convolution and pooling continue until the image features reach the last layer which is a fully connected neural network. The output of the last layer consists of two neurons which classify whether bonding balls exist in image patterns. A bonding ball is detected and $+1$ is generated by f^{CNN} when the 'With bond' value is higher. Otherwise, no bonding ball is detected when the 'Without bond' value is higher. f^{CNN} generates -1.

B. Proposed training function and Circle hough transform

f^{CNN} is developed based on a set of collected ball bonding images with pattern labels $(M^{(1)}, d^{(1)}), ..., (M^{(P)}, d^{(P)})$, where $M^{(k)}$ is the k^{th} image pattern of the chip with ball bonding; $d^{(k)}$ is either $+1$ or -1. When $d^{(k)} = +1$, a bonding ball is detected in $M^{(k)}$. When $d^{(k)} = -1$, no bonding ball is detected in $M^{(k)}$. Generally, a CNN is optimized

by the back propagation algorithm with respect to the mean square error function in (4). Unlike the mean absolute error which is not differentiable, the mean square error function is involved with the square term which is differentiable. Hence, the back propagation can be used to determine the network weights. Also the mean square error function attempts to eliminate the predictions with high errors, since the high errors between the predictions and actual observations dominate the function value when the square operation is used on each error.

$$MSE(\alpha) = \frac{1}{P}\sum_{i=1}^{P}(d^i - f^{CNN}(M^{(i)}, \alpha))^2 \tag{4}$$

where $\alpha = \{w_{ij}^l, b_j^l, \theta\}$ consists of all CNN parameters, namely network weight w_{ij}^l, network bias b_j^l, and the parameters of the fully connected neural network, θ. There is a potential limitation, when (4) is used to train the CNN. The CNN training is fully relied on $M^{(i)}$ and does not use the CHT features which are commonly used to detect circle patterns [6]. To overcome this limitation, a novel training function (5), is proposed by integrating the CHT features as the penalty,

$$\begin{aligned} RMSE(\alpha) &= \lambda\Big(\frac{1}{P}\sum_{i=1}^{P}(d^i - f^{CNN}(M^{(i)}, \alpha)^2\Big) \\ &+ \sum_{\substack{i,j=1 \\ i\neq j}}^{P} a_{i,j}(f^{CNN}(M^{(i)}, \alpha) - f^{CNN}(M^{(j)}, \alpha))^2 \end{aligned} \tag{5}$$

In (5), the first term is the product of $MSE(\alpha)$ in (4) and a large factor, λ. The first term forces $f^{CNN}(M^{(i)}, \alpha)$ to agree with $M^{(i)}$. The second term is the graph Laplacian regularization, which incurs a penalty when the patterns of $M^{(i)}$ and $M^{(j)}$ are similar but their predictions are different, i.e. $f^{CNN}(M^{(i)}, \alpha) \neq f^{CNN}(M^{(j)}, \alpha)$. $a_{i,j}$ indicates the similarity between $M^{(i)}$ and $M^{(j)}$ which is computed based on the CHT features. When $a_{i,j}$ is large, $M^{(i)}$ is similar to $M^{(j)}$. When $a_{i,j}$ is small, $M^{(i)}$ is very different to $M^{(i)}$. When $M^{(i)}$ and $M^{(j)}$ are similar, this is unlikely that $f^{CNN}(M^{(i)}, \alpha) \neq f^{CNN}(M^{(j)}, \alpha)$. Hence, the prediction is likely to be incorrect and a penalty is added to (5). The penalty is related to the value of $a_{i,j}$. When $a_{i,j}$ is larger and $f^{CNN}(M^{(i)}, \alpha) \neq f^{CNN}(M^{(j)}, \alpha)$, the

prediction is very likely to be incorrect. Hence, large penalty is imposed to (5). As the CNN attempts to determine whether circle bonding balls exist in the image patterns, $a_{i,j}$ is computed by (6) which indicates the correlation regarding the CHT features in $M^{(i)}$ and $M^{(j)}$. $a_{i,j}$ is high when similar CHT features exist in $M^{(i)}$ and $M^{(j)}$. Otherwise, $a_{i,j}$ is low.

$$a_{i,j} = C - \frac{\sum_{k=0}^{\tau} \left(h_{i,R+k} - h_{j,R+k} \right)}{\tau + 1} \tag{6}$$

where $h_{i,R+k}$ and $h_{j,R+k}$ indicate the likelihoods that circles with radius, $R + k$, exist in $M^{(i)}$ and $M^{(j)}$; R, $R + 1$,...and $R + \tau$ are the typical bonding ball radius; C value is defined based on the values of all $h_{i,R+k}$ and $h_{j,R+k}$. C ensures that all $a_{i,j} > 0$. All $h_{i,R+k}$ and $h_{j,R+k}$ are determined by the CHT features, which are recently used to determine bold balls [7], [8], [10]; The CHT algorithm attempts to find circle candidates in a digital image where each candidate is formulated by a center, (c_x, c_y), and a particular radius, r, in 7. The circle candidates are produced by "voting" the Hough space and the best candidate is selected with a maximum voting. The CHT algorithm is given in Algorithm 2, namely CHT$(M^{(i)}, R, \tau)$, which computes $h_{i,R+k}$ with $k = 0, 1, .., \tau$ when an image, $M^{(i)}$, is given.

$$(x - c_x)^2 + (y - c_y)^2 = r^2 \tag{7}$$

Algorithm 2 computed the correlations, a_{ij} in (6), for the four image patterns given in Fig. 3 which consists of two images with bonding balls and two images with no bond. a_{ij} are given in the following correlation matrix for the four images. We can see that the correlations between bonds and those between no bond are generally larger than those of the mutual ones. Hence, a penalty is imposed in (5) when image patterns with high

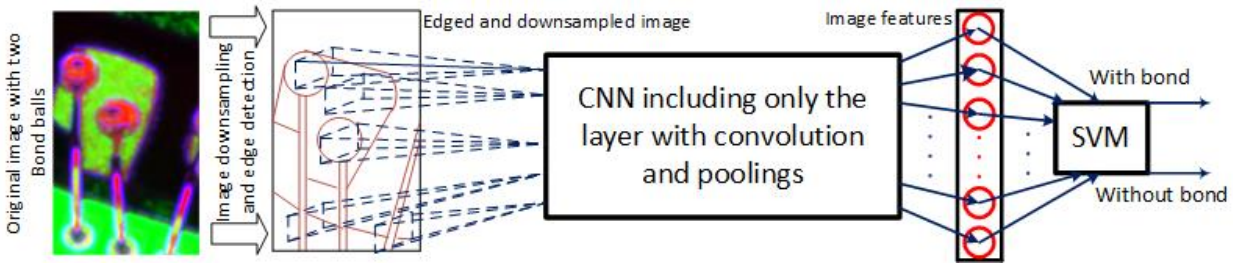


Figure 6: Architecture of the proposed CNN for bond detection

correlations are predicted as nonidentical.

$$\begin{array}{c} \\ \text{bond} \\ \text{bond} \\ \text{no bond} \\ \text{no bond} \end{array} \begin{array}{c} \begin{array}{cccc} \text{bond} & \text{bond} & \text{no bond} & \text{no bond} \end{array} \\ \begin{pmatrix} 1.3596 & 0.8025 & 0.3888 & 0.2373 \\ 0.8025 & 1.3596 & 0.1050 & 0.3594 \\ 0.3888 & 0.1050 & 1.3596 & 0.9996 \\ 0.2373 & 0.3594 & 0.9996 & 1.3596 \end{pmatrix} \end{array}$$

C. Proposed CNN architecture

Fig. 5 shows that a classical CNN consists of two parts, layers with convolutions and poolings and a fully connected neural network, where the layers with convolutions and poolings extract important image features and the fully connected neural network determines whether bonding balls exist on the image patterns. In the proposed CNN, the fully connected neural network is replaced by the SVM [26] which achieves higher generalizations in binary classifications than the fully connected neural network [22]–[24]. After the CNN is trained with (6), the image features generated at the last layer with convolution and pooling is used as the inputs of the SVM. The proposed CNN is illustrated in Fig. 6, where the fully connected neural network is replaced with the SVM [26] in order to detect whether bonding balls exist. In the SVM, existing of the bonding ball is classified by a hyperplane given in (8) and a bond factor ,"$bond\_factor$" is generated to determine whether the bond exists. If "$bond\_factor < 0$" , the bond exists; otherwise the bond does not exist.

$$\begin{aligned} &\text{if bonding balls exist, } bond\_factor = f^{PCNN}(\bar{x}_i) = \bar{w} \cdot \Phi(\bar{x}_i) + \beta \geq 0 \text{ and } d^i = +1; \\ &\text{if no bonding ball, } bond\_factor = f^{PCNN}(\bar{x}_i) = \bar{w} \cdot \Phi(\bar{x}_i) + \beta \leq 0 \text{ and } d^i = -1 \end{aligned} \tag{8}$$

where $\bar{w}$ is the SVM weight vector; $\Phi(..)$ is a feature-space transformation; $\bar{x}_i$ is the vector of the bonding ball features generated by the last CNN layer; $\bar{x}_i$ is corresponding to the image pattern $M^{(i)}$; $F^{PCNN}(\bar{x}_i)$ is the outcome of the proposed CNN; when $bond\_factor$ is positive, the bond ball is detected; when $bond\_factor$ is negative, no bond ball is detected; The dimensions of both $\bar{w}$ and $\bar{x}_i$ are the same. β is the SVM bias. (8) can be reformulated as the following Lagrange multiplier (9).

$$L(\bar{w}, \beta) = \frac{1}{2}||\bar{w}||^2 - \sum_{i=1}^{P} C_{SVM} \cdot ((d^i \cdot (\bar{w} \cdot \Phi(\bar{x}_i) + \beta)) - 1) \tag{9}$$

where C_{SVM} is a positive constant which ensures all d^i have to be classified correctly within a certain margin. (9) can be solved by the derivatives of $L(\bar{w}, \beta)$ with respect to $\bar{w}$ and β. The details of solving (9) can be referred to [26]. After determined $\bar{w}$ and β, the hyperplane (8) can be used to determined whether a bonding ball exists when the image features are given. In Figure 4, the two thresholds, "$bond\_threshold$" and "$inbond\_threshold$", are determined based on the correct detection of f^{PCNN} for the images with bonds and no bond respectively. The thresholds are defined in (10) based on the certain proportion of correct detections namely γ, which are nearest to the SVM hyperplane.

$$\begin{aligned} bond\_threshold &= f^{PCNN}(\bar{x}_i), \text{ with } f^{PCNN}(\bar{x}_i) > f^{PCNN}(\bar{x}_j) \text{ and } j = 1, 2, ..., \text{ round}(P^+ \times \gamma) \\ nobond\_threshold &== f^{PCNN}(\bar{x}_i), \text{ with } f^{PCNN}(\bar{x}_i) < f^{PCNN}(\bar{x}_j) \text{ and } j = 1, 2, ..., \text{ round}(P^- \times \gamma) \end{aligned} \tag{10}$$

where P^+ and P^- are the numbers of data with bonds and no bond respectively.

IV. Performance evaluations for the proposed framework

Algorithm 2 Circle hough transform algorithm (CHT)

procedure CHT($M^{(i)}, R, \tau$) ▷ $M^{(i)}$ is the image
▷ Required radius: R to $R + \tau$
Convert $M^{(i)}$ to gray scale
Detect edges on the gray $M^{(i)}$ by Canny operator
$A[x, y] \leftarrow 0$ ▷ Setting all $A[x, y]$ to 0
▷ (x, y) are the pixels in $M^{(i)}$
for r = R to $(R + \tau)$ **do**
for $\forall(x, y)$ in $M^{(i)}$ **do**
for $t = 0$ to 2π **do** ▷ angle from 0 to 2π
c_x=x–$r * cos(t)$
c_y=y–$r * sin(t)$
▷ polar coordinate for r and t
$A[c_x, c_y] + +$
▷ voting by adding $A[c_x, c_y]$ by 1
$h_{i,r} = \max_{\forall c_x, c_y}(A[c_x, c_y])$ ▷ Maximum value in A
return $(h_{i,R}, h_{i,R+1}, ..., h_{i,R+\tau})$

A. Image patterns of ball bonding and experimental setup

In this section, we evaluated the generalization capability of the proposed CNN, namely CNN-Hough-SVM, when performing bonding ball detections, where the CNN-Hough-SVM is in the proposed framework illustrated in Fig. 4 The CNN-Hough-SVM is integrated with the training function (5) in Section III-B and the SVM mechanism in Section III-C. The image patterns were collected from the ball bonding machines which are installed with a high resolution camera. Copper wires are used as the material for ball bonding interconnects, where the copper can be developed as a fine bonding in sizes up to the 0.003 inch (75 micrometers). We have collected 494 image patterns where bonding balls exist in 384 image patterns and 110 image patterns carry no bonding balls. The images are 24 bit depth in the bitmap format, which can represent more than 16 million unique colors and is enough for humans to distinguish whether bonding balls exist. The collected images have the resolutions between 129×210 pixels to 129×283 pixels. Based on the collected patterns, the CNN-Hough-SVM are developed. We have also evaluated another proposed CNN, namely CNN-Hough, which is integrated with the proposed training function (5). SVM is used in the last layer of CNN-Hough-SVM and the fully connected neural network is used in the last layer of CNN-Hough. This

comparison attempts to evaluate the classification performances when the proposed training function (5) is used but different mechanisms are used in the last layer.

The manufacturing company supporting this research collected the images which are used for developing the CNNs. The samples are included with both regular and irregular images. The irregular images were particularly collected in order to evaluate the classification performance, where the irregular images were captured by numerous bonding machines. The images were captured on a bonding machine which was performing the bonding operations. Light flashes were performed when capturing the images. The regular images are captured, when the lighting conditions are satisfactory. The irregular images were particularly collected, when the unsatisfactory lighting are accidentally flashed. Also some of the irregular images are included with defects. The irregular images are more challenging to be recognized correctly by the CNN. Among all 494 images, 69 irregular images are included, where 24 of those are involved with bonding and 45 of those have no bond. Since the total number of image patterns are limited, we have generated extra image patterns by performing data augmentation. We have used the ImageDataGenerator function in python tensorflow, in order to generate 9 extra images from each collected image by angle rotations. Hence we have created 4446 images (i.e. 9×494)and have used 4940 (i.e. 10×494) to generate the classifiers.

Despite the proposed methods, we have also evaluated the performance of the classical CNN, namely CNN-FCN which is identical to the CNN mechanism in Fig. 5. CNN-FCN is identical to the proposed CNN-Hough except that different training function is used, where (4) is used in CNN-FCN and (5) is used in CNN-Hough. (4) is similar to (5) except that the penalty term in (5) is omitted in (4). This comparison attempts to evaluate the effectiveness of the proposed penalty function. Table I shows the parameters used in the three CNNs. The parameters of the CNNs were determined based on the default values used in the tool developed by [27]. Based on the default values, we tested the CNN for several times until no significant improvement can be achieved. The classification performances of the three CNNs were evaluated when long and short training time was used. For the long training time, 4000 epochs were used on the three CNNs namely CNN-FNN-4000, CNN-Hough-4000 and CNN-Hough-SVM-4000. For the shorter training, 400 epochs were used for CNN-FNN-400, CNN-Hough-400 and CNN-Hough-SVM-400. We have also compared the results obtained by the

Table I: CNN parameters used in CNN-FCN, CNN-SVM-Hough and CNN-Hough

CNN Parameters	Settings
Kernel size of the 1^{st} and 2^{nd} convolution layers	5×5
Kernel size of the 1^{st} and 2^{nd} pooling layers	2×2
Number of kernels in the 1^{st} convolution layer	12
Number of kernels in the 2^{nd} convolution layer	8
Number of features in the image feature layer	16
Number of hidden nodes in the CNN-FCN	128
Initial learning rate of the backpropagation	0.01

three commonly used classifiers namely, logistic regression (LR) [28], support vector machine (SVM) [29], and decision tree (DT) [30]. The matlab version of LR, SVM and DT has been implemented. To further compare the performance of the proposed CNNs, we have implemented the XGBoost python version of boosted tree algorithm (BT).[1] The CHT values of the typical bonding ball radius were used as the input features of the classifiers, where the typical bonding ball radius are 17 to 21 pixels of the captured images. These classifiers are considered since they are commonly used for classifications when numerical features are used. Also the CHT values are commonly used for identifying bonding balls or soldering patterns in electronic packaging [3], [8], [10]. Based on this comparison, the generalization capabilities of the proposed methods can be further validated. All the algorithms were implemented in MatlabR2017a and the scripts were developed in Matlab codes based on [27]. The algorithms were executed by a P510 Xeon E5-1630 v4 machine with 32GB memory and two xGTX1080 GPU cards.

B. Ball bonding detection with CNN

This section presents the detection results of the proposed framework of which the CNN is only used and the human judgement is not involved. The performance of the proposed CNNs is evaluated. Cross-validations were performed to evaluate the fitting and generalization capabilities of the proposed CNN-Hough and CNN-Hough-SVM, and the other tested classifiers. Since different classifiers are generated when different training datasets are used, more than one cross validation are used to evaluate the overall performance of the classifiers. In this experimental study, 30 cross validations

[1]XGBoost Documentation: https:xgboost.readthedocs.io

were performed to develop the classifiers. Among the 4940 image patterns, 3940 image patterns were randomly selected as the training dataset and the rest 1000 image patterns were used as the test dataset to evaluate the generalization capability of the trained classifiers. In the training dataset, 3000 image patterns were randomly selected and were used to develop the classifiers, and the rest 940 image patterns were used for the validations in order to avoid overtraining. The cross validation is used, since the number of collected images is limited. The cross validation attempts to maximize the use of the collected images for training and testing the classifiers. It assesses the generalization capabilities of the classifiers when different combinations of training and test samples are used. The cross validation helps to assess the CNN performance in a more unbiased way than solely training and testing the CNN with a single fold.

The CNNs are trained by minimizing the cost functions, where (4) is used for the CNN-FCN and (5) is used for the proposed CNN-Hough and CNN-Hough-SVM. The CNN was trained when the validation error started to increase in order to avoid overtraining the CNN. Figure 7 shows the characteristic of the CNN-Hough-4000 of which 4000 epochs were pre-defined. The figure shows that the validation error started to increase after 450 epochs. To avoid overfitting, the trained CNN-Hough-4000 was generated in this epoch number. After the classifier was trained, its generalization capability was evaluated based on the test dataset of which no sample is duplicate to the training dataset. The training and test errors indicate the incorrect rates of the training and test dataset respectively. They indicate the fitting and generalization capabilities of the classifiers respectively.

Fig. 9 and 10 show the training and test errors for the CNNs when 400 and 4000 epochs were used respectively. For 400 epochs, Fig. 9a shows that the three CNNs namely CNN-FCN-400, CNN-Hough-400 and CNN-Hough-SVM-400, achieved smaller training errors than those of the classical classifiers, LR, SVM and DT. Fig. 9b also shows that the three CNNs achieved smaller test errors than the classical classifiers. Therefore the fitting and generalization capabilities of the CNNs are generally better than the classical classifiers, although small number of epochs was used in the CNNs. Fig. 10a and 10b show the training and test errors when 4000 epochs were used. The results of classical classifiers, LR, SVM and DT are not shown in the two figures, as those results are already shown in Fig. 9. Fig. 10a shows that the training errors of the proposed

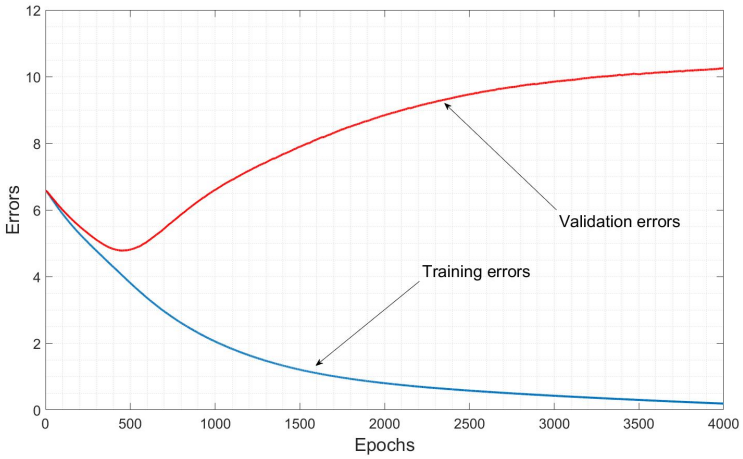


Figure 7: Characteristics of training and validation

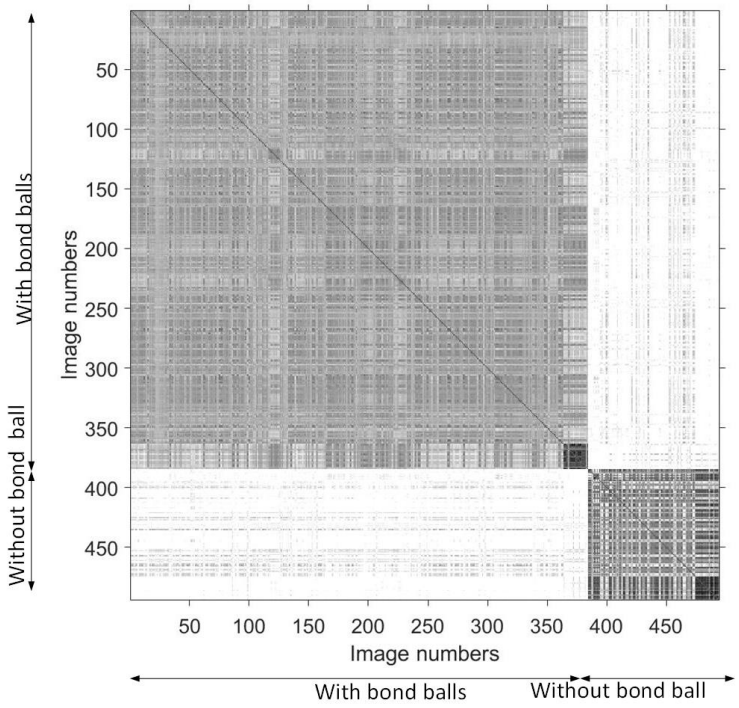


Figure 8: Correlations between image patterns

CNN-Hough-SVM-4000 and CNN-Hough-4000 are generally smaller than the classical CNN-FNN-4000. Hence the fitting capabilities of the proposed methods are better than the classical one.

For the test errors, it is difficult to distinguish the differences between the proposed and the classical methods. Since cross validation is a stochastic operation, the training

samples, validation samples and test samples are randomly split from the whole set of collected samples. Hence, in each split, the training samples, validation samples and test samples are different for each validation. When ones attempt to compare the performances of two methods, ones would analyze the average training and test errors obtained in all validations. Also ones would calculate the variances for the errors for all validations. Based on the means and the variances, the t-test can be used in order to determine the confident level of how the proposed method outperforms the tested method. Tables II and III show the means, variances, improvement and t-tests. The second rows show that the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000 achieved smallest means and variances for both training and test errors, comparing to the commonly used classifiers, LR, SVM and DT. Also the results of CNN-Hough-400 and CNN-Hough-SVM-400 with smaller epochs are shown. These results demonstrate that larger epochs are necessary since smaller training and test errors can be achieved. To further demonstrate the performance of the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000, the percentages of relative improvements are shown. For the training errors, the third and the fourth rows show that more than 80% improvements can be achieved by the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000, compare to the tested classifiers. For the test errors, more than 40% of improvement can generally be achieved.

The last two rows of Tables II and III show that the t-values between the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000 to the other tested methods. Based on the t-distribution table, if the t-value is higher than 2.01, the misclassification rates obtained by the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000 are significantly smaller than those of the other tested methods at a 97.5% confidence level. The results show that the t-values of the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000 are higher than 2.24. Hence, the proposed methods are able to obtain better classification rate than the tested methods with a 97.5% confident level. In Tables II and III, the comparison is indicated by the improvements and t-values, and the comparison is corresponded between the proposed method to the other methods. Since we cannot compare the performance of the proposed method to its own, 'Nil' is put on the tables.

Despite the classification rates, the three performance metrics namely precision, recall, F-measure, type-I error and type-II error are also used to further validate the generaliza-

tion capability of the proposed CNN methods. Precision indicates the rate that the bond ball samples are correctly predicted to the total bond ball samples are predicted. Recall indicates the rate that the bond ball samples are correctly predicted to the total bond band samples exist in the dataset. F-Measure indicates a score that evaluates whether the precision and recall are balance. F-Measure is approximately the average of precision and recall when both are close. F-Measure is formulated as (11)

$$\text{F-Measure} = 2 \cdot \frac{(\text{precision}) \cdot (\text{recall})}{(\text{precision}) + (\text{recall})} \tag{11}$$

The type I error (i.e. the false positive) is the rejection of a true null hypothesis of which at least a bond ball exists but no detection is predicted. The type II error (i.e. the false negative) is the non-rejection of a false null hypothesis of which no bond ball exists but detection is predicted.

