



A flexible machine vision system for small part inspection based on a hybrid SVM/ANN approach

Keyur D. Joshi¹ · Vedang Chauhan¹ · Brian Surgenor¹

Received: 30 November 2017 / Accepted: 16 July 2018 / Published online: 24 July 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Machine vision inspection systems are often used for part classification applications to confirm that correct parts are available in manufacturing or assembly operations. Support vector machines (SVMs) and artificial neural networks (ANNs) are popular choices for classifiers. These supervised classifiers perform well when developed for specific applications and trained with known class images. Their drawback is that they cannot be easily applied to different applications without extensive retuning. Moreover, for the same application, they do not perform well if there are unknown class images. This paper proposes a novel solution to the above limitations of SVMs and ANNs, with the development of a hybrid approach that combines supervised and semi-supervised layers. To illustrate its performance, the system is applied to three different small part identification and sorting applications: (1) solid plastic gears, (2) clear plastic wire connectors and (3) metallic Indian coins. The ability of the system to work with different applications with minimal tuning and user inputs illustrates its flexibility. The robustness of the system is demonstrated by its ability to reject unknown class images. Four hybrid classification methods were developed and tested: (1) SSVM–USVM, (2) USVM–SSVM, (3) USVM–SANN and (4) SANN–USVM. It was found that SANN–USVM gave the best results with an accuracy of over 95% for all three applications. A software package known as FlexMVS for flexible machine vision system was written to illustrate the hybrid approach that enabled easy execution of the image conditioning, feature extraction and classification steps. The image library and database used in this study is available at <http://my.me.queensu.ca/People/Surgenor/Laboratory/Database.html>.

Keywords Machine vision (MV) · Part classification · Support vector machines · Artificial neural networks · Flexible inspection · Identification · Sorting

Introduction

Traditionally, human operators perform the task of part identification and sorting through manual inspection (Huang and Pan 2015; Malamas et al. 2003). Under ideal conditions, an operator can perform well for inspection speeds of up to 20 parts/min (Schoonahd et al. 2007). However, target production speeds for automated machinery are typically over 200 part/min (Chauhan and Surgenor 2015). Furthermore, human performance drops as operators suffer from fatigue, stress and lack of concentration for tasks conducted over a long period of time. Semi-automatic mechanisms are available that can ease the task for the human operators. For example, the introduction of a rejection mechanism wherein an operator presses

a button on the machine that will reject the part. Regardless, it has been shown that machine vision (MV)-based inspection systems can obtain higher accuracy results at higher production rates than with people (Batchelor 2012). MV inspection systems can help industry gain a competitive advantage in terms of better product quality, higher customer satisfaction and improved productivity.

The automated sorting of parts is a common application of MV. Figure 1 illustrates how the application can be divided into two groups: binary and non-binary. A class in general, is a set that contains entities with similar properties. Most sorting applications are binary in nature, where a part is either accepted (class 1) or rejected (class 2) on the basis of an easily recognizable feature such as size, shape or color. For example, Abdullah et al. (2000) used a binary color based MV system for quality inspection of bakery products. Cao et al. (2015) performed binary sorting of safety belt pins

✉ Keyur D. Joshi
joshikeyurd@gmail.com

¹ Queen's University, Kingston, ON, Canada

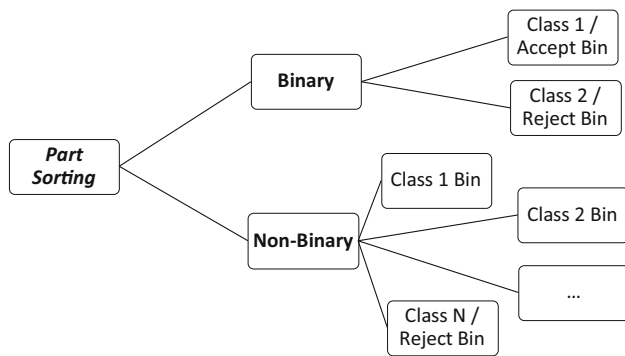


Fig. 1 Two kinds of part sorting, binary and non-binary

using MV. Park (2015) used MV system for binary sorting of semiconductors.

Sorting applications that are not binary in nature are more complicated as they require more effort in feature recognition and classification. The range of applications can be wide. Penaranda et al. (1997) used a color MV system to sort tiles into five different lots where tiles were of similar color and visual appearance. Leemans et al. (2002) graded two types of apples according to their external appearance using MV and sorted them into four different grades. Tessier et al. (2007) employed a MV approach to the automated sorting of five different types of mine ore on conveyor belts, as sorted by composition (soft, medium or hard) and moisture content (dry or wet).

As a more targeted non-binary example that involves the sorting of small parts, Wu et al. (2015) sorted gears into five different categories using a monocular vision technique. It included features such as number of holes, number of teeth and color of the gear. Niklaus and Ulli (2015) dealt with resistors classification. Shen et al. (2012) addressed bearing classification. Nilsback and Zisserman (2006) were able to find the best match for a flower image from a database of other flowers with visual similarity. Other examples of non-binary sorting include Akhtar et al. (2013), Nashat et al. (2011) and Kim et al. (1999) who looked at the sorting of plant leaves, baked biscuits and solder joints, respectively, using various techniques including SVM and two stage (2D and 3D) classifiers.

Figure 2 shows a typical MV-based system for inspection. When a part is in its correct position, one or more cameras are used to acquire the image of the part for processing using a computer equipped with special purpose image processing analysis and classification software. The scene under the camera is well illuminated to highlight a Region of Interest (ROI). Various types and positions of illumination sources are possible and the selection of them is application dependent (Yan and Surgenor 2011). Image acquisition hardware (i.e. camera) conducts the image acquisition and digitization process, while the vision computing device (i.e. computer)

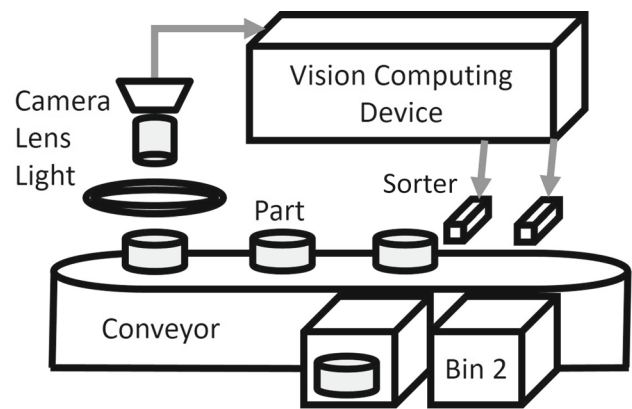


Fig. 2 A typical MV-based system for inspection

enhances and processes images for extracting useful information or template matching. The computer interprets the processed information and generates output signals for a resulting action. The action is typically acceptance or rejection of the part, or if there are multiple types of parts, routing of the part to the appropriate sorting bin.

A Flexible Machine Vision System, ‘FlexMVS’ for object detection and classification was developed, trained and tested for this work. An overview of ‘FlexMVS’ is provided as an appendix to this paper. The main goal of the research was to develop a method that could be applied to various applications with minimum user inputs. Supervised Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) are popular for the task of image classification. They work well with the application for which they are developed. However, in their supervised form, performance degrades when images from an unknown class are introduced. This paper proposes a novel solution to this limitation with ANNs and SVMs by developing a hybