Detecting Modifications in Printed Circuit Boards from Fuel Pump Controllers

Thomas Jose Mazon de Oliveira, Marco Aurelio Wehrmeister, Bogdan Tomoyuki Nassu
Federal University of Technology — Parana
Curitiba, Brazil
e-mail: thomasoliveira@alunos.utfpr.edu.br, wehrmeister@utfpr.edu.br, btnassu@utfpr.edu.br

***Abstract*— Frauds involving illegal modifications to the printed circuit boards from fuel pump controllers are a serious problem, which not only harms customers, but also connects to other crimes, such as money laundering and tax evasion. The current state-of-practice for inspecting these boards is a visual analysis performed by a human. In this paper, we introduce an image-based approach that can provide support to the human inspector by automatically detecting suspicious regions in the boards. The proposed approach aligns a photograph of the inspected board to a reference view, partitions the image in sub-regions, extracts features using a variation of the popular Scale-Invariant Feature Transform, classifies the features against previously trained Support Vector Machines, and integrates the results for presentation. In experiments performed on a dataset containing 649 images from a board, with and without modifications, our approach achieved a precision of 0.7739, a recall of 0.9434, and an F-measure 0.8503. These results indicate that our approach can effectively identify suspicious regions, providing invaluable help to the human inspector.**

I. Introduction

Modern society is highly dependent on gasoline and oil-based transportation vehicles. A gas station can be seen as a worthwhile and lucrative business. Moreover, in Brazil, there are many taxes on the fuel businesses (production, distribution, selling, and others), making it a great source of income for the government. This scenario opens room for many sorts of frauds. The first kind of fraud that usually comes to peoples' minds is chemical adulteration, e.g. adding ethanol to gasoline, which increases fuel volume at a fraction of the cost. However, chemical adulterations can be easily discovered, and hence, criminals create other ways to deceive their victims.

Currently, the most common type of fraud encountered in Brazilian gas stations is volumetric. Volumetric frauds are presented in two variations: (i) to buy fuel without invoice, i.e. the selling operations are invisible to the government taxation departments; and (ii) to charge for a wrongful fuel volume, i.e. during fueling, the gas pump registers a higher amount in comparison with the actual fuel volume filled in the vehicle's tank. Criminals adopt the latter strategy because this volumetric fraud is more difficult to be discovered. It is also more lucrative, since the gas station receives the payment proportional to the wrongful volume sold, and the government is not able to issue the taxation properly.

In both cases, the criminals must hide the difference in the registered fuel volume. They often use means to fake the values (related to the fuel volume expelled by the gas pump) that are registered and informed to both the customer and the government. During regular inspections, Brazilian inspectors from a government agency have discovered illegal modifications on the printed circuit boards (PCBs) that control gas pumps. These modifications include substitution of components in order to add functionalities, addition of new components in order to modify the original system's behavior, and even the replacement of the whole board by an uncertified (and illegal) clone board.

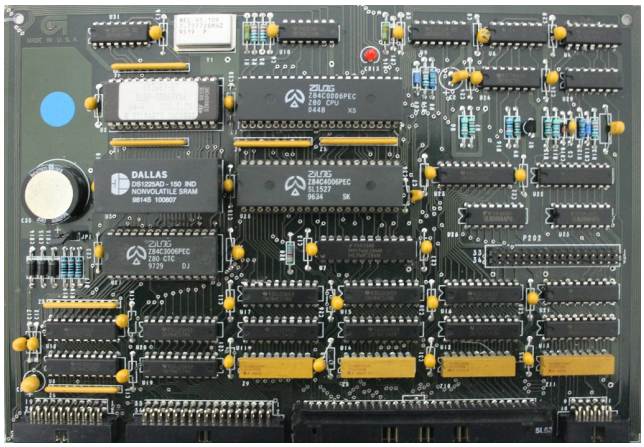
The current state-of-practice for inspection is simply a visual analysis performed by a human. He/she inspects the board in order to find suspicious elements onto the printed circuit board. Suspicious boards are then compared with a sample of the original board at an official analysis laboratory. Once the inspectors determine that the board has any illegal modification, legal action is taken. Figure 1 shows an example of modification (zoomed in in Fig. 2). As one can see, it is very difficult to identify illegal adulterations due to the large amount of small components in the PCB.

In this paper, we introduce an image-based approach for automatically detecting modifications in PCBs from fuel pump controllers. Our final aim is developing a tool that can be used to support the work of a human inspector, allowing her to focus her attention in suspicious regions of the board. That aim has guided some design choices we made, regarding the type of image capture equipment that can be expected (which is different, for example, from what would be reasonable in a production line setting), as well as the fact that false detections are more acceptable than not detecting some modification. Although we are dealing with PCBs from fuel pump controllers, the work is also applicable to other similar scenarios.

The proposed approach follows the steps shown in Fig. 3. The input image is aligned to a reference that shows a board of the same model from an ideal position. This is done using the popular SIFT method [1] and a standard image registration algorithm [2]. The registered image is partitioned into smaller sub-regions, and a feature descriptor is extracted from each sub-region, using a variation of SIFT [1]. These descriptors are then classified by Support Vector Machines (SVM) [3], which were previously trained to detect features that differ too much from those found in an unmodified board. The results can then be integrated and presented to the human inspector.

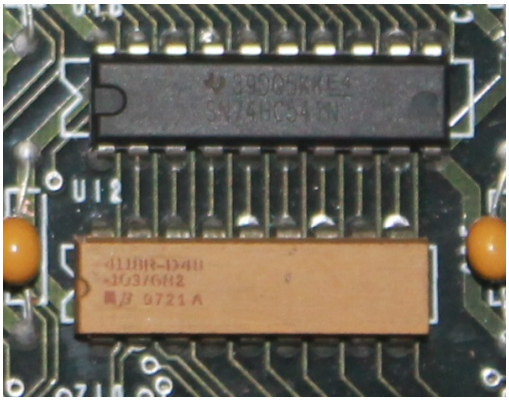
The proposed approach was tested on a dataset containing 572 photographs of an unmodified board, as well as 77

(a) Original PCB.

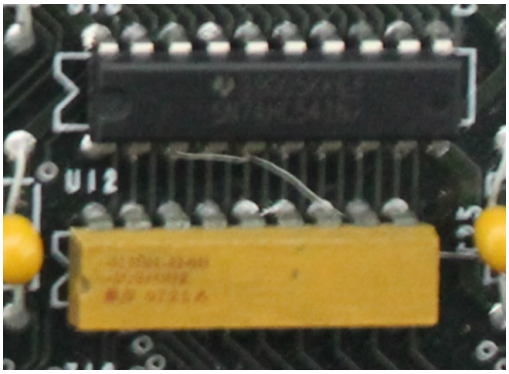


(b) Modified PCB.

Fig. 1. An original and a modified PCB. There is a thin wire connecting hidden components near the bottom right corner, but locating it is challenging for a human, given the amount of detail in the images. Note how the color of the components also changes, due to differing lighting conditions when the pictures were taken. A closer view of the modified area is shown in Fig. 2



(a) Original PCB (detail).



(b) Modified PCB (detail).

Fig. 2. A closer view of the modified region from Fig. 1

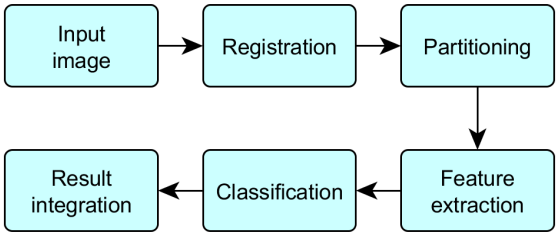


Fig. 3. Steps of the proposed approach.

photographs of the same board, after being modified. The photographs were taken under conditions similar to those found in a practical situation, where the human inspector uses a handheld camera, without additional equipment. Experiments using cross-validation produced a precision of 0.7200 and a recall of 0.9782 when we favor a high detection rate, and a precision of 0.7739, a recall of 0.9434 when we try to maximize the F-measure. Although not high enough for a completely automated solution, these values indicate that our approach can provide invaluable help to the human inspector, pointing at suspicious regions that can be further analyzed.

The rest of this paper is organized as follows. In Sec. II, we discuss related work. Section III describes the proposed approach in depth. In Sec. IV, we describe the experiments performed to evaluate our approach and the obtained results. Section V concludes the paper and points to future work.

II. Related Work

We have performed an extensive and comprehensive review on the current state-of-the-art and state-of-the-practice for printed circuit board inspection. We have found many algorithms for PCB inspection, but only a few of them focus on finding modifications on the assembled board. This section discusses some of these works.

Ardhy *et al.* [4] propose a system that uses low cost devices (e.g. Raspberry Pi and Arduino) to detect imperfections on PCBs. For that, the system compares an image of the board being analyzed with a reference image. They analyze grayscale images, which are median filtered and binarized using adaptive Gaussian thresholding. The algorithm then extracts edges using the Sobel operator, to reduce the impact of lighting

variations. The final analysis is based on a simple subtraction between the processed image and the reference. Their algorithm was able to detect complex borders and missing components, but they have a major drawback in that the images must be obtained from the same viewpoint, with very similar positioning and lighting — even a slight displacement may lead to a number of false negatives.

Xie *et al.* [5] present a visual inspection software that is able to automatically detect defects on an assembled PCB. The proposed method is based on genetic programming and does not require previous knowledge about the board layout, lighting conditions, or visual features from the components. The authors claim their software was able to identify 100% of the defects, while also highlighting some suspicious areas of the board that could present defects. The same way as in [4], the approach from Xie *et al.* [5] uses a reference image of the board without defects, and compares it with images of the boards being analyzed. These images must be aligned, but small alignment differences can be tolerated. One disadvantage of their approach is the excess of information necessary to train the system. When using the complete image for the genetic programming training phase, the amount of components, as well as the size and distance between components, generate an enormous amount of information that leads to a long training time.

Bhardwaj [6] proposes an algorithm named Automated Optical Inspection (AOI) that detects defects on the components' holes during the manufacturing process of PCBs. The proposed method includes several steps: low-pass Gaussian filtering, binarization for background separation, and morphological operators to remove undesired image parts. Particle-based analysis is performed on the resulting image in order to detect circles and measure the holes. Results are given in terms of the radius and the position of the holes in the board. This work presents two limitations: (i) the algorithm can identify only defects on the components' holes, not defects associated with soldering or the board components; (ii) the approach works only on boards without any mounted component, i.e. it is useless with assembled printed circuit boards.

An inspection system based on pattern matching and morphological operators is presented in [7]. The proposed method comprises a preprocessing phase that includes several filters; histogram-based segmentation, morphology-based skeletonization, descriptor extraction, and classification by an artificial neural network. The neural network is able to classify an image in four categories: (i) correct component; (ii) missing component; (iii) displaced component; (iv) inverted component. A drawback of this approach is that the system requires segmentation information for each component as input. This is done manually, leading to a high cost when the number of distinct components is large. In addition, the authors suggest using cross-correlation in the first stage in order to decrease the amount of false negatives. However, this technique is sensitive to lighting variations as well as to scratches on the surface of components.

III. Proposed Approach

In this section, we describe the proposed approach for detecting modifications in PCBs from fuel pump controllers. Our approach follows the steps previously depicted in Fig. 3. The following subsections detail each of these steps.

A. Input Images

Our approach does not place strict demands on how input images must be captured, but they must show the PCB being analyzed from an overhead view. Some degree of perspective distortion, noise, and lighting variations are expected, but higher resolutions and better lighting conditions will likely lead to better results. In smaller resolutions, certain PCB modifications may become indistinguishable from noise; and unfavorable lighting may create shadows and reflections that are detected as modifications. In our experiments, we used 8-bit grayscale images, with 4272×2878 pixels.

B. Image Registration

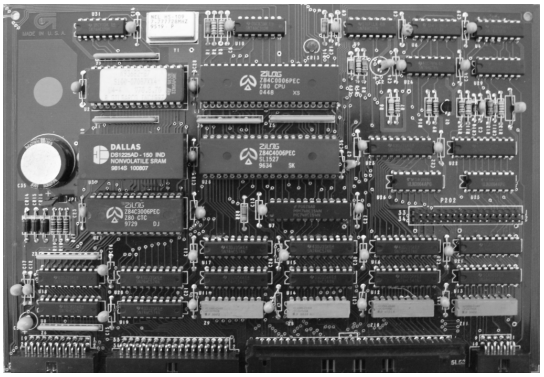
The proposed approach begins by aligning the input image to a reference view of the target circuit board. This reference is an image showing the whole board, captured under good conditions, and with minimal perspective distortion. A different reference must be used for each target PCB model.

Image registration is performed using a standard feature-based method [2]. Sparse sets of image features are detected and described using the Scale-Invariant Feature Transform (SIFT) [1]. Correspondences are located between features extracted from the input and the reference images based on the Euclidean distance in the SIFT feature space. The located correpondences are then filtered using the Random Sampling Consensus (RANSAC) algorithm [8].

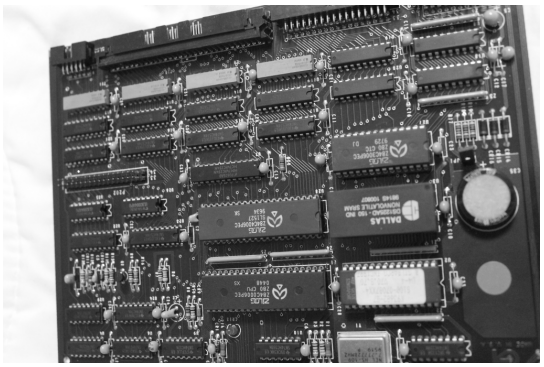
RANSAC produces a matrix that describes a homography — a plane-to-plane projective transformation [9]. That allows us to put the PCB observed image in approximately the same perspective as the reference view, as shown in Fig. 4. However, two limitations of this method must be highlighted. The first is that, since our distortion model only describes plane-to-plane transformations, components protruding from the board, such as capacitors or thick chips, can have distortions in position and aspect. The second limitation is that, due to these distortions, as well as inherent imprecisions from the feature extraction and matching methods, the alignment is not perfect — components usually appear in slightly different positions, and with slightly different aspects. These limitations were kept in mind when defining the remaining steps — for example, a simple image difference would not work in this scenario for detecting modifications, as the number of differences will be large even for an unmodified board.

C. Image Partitioning

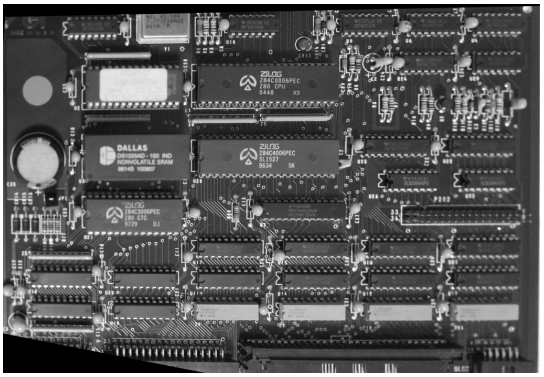
After the initial registration, the input image is partitioned in smaller sub-regions, which are handled independently. This step is needed because PCBs contain a large number of components, and building a single model capable of describing all the necessary information would require high dimensional



(a) Reference image.



(b) Input image.



(c) Registered input image.

Fig. 4. Image registration.

descriptors and classifiers. The smaller sub-regions also make it easier to determine the position of a detected modification.

In our experiments, we take images aligned to a reference with 4096×2816 pixels, and split them into 31×21 sub-regions of 256×256 pixels. The sub-regions are separated by 128×128 pixels, so they overlap, as shown in Fig. 5. The technique chosen for overlapping the sub-regions increase the computation time and required memory, since each pixel from the original image appearing in up to four sub-regions, but the redundant data may improve robusness to variations introduced during image registration or capture.

Depending on the point of view, some sub-regions will have many pixels falling outside the original input image. These

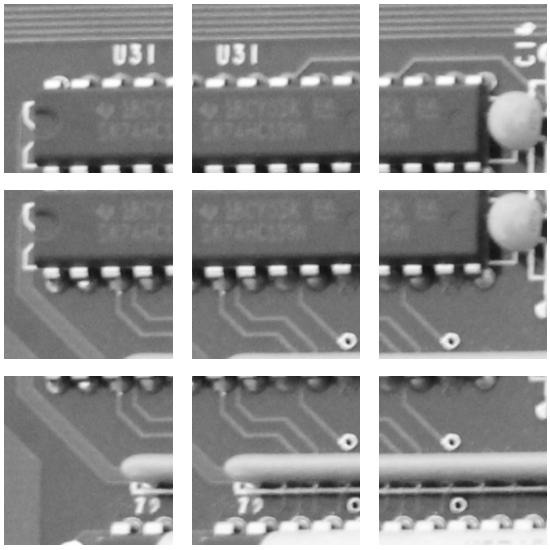


Fig. 5. Example overlapping sub-regions extracted from an image.

sub-regions must be discarded, so to detect modifications in the entire PCB, it must be contained in the image, or multiple images must be analyzed.

D. Feature Extraction

For each sub-region from the input image, we extract a descriptor, which encodes information about the properties of the sub-region. We employ a variation of the SIFT [1] descriptor. First, we compute the gradient magnitude G and orientation θ for each pixel (x, y) inside region f:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{1}$$

$$\theta(x,y) = tan^{-1}(G_y(x,y)/G_x(x,y)) \tag{2}$$

where G_x and G_y are the partial derivatives:

$$G_x(x,y) = f(x+1,y) - f(x-1,y) \tag{3}$$

$$G_y(x,y) = f(x,y+1) - f(x,y-1) \tag{4}$$

The region is then split into a grid of 12x12 blocks — for example, a region with 256×256 pixels will have blocks with 21×21 pixels. A histogram of gradient orientations is computed for each block, with 8 orientation bins per histogram. The magnitude at each pixel position is divided into up to 8 histogram bins using trilinear interpolation. The final descriptor is obtained by concatenating all the values from the histograms from each block, resulting in a vector with 12x12x8 = 1152 dimensions. To make the descriptor more robust to lighting variations, we normalize it to a unit vector, truncate any values above 0.2, and re-normalize it. More details on these steps can be found in the original SIFT paper [1].

Note that, although we compute descriptors as described above, the general idea of the proposed approach is not

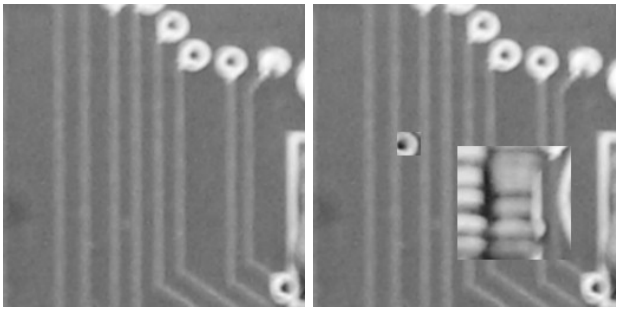


Fig. 6. Left: a negative training sample. Right: artificially generated positive training sample.

dependent on the specific descriptor, so other alternatives could be employed, such as Histograms of Oriented Gradients [10] or Local Binary Patterns [11].

E. Feature Classification

The descriptors extracted from each sub-region are classified as "normal" (negative) or "modified" (positive) by a Support Vector Machine (SVM) [3]. Here, we must divide the pipeline into training and operation phases.

For training, we take a set of images, and follow the same steps described in the previous subsections to obtain a set of training samples (descriptors) for each sub-region. As sub-regions are handled independently, a different SVM will be trained for each sub-region from each PCB model we wish to analyze. One practical limitation faced in the problem domain we are attacking is that obtaining negative examples (normal regions from unmodified PCBs) is easy, but positive examples (modified regions) are rare. More than that, the number of possible modifications is extremely large, so we must prevent the SVM from learning to describe specific modifications — instead, it must describe a normal region, and any sample that is deemed "too different" must be classified as a modification.

With that in mind, we first build the training set using only images from a normal, unmodified board. For each negative training sample, we artificially generate a positive sample, by placing over it small square patches, randomly taken from other regions from other images in the training set. Up to 10 patches may be added to an image, each being 5 to 15 pixels wide. Figure 6 shows one positive sample generated by that approach.

Features are extracted from the training set, and are then used to train an SVM, with positive and negative samples targeting outpus of 1 and -1, respectively. The SVM uses a Radial Basis Function kernel, and its parameters were automatically defined using a grid-based parameter selection algorithm [3].

Once the SVM for a sub-region is trained, it can be used for classifying the descriptor extracted from the same sub-region from an input image in operation time. We do not take as output a binary positive/negative classification, but the distance to the SVM decision margin multiplied by -1 — that results in a "score", so that the higher the score, the more likely the region is modified. We leave to the human inspector the decision about how low the SVM output must be for being taken a sign of a modification.

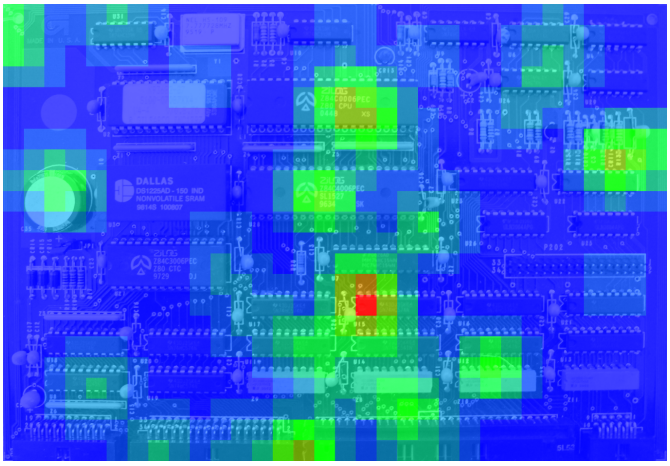


Fig. 7. Final results, encoded as a "heat map" for visualization. "Warmer" colors indicate a higher probability of a modification (best viewed in color).

F. Result Integration

The results obtained for each image sub-region can be combined in a simple manner by setting the output for a pixel to the average, median, or minimum output from all the sub-regions containing the pixel (each pixel may appear in up to four sub-regions). It may also be interesting to produce an output image for visualization. Pixel values can be mapped to different colors, simulating a "heat map" that uses "warmer" colors to indicate a higher probability that a modification is present, as exemplified in Fig. 7.

This output is only created for presenting results to the human inspector, and all the relevant data is already present at an individual sub-region level. For that reason, in Sec. IV we evaluate our approach considering only isolated sub-regions.

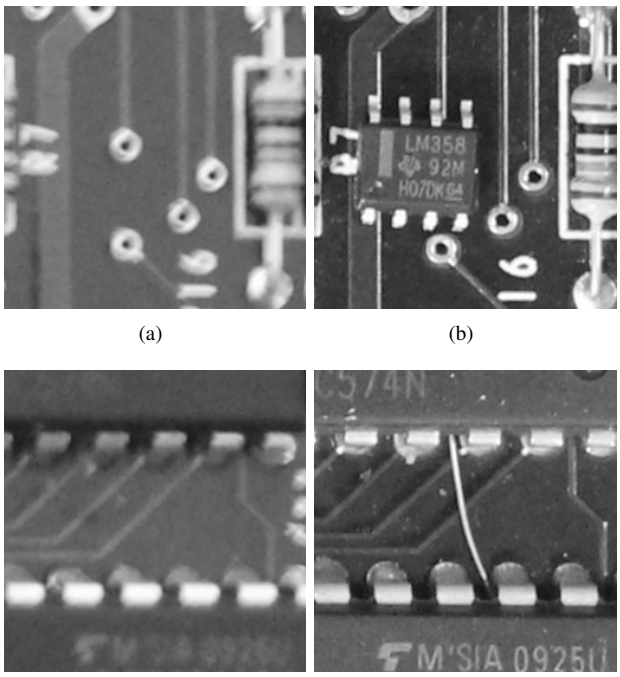
IV. Experiments

The proposed approch was implemented and experimentally evaluated. Implementation is in the Java language, and uses basic functions from the OpenCV[1] library. The following sections detail the tests and the obtained results.

A. Dataset

Our dataset contains 572 images showing an unmodified board, with 4272×2878 pixels. A single board was used, but we must remind that each image sub-region is treated independently — i.e. there are 651 regions showing different parts of the same board. The images were captured using a Canon EOS 1100D camera, with 18-55mm lenses. The same way as would be expected in practice, no other special equipment is required for capturing the images, and the perspective slightly changes between captures. Moreover, lighting conditions were changed during the capture procedure.

[1] www.opencv.org



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

Fig. 8. (a) and (c): unmodified boards. (b): an easily detected modification (an integrated circuit). (d): a more complex case (thin wire connecting hidden components).

In addition to the images from the unmodified board, we have also obtained 77 images showing the board with modifications. These modifications were manually added by the authors, and are meant to be representative of situations commonly encountered in practice. Figure 8 shows some examples of images containing modifications. It is important to note that these examples will not be used for training the SVMs — that way, we know the SVMs will not learn to describe specific modifications.

B. Experiment design

To evaluate our approach, we have first selected 6 image sub-regions in which the manually added modifications are visible. Then, each sub-region was tested using a cross-validation method.

For each test iteration, we divide the negative samples (descriptors extracted from images of the unmodified board) into a training and a test subset. The training subset contains 90% (515) of the samples, which are randomly selected, with the remaining 10% (57) of the samples being used as the test subset. Any samples obtained from images that do not contain the target sub-region are removed from the sets. Positive samples for the training subset are artificially generated as explained in Sec. III. Positive samples for the test subset are extracted from the imges of the modified board. We randomly pick these samples so that the number of positive and negative samples remains the same. The SVM for the region is then trained, and the outputs produced for the test set are saved, along with the expected labels (1 for positive samples, -1 for negative samples). The above procedure was repeated 10 times for each of the selected sub-regions, resulting in 60 test iterations.

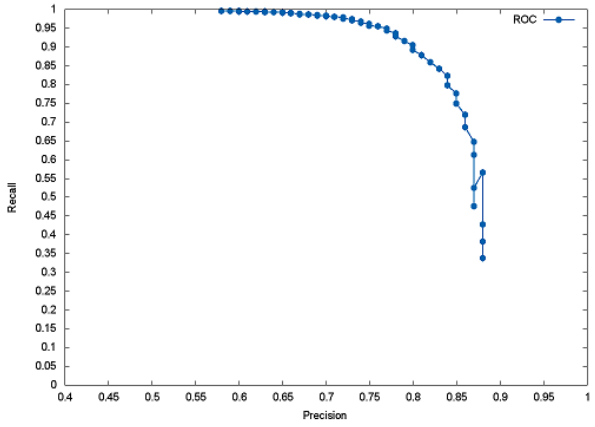


Fig. 9. ROC curve obtained in the experiments.

Having the SVM ouptuts and labels for all the samples from all the test iterations, we define a decision threshold τ. Any output above τ is deemed as *positive*, with the remaining outputs being deemed as *negative*. By comparing the thresholded outputs to the expected labels, we count the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). These values allow us to compute precision (P) and recall (R) metrics, as well as the F-measure:

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{7}$$

We obtain several different values for the above metrics by testing all possible values for τ in the $[-2.6, +2.6]$ interval, with increments of 0.05. Each tested τ can then generate one point in a receiver operating characteristic (ROC) curve. Note that these values are important so that we can objectively evaluate our approach, but in a practical scenario, we can leave to the human inspector any interpretation about the outputs.

C. Results

The graphic in Fig. 9 shows the ROC curve produced by our experiments. The graphic in Fig. 10 shows how the F-measure changes as the decision threshold τ increases. The best F-measure was 0.8503, obtained for $\tau = -0.2500$. The precision was 0.7739, and the recall was 0.9434.

Given the nature of the problem we are attacking, a high detection rate can be desirable, even if that leads to some false positives. The threshold $\tau = -0.7500$ resulted in a precision

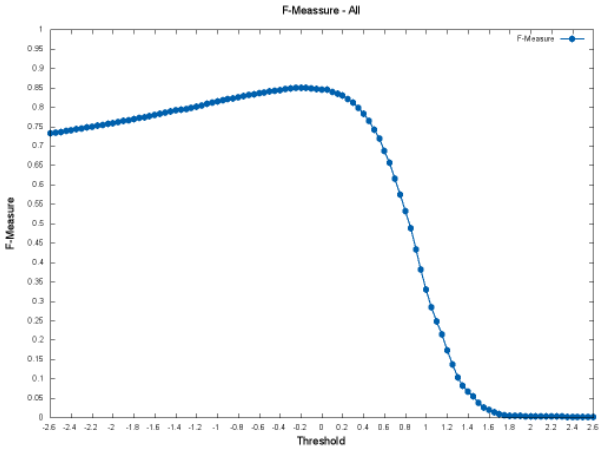


Fig. 10. Values for the F-measure, for different decistion threholds τ.

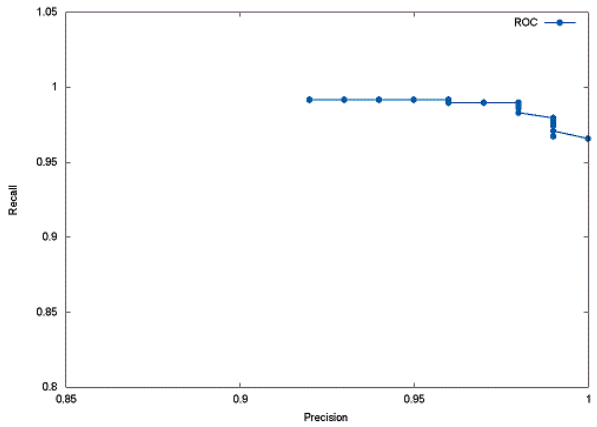


Fig. 11. ROC curve obtained for the region with the best results. Note the scale was adjusted, for better visualization.

of 0.7200, and a recall of 0.9782, which we see as a good compromise.

We have also analyzed the results obtained for each individual region. The graphics in Fig. 11 and Fig. 12 show, respectively, the ROC curve and the F-measure evolution as τ increases for the region with the best results. For that region, we obtained an F-measure of 0.9854, with a precision of 0.9828 and a recall of 0.9879. That region is shown in Fig. 13. It contains an LED protruding from the board, and a chip that takes a considerable portion of the region in the modified board. Even though the LED appearance changes between the images, and even with a variation in aligment, our approach was capable of detecting a chip that was not observed during training.

The graphics in Fig. 14 and Fig. 15 show, respectively, the ROC curve and the F-measure evolution as τ increases for the region with the worst results. For that region, we obtained an F-measure of 0.6843, with a precision of 0.5386 and a recall of 0.9378. That region is shown in Fig. 16.

The results described above were obtained considering 6

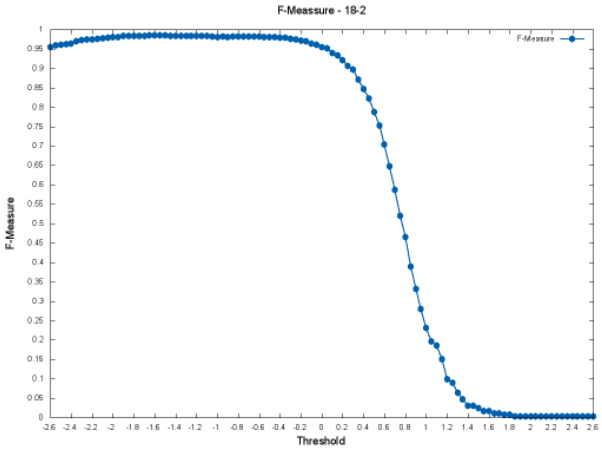
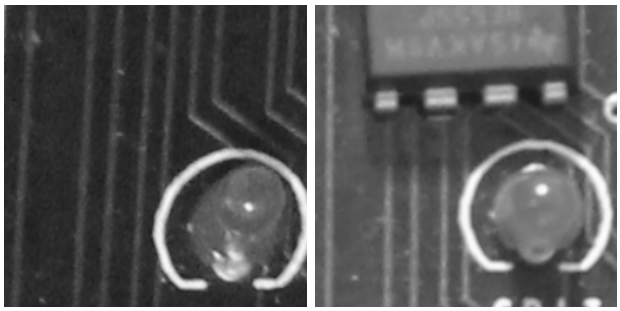


Fig. 12. Values for the F-measure, for different decistion threholds τ, for the region with the best results.



(a) (b)

Fig. 13. Region for which the best results were obtained, with and without modifications. (a) unmodified board. (b) modified board.