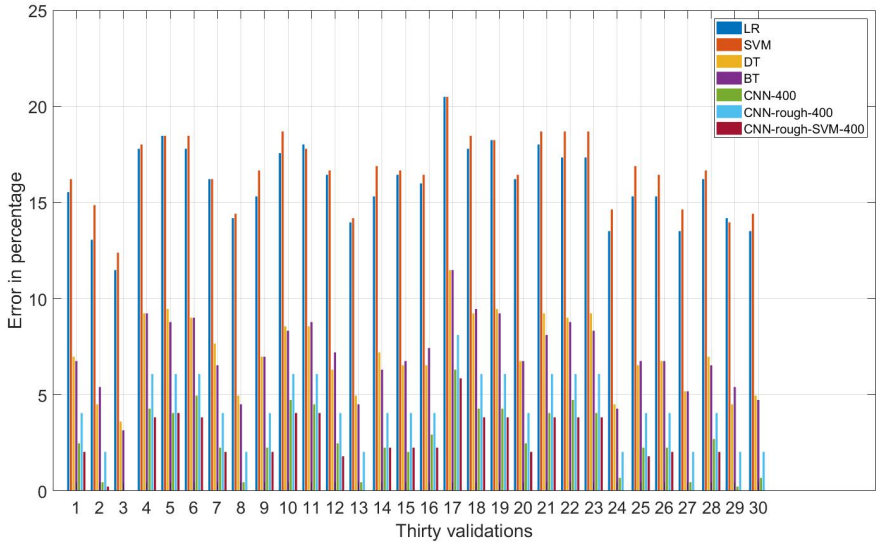
The results obtained by CNN-Hough-4000 and CNN-Hough-SVM-4000 are shown, since both of the proposed CNN methods outperform the other tested methods in term of the classification rates. The precision, recall, F-measure, type-I error and type-II error are shown in the first, second, third fourth and fifth main rows of Table IV respectively. For each metric, the means and variances obtained by each method are also shown. These results show that the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000 are better than the CNN-FCN-4000 which is generally better than LR, SVM, DT and BT. Also the t-tests were used to evaluate the significance of the hypothesis that the averaged metric value obtained by the proposed CNN-Hough-4000 and CNN-Hough-SVM-4000 are better than that obtained by another method. If the t-values are higher than 2.24 and 1.89, the metric values obtained by the proposed CNN approach are significantly larger than those of another tested method at 97.5% and 94.1% confidence levels. The t-values show that both the proposed CNN-Hough-SVM-4000 and CNN-Hough-4000 are able to obtain better metric values than the LR, SVM, DT and BT with a 97.5% confident level. Also both the proposed CNN-Hough-SVM-4000 and CNN-Hough-4000 obtained significantly better metric values than the CNN-FCN-4000 with 97.5% and 94.1% confidence levels respectively. These results further demonstrate the generalization capability of the proposed CNN methods.

Although the CNNs can generate more accurate detections compared to the classical classification methods including the LR, SVM, DT and BT, a large number of parameters are required in the CNNs. A large memory space is required to implement the CNNs. Table V shows the numbers of parameters required on the CNN-Hough-4000 and CNN-Hough-SVM-4000 of which the network configurations can be referred to Table I. The numbers of parameters of LR and SVM are also shown. The table shows that the numbers parameters required by the CNN-Hough-4000 is much larger than that of CNN-Hough-SVM-4000 which is larger than those required by the LR and SVM. Therefore, this is a trade-off to select the classification methods. When ones attempt to have more accurate detections, CNN-Hough-4000 is used; when the memory space is concerned, CNN-Hough-SVM-4000 are used. When the ensemble scheme [31] is used, the fully connected network based CNN is parallelly connected with the SVM. The total number of parameters is the parameters of the fully connected network based CNN and those of the SVM; the total number of parameters is much larger than the one in the CNN-Hough-SVM-4000. This is another reason that the CNN is modified by replacing the fully connected network with the SVM rather than parallelly connecting the fully connected network based CNN with the fully connected network and the SVM as in the ensemble scheme.

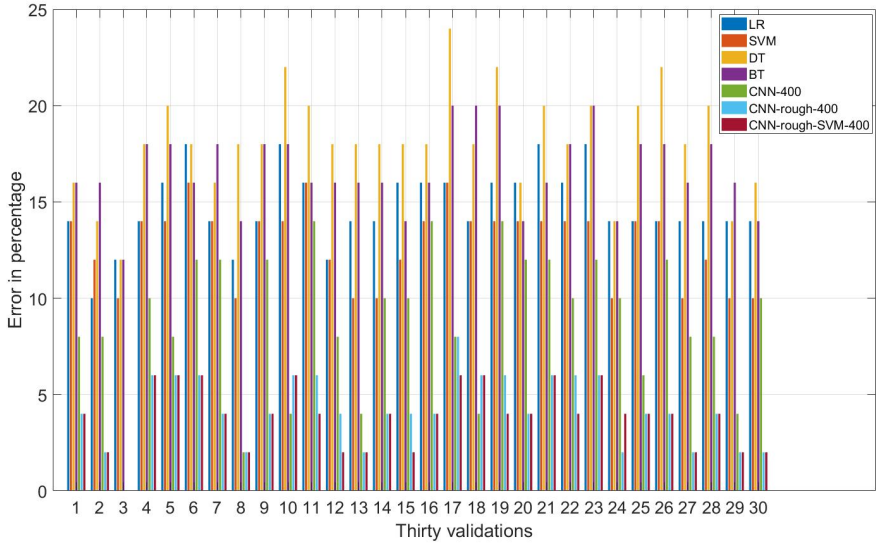
C. Analytical discussion and human judgement

This section discusses why the proposed CNNs are able to obtain better classifications and the proposed detection framework is able to achieve more robust detection. This section also discusses how we set the parameters to tradeoff the performance.

The classification results show that the three CNNs, CNN-FCN, CNN-Hough and CNN-Hough-SVM, are better than the three classical classifiers, LR, SVM, DT and BT. Better results are achieved since the three CNNs uses the image features which are generated from the original image patterns by the pooling and convolution layers. However, the three classical classifiers only use the CHT values as the image features. These results indicates that solely using the CHT values is not enough since the CHT values cannot fully cover all image features for classifications. CNNs can obtain higher classification rates, since image features from the original image patterns are used.