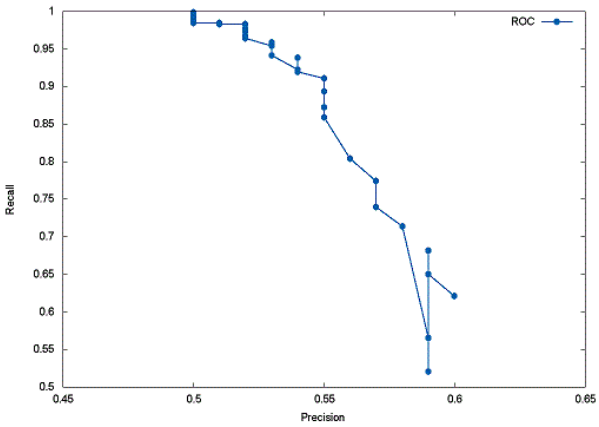


Fig. 14. ROC curve obtained for the region with the worst results.

regions in which the modified board had the modifications clearly visible. In some cases, the modifications may appear at the edges of the region, and even leave the region, due to variations during the registration phase. We performed some tests considering these types of regions and, as expected, there were severe losses in performance. That would be a problem if

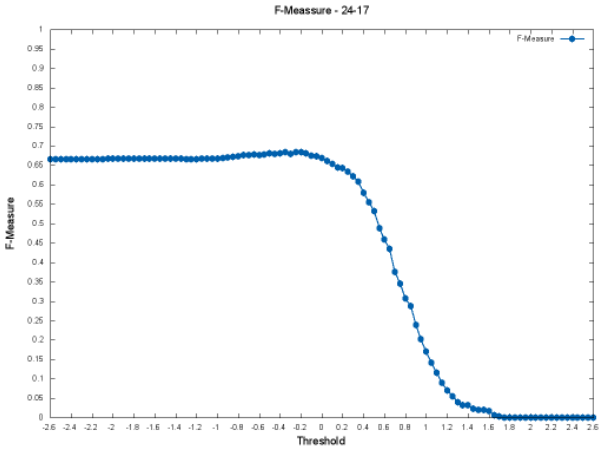
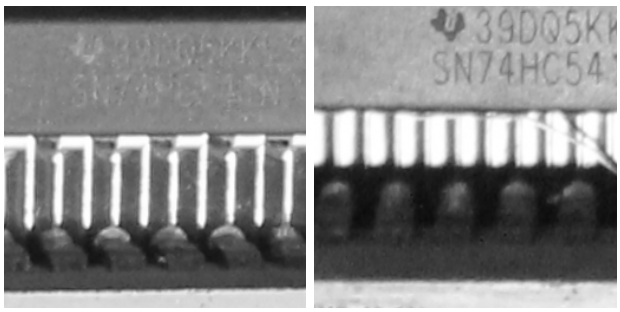


Fig. 15. Values for the F-measure, for different decistion threholds τ, for the region with the worst results.



(a) (b)

Fig. 16. Region for which the worst results were obtained, with and without modifications. (a) unmodified board. (b) modified board.

a modification appeared between two regions. However, since we use an overlapping grid to partition the image, our approach guarantees that each modification will be visible in at least one region, avoiding the problem.

V. Conclusion

We have introduced an approach for detecting modifications in printed circuit boards, keeping in mind the particularities of PCBs used in fuel pump controllers. The proposed approach was tested on a dataset containing 572 photographs from an unmodified board, as well as 77 photographs from a modified board. Experiments using cross-validation produced a precision of 0.7200 and a recall of 0.9782 when we favor a high detection rate, and a precision of 0.7739, a recall of 0.9434 when we try to maximize the F-measure. We consider these promising results, which indicate that our approach can provide invaluable help to the human inspector, pointing at suspicious regions that can be further analyzed.

Future work will focus on improving the obtained results. That includes testing alternative descriptors, such as HOG [10] and LBP [11], as well as different classifiers, and proposing a more "realistic" method for generating positive training samples. We may also replace the descriptor and classifier by a deep convolutional neural network. Combining the results obtained from multiple photographs of the same board is also a possible line for future work.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[2] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[3] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[4] F. Ardhy and F. I. Hariadi, "Development of sbc based machine-vision system for pcb board assembly automatic optical inspection," in *2016 International Symposium on Electronics and Smart Devices (ISESD)*, Nov 2016, pp. 386–393.

[5] F. Xie, A. Uitdenbogerd, and A. Song, "Detecting pcb component placement defects by genetic programming," in *2013 IEEE Congress on Evolutionary Computation*, June 2013, pp. 1138–1145.

[6] S. C. Bhardwaj, "Machine vision algorithm for pcb parameters inspection," *IJCA Proceedings on National Conference on Future Aspects of Artificial intelligence in Industrial Automation 2012*, vol. NCFAAIIA, no. 2, pp. 20–24, May 2012, full text available.

[7] L. zong Lin, L. shan Zhou, J. ding Wan, and Z. qin Qian, "Study of pcb automatic optical inspection system based on mathematical morphology," in *Computer Technology and Development, 2009. ICCTD '09. International Conference on*, vol. 2, Nov 2009, pp. 405–408.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[9] G. Wang, Z. Hu, F. Wu, and H.-T. Tsui, "Single view metrology from scene constraints," *Image and Vision Computing (IVC) - Elsevier*, vol. 23, no. 9, pp. 831–840, 2005.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[11] M. Pietikainen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*, ser. Computational imaging and vision. London: Springer Verlag, 2011.