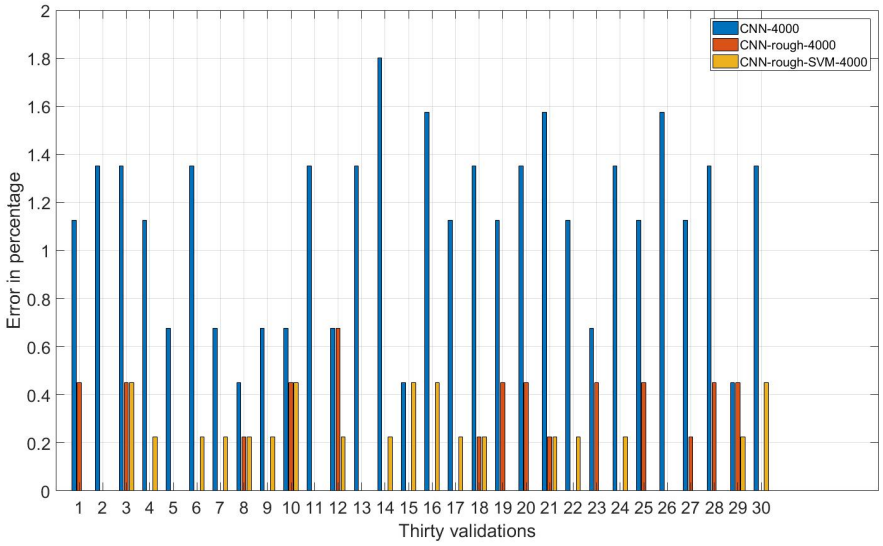
(a) Training errors



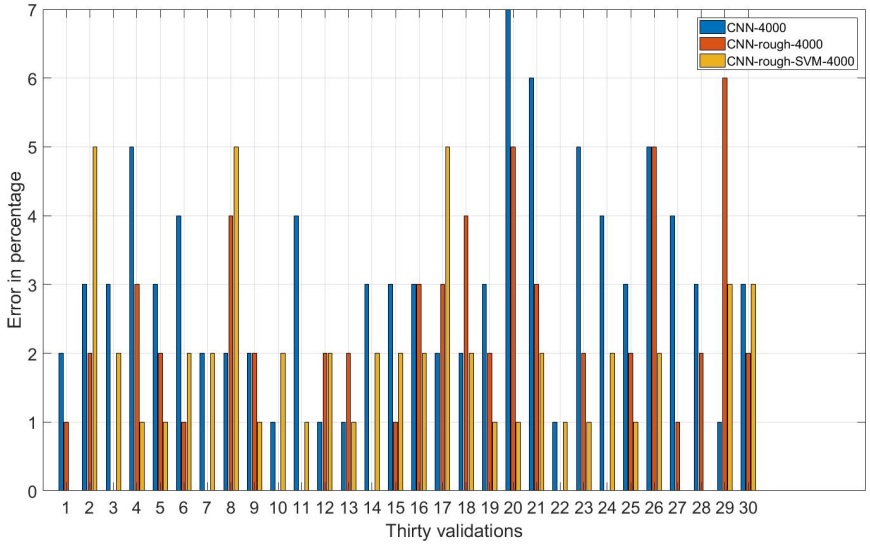
(b) Test errors

Figure 9: Training and test errors for 400 epochs

Among the three CNNs, the proposed CNN-Hough and CNN-Hough-SVM are significantly better than the classical CNN-FCN. The mechanism of the proposed CNNs is similar to CNN-FCN except that different cost functions are used. The mean square error (4) is used by the CNN-FCN and the proposed CNNs uses (5) which is engaged with the penalty. Potentially wrong classifications are penalized in (5) when image



(a) Training errors



(b) Test errors

Figure 10: Training and test errors for 4000 epochs

patterns are similar but their classifications are different. Fig. 8 shows the correlations for all collected 494 image patterns. Darkness indicates the correlations. Bonding ball patterns with higher darkness are indexed from 1 to 384; no bonding patterns with lower darkness are indexed from 385 to 494. Based on the correlations, penalties are forced on the cost function when potentially wrong classifications are generated. This

explains why better generalization can be performed by the proposed CNNs.

For the two proposed CNNs, the configurations of CNN-Hough and CNN-Hough-SVM are the same except the last layer. Fully connected neural network is used in the CNN-Hough and SVM is used in the CNN-Hough-SVM. Based on the measured execution time, the CNN-Hough requires 14.28 μs and the proposed CNN-Hough-SVM requires 68.02 ms. Hence, the CNN-Hough is more recommendable in real-time implementation since the execution time is shorter. Tables III and IV show that, the CNN-Hough-SVM can achieve better results in term of the classification performances than the CNN-Hough. The trade-off between the execution time and classification performance have to be decided which of the two proposed CNNs should be implemented.

In the proposed detection framework in Fig. 4, "$bond\_factor$" is generated by the CNN to determine whether bonds exist. Bonds exist when "$bond\_factor > 0$"; otherwise, a bond does not exist. To ensure the determination is certain, the threshold check is used. If "$bond\_factor > bond\_threshold$" or "$|bond\_factor| > unbond\_threshold$", the detection is certain. Both "$bond\_threshold$" and "$unbond\_threshold$" are defined based on (10), where γ adjusts the thresholds. Here γ is set as 0.8 such that 20% of the training data above the hyperplane is considered to be certain. In the proposed detection framework, our records show that 15 samples among all the 100 test samples are uncertain. These 15 uncertain samples are detected based on the human judgement and the other 85 samples are detected by the CNN. Hence, 85 detections are performed without human. When γ is set to high value, the thresholds are high and more samples detected by the CNN are considered as uncertain. Although detection robustness of the proposed framework is higher, more human judgements are required. Hence this is a tradeoff to select γ.

V. Conclusion and future work

In this paper, a framework involving machine learning and human judgements was proposed for ball bonding detections. Machine learning integrated with CNN, SVM and CHT, is used to detect in the front line; the detection can be finally outcome if the detection uncertainty is above the threshold. Otherwise, human judgement is used when the detection uncertainty is below the threshold; hence significant numbers of

human judgements can be saved, compared to the current manufacturing inspection which is fully relied on human judgements.

To improve the classification rates, two strategies were proposed. First, a novel training function was proposed in the CNN which is the backbone of the detection framework. Unlike the classical CNNs which optimize the loss function involving only the classification rates, the proposed cost function was integrated with a penalty term which takes into account similarities between image patterns. When similar features are inspected by the CNN but different outcomes are generated, the CNN is penalized through the training. The similarities are measured by the CHT features which are commonly used for ball bonding inspections. Hence, more information is used to train the CNN. Second, the CNN configuration was redeveloped by integrating with the SVM. The last layer of the proposed CNN is replaced with the SVM, since the SVM is more effective effective for binary classifications when comparing to the fully connected neural network which is originally used in CNNs.

To evaluate the performance of the proposed detection framework, image data was collected from ball bonding machines. We have compared the classification results obtained by the proposed framework with the classical CNN and three commonly used classifiers, SVM, logistic regression and decision tree. Experimental results showed that the proposed framework was able to achieve significant better classification rates. Better results can be explained by three reasons: **1)** more image features were used in last layer of the proposed CNN backbone to perform classification; the commonly used classifiers only use a few CHT features and some important image features are not used. **2)** the classical CNN only used the convolved images to perform classification and the CHT features may not be included; the proposed framework used both the images and CHT features to perform classifications. **3)** a SVM with better generalization capabilities is used in the proposed framework; the classical CNN only uses the fully connected networks which perform poorer in binary classification than the SVM used in the proposed CNN backbone. Therefore, better classification results are achieved by the proposed framework. Also the experimental results show that only the machine learning determinations below the predefined threshold are necessary to be reassessed by human judgements when the proposed detection framework is used. Our experimental results showed that 15 samples among all the 100 test samples were below the predefined

threshold. These 15 uncertain samples were required to be reassessed manually based on the human judgement. Hence 85% of detections are automatically conducted by the proposed framework

CONFLICT OF INTEREST: AUTHOR, KIT YAN CHAN, DECLARES THAT HE HAS NO CONFLICT OF INTEREST; AUTHOR, KA FAI CEDRIC YIU, DECLARES THAT HE HAS NO CONFLICT OF INTEREST; AUTHOR, HAK-KEUNG LAM, DECLARES THAT HE HAS NO CONFLICT OF INTEREST; AUTHOR, BERT WEI WONG, DECLARES THAT HE HAS NO CONFLICT OF INTEREST.

ETHICAL APPROVAL: THIS ARTICLE DOES NOT CONTAIN ANY STUDIES WITH HUMAN PARTICIPANTS PERFORMED BY ANY OF THE AUTHORS.

FUNDING: THIS STUDY IS NOT FUNDED BY ANY ORGANIZATIONS.

References

[1] C. Bishop, Advanced Wire Bonding Technology: Materials, Methods, and Testing. Springer, 2006.

[2] Bruker, "Assisting the Transition to Cu Wire Bonding: Bruker Nano Surfaces ContourGT 3D Optical Microscopes," Bruker Corporation, Tech. Rep., August 2012.

[3] D. Perng, S. Lee, and C. Chou, "Automated bonding position inspection on multi-layered wire ic using machine vision," International Journal of Production Research, vol. 48, no. 23, pp. 6977–7001, 2017.

[4] K. Ngan and S. Kang, "Automated inspection of ic bonding wires using hough transform," IEEE Transactions on Components, Hybrids and Manufacturing Technology, pp. 333–338, 1988.

[5] A. Khotanzad, H. Banerjee, and M. Srinat, "A vision system for inspection of ball bonds and 2-d profile of bonding wires in integrated circuits," IEEE Transactions on Semiconductor Manufacturing, vol. 7, no. 4, pp. 413–422, 1994.

[6] K. Sreenivasan, M. Srinath, and K. A, "Automated vision system for inspection of ic pads and bonds," IEEE Transactions on Components, Hybrids and Manufacturing Technology, vol. 16, no. 3, pp. 333–338, 1993.

[7] H. Wu, X. Zhang, H. Xie, Y. Kuang, and G. Ouyang, "Classification of solder joint using feature selection based on bayes and support vector machine," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 3, no. 3, pp. 516–522, 2013.

[8] N. Cai, Y. Zhou, Q. Ye, G. Liu, H. Wang, and X. Chen, "IC solder joint inspection via robust principle component analysis," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 7, no. 2, pp. 300–309, 2017.

[9] J. Jiang, J. Cheng, and D. Tao, "Color biological features-based solder paste defects detection and classification on printed circuit boards," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 2, no. 9, pp. 1536–1544, 2012.

[10] C. Benedek, O. Krammer, M. Janóczki, and L. Jakab, "Solder paste scooping detection by multilevel visual inspection of printed circuit boards," IEEE Transactions on Industrial Electronics, vol. 60, no. 6, pp. 2318–2331, 2013.

[11] H. Yang, S. Zheng, J. Lu, and Z. Yin, "Polygon-invariant generalized hough transform for high-speed vision-based positioning," IEEE Transactions on Automation Science and Engineering, vol. 13, no. 3, pp. 1367–1384, 2016.

[12] K. Choi, Y. Lee, J. Moon, C. Park, and F. Harashima, "Development of an automatic stencil inspection system using modified hough transform and fuzzy logic," IEEE Transactions on Industrial Electronics, vol. 54, no. 1, pp. 604–611, 2007.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural network," in Proceedings of the Conference of Neural Information Processing Systems, 2012, pp. 1106–1114.

[14] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," CIRP Annals - Manufacturing Technology, vol. 65, pp. 417–420, 2016.

[15] N. Cai, G. Cen, J. Wu, F. Li, H. Wang, and X. Chen, "Smt solder joint inspection via a novel cascaded convolutional neural network," IEEE Transactions on Components, Packaging and Manufacturing Technology, pp. 1–8, 2018.

[16] P. Dolata, M. Mrzygłód, and J. Reiner, "Double-stream convolutional neural networks for machine vision inspection of natural products," Applied Artificial Intelligence, vol. 31, no. 7-8, pp. 643–659, 2017.

[17] J. Feng, F. Li, S. Lu, J. Liu, and D. Ma, "Injurious or noninjurious defect identification from mfl images in pipeline inspection using convolutional neural network," IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 7, pp. 1883–1892, 2017.

[18] S. Yang, W. Wang, C. Liu, and W. Deng, "Scene understanding in deep learning-based end-to-end controllers for autonomous vehicles," IEEE Transactions on Systems, Man and Cybernetics: Systems, 2018.

[19] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and C. Loupos, "Deep convolutional neural networks for efficient vision based tunnel inspection," in IEEE International Conference on Intelligent Computer Communication and Processing, 2015.

[20] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen, and D. Feng, "Deep convolutional neural networks for human action recognition using depth maps and postures," IEEE Transactions on Systems, Man and Cybernetics: Systems, 2018.

[21] B. Pourbabaee, M. Roshtkhari, and K. Khorasani, "Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients," IEEE Transactions on Systems, Man and Cybernetics: Systems, vol. 48, no. 12, pp. 2095–2104, 2018.

[22] X. Niu and C. Suen, "A novel hybrid cnn-svm classifier for recognizing handwritten digits," Pattern Recognition, vol. 4, no. 1, pp. 1318–1325, 2012.

[23] S. Wagner, "Sar atr by combination of convolutional neural network and support vector machines," IEEE Transactions on Aerospace and Electronic Systems, vol. 20, no. 6, pp. 2861–2872, 2016.

[24] S. Suzuki, N. Iida, H. Shouno, and S. Kido, "Architecture design of deep convolutional neural network for diffuse lung disease using representation separation information," in Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications. IEEE, 2017, pp. 1318–1325.

[25] D. Perng, C. Chou, and S. Lee, "Design and development of a new machine vision wire bonding inspection system," The International Journal of Advanced Manufacturing Technology, vol. 34, pp. 323–334, 2017.

[26] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, pp. 273–297, 1995.

[27] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's Thesis, Technical

University of Denmark, Informatics and Mathematical Modelling, Building 321, DK-2800 Kongens Lyngby, Denmark, 2012.

[28] D. Hosmer, S. Lemeshow, and R. Sturdivant, Applied logistic regression. John Wiley and Sons, 2013.

[29] B. Boser, I. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in Proceedings of the fifth annual workshop on Computational learning theory. IEEE, 1992, pp. 144–152.

[30] J. Quinlan, C4. 5: Programming for machine learning. Morgan Kauffmann, 1993.

[31] M. P. Perrone and L. N. Cooper, "When networks disagree: ensemble methods for hybrid neural networks," Brown University, Tech. Rep., December 1992.

Table II: Analysis of training errors

Classification methods		LR	SVM	DT	BT	CNN-FCN-400	CNN-Hough-400	CNN-Hough-SVM-400	CNN-FCN-4000	CNN-Hough-4000	CNN-Hough-SVM-4000
Training errors	**Mean**	15.9355	16.7004	6.9186	7.4212	2.7816	3.8857	2.5075	2.2766	1.1134	1.4896
Improvement in (% when using CNN-Hough-4000)		99.6342	122.8591	32.2125	37.9714	19.5362	58.9881	30.0790	32.1339	Nil	16.8881
Improvement in (% when using CNN-Hough-SVM-4000)		90.6523	91.0804	78.4697	79.9278	46.4488	61.6649	40.5948	34.5697	-33.7907	Nil
T values (to CNN-Hough-4000)		40.3382	45.5527	16.0671	18.1108	5.2613	8.0605	4.5444	14.5216	Nil	7.4649
T values (to CNN-Hough-SVM-4000)		39.4235	44.5957	15.0690	17.0833	4.0900	6.9888	3.3313	10.4519	-7.4649	Nil

Table III: Analysis of test errors

Classification methods		LR	SVM	DT	BT	CNN-FCN-400	CNN-Hough-400	CNN-Hough-SVM-400	CNN-FCN-4000	CNN-Hough-4000	CNN-Hough-SVM-4000
Test errors	**Mean**	14.8765	12.6869	18.0216	16.6121	8.5393	4.6999	4.6349	3.6200	2.4735	2.2277
	Variance	3.7195	4.0506	7.1540	4.2299	13.4299	3.4069	2.7402	2.8517	2.6897	1.7299
Improvements (in % when using CNN-Hough-4000)		83.3733	80.5037	86.2750	85.1104	71.0342	47.3713	46.6341	31.6714	Nil	-11.0319
Improvements (in % when using CNN-Hough-SVM-4000)		85.0253	82.4408	87.6387	86.5898	73.9122	52.6004	51.9365	38.4604	9.9358	Nil
T values (to CNN-Hough-4000)		26.8341	21.5472	27.1432	29.4393	8.2751	4.9387	5.0806	2.6676	Nil	-0.6403
T values (to CNN-Hough-SVM-4000)		29.6781	23.8273	29.0234	32.2728	8.8788	5.9743	6.2362	3.5626	0.6403	Nil

Table IV: Analysis of recalls, precisions and F-measures

Classification methods		LR	SVM	DT	BT	CNN-FCN-4000	CNN-Hough-4000	CNN-Hough-SVM-4000
Precision	**Mean**	84.9892	87.4350	81.9035	83.2398	96.2823	97.3781	97.5992
	Variance	6.1583	8.2817	6.5744	6.5794	6.0393	4.7213	4.3988
	T values (to CNN-Hough-4000)	20.5724	15.1029	25.2187	23.0359	1.8296	Nil	-0.4011
	T values (to CNN-Hough-SVM-4000)	21.2570	15.6339	25.9521	23.7373	2.2325	0.4011	Nil
Recall	**Mean**	94.8035	95.4068	93.4495	94.1112	98.9039	99.3398	99.4504
	Variance	0.7682	0.8374	0.8285	0.7793	0.8319	0.5783	0.4988
	T values (to CNN-Hough-4000)	21.4118	18.1048	27.2004	24.5787	2.0103	Nil	0.5838
	T values (to CNN-Hough-SVM-4000)	22.6112	19.1594	28.5286	25.8670	2.5947	-0.5838	Nil
F-measure	**Mean**	89.6285	91.2471	87.2964	88.3423	97.5755	98.3491	98.5161
	Variance	1.3661	1.5211	1.4716	1.3936	1.4623	1.0303	0.8960
	T values (to CNN-Hough-4000)	30.8553	24.3530	38.2726	35.2046	2.6839	Nil	0.6589
	T values (to CNN-Hough-SVM-4000)	32.3661	25.6087	39.9376	36.8269	3.3547	-0.6589	Nil
Type I error	**Mean**	15.0108	12.5650	18.0965	16.7602	3.7177	2.6219	2.4008
	Variance	0.0616	0.0828	0.0657	0.0658	0.0604	0.0472	0.0440
	T values (to CNN-Hough-4000)	205.7243	151.0295	252.1865	230.3589	18.2959	Nil	-4.0105
	T values (to CNN-Hough-SVM-4000)	212.5698	156.3387	259.5209	237.3726	22.3251	4.0105	Nil
Type II error	**Mean**	14.4605	13.0644	17.7898	16.1531	3.3172	2.0135	1.6914
	Variance	0.0701	0.0733	0.0652	0.0660	0.0757	0.0537	0.0473
	T values (to CNN-Hough-4000)	193.7234	169.8336	250.5915	223.8344	19.8519	Nil	-5.5501
	T values (to CNN-Hough-SVM-4000)	204.0570	179.3415	262.8502	235.2829	25.3895	5.5501	Nil

Table V: Analysis of classification methods.

Classifiers	Analysis	Parameters
SVM	• 5 weights due to 5 features • 1 bias for the SVM model • hence 6 parameters are required	6
Logistic regression (LR)	• 5 weights due to 5 features • 1 bias for the regression model • hence 6 parameters are required	6
CNN-FCN-4000	• 4596=300+200+4096 = (parameters in the first layer) + (parameters in the second layer) + (parameters in the fully connected network) • parameters in the first layer = 300 (i.e. = $5 \times 5 \times 12$), since the kernel size is 5×5 and the number of kernels is 12 • parameters in the second layer = 200 (i.e. = $5 \times 5 \times 8$), since the kernel size is 5×5 and the number of kernels is 8. • parameters in the fully connected network = 4096 (i.e. = $16 \times 128 \times 2$), since the number of input nodes is 16, the number of hidden nodes is 128, and the number of output nodes is 2.	4596
CNN-Hough-4000 / CNN-Hough-SVM-4000	• 506=300+200+17 = (parameters in the first layer) + (parameters in the second layer) + (parameters in the SVM) • parameters in the first layer = 300 (i.e. = $5 \times 5 \times 12$), since the kernel size is 5×5 and the number of kernels is 12 • parameters in the second layer = 200 (i.e. = $5 \times 5 \times 8$), since the kernel size is 5×5 and the number of kernels is 8. • parameters in the SVM = 17 (i.e. = 16+1), since the number of inputs to the SVM is 16 and the SVM has one bias.	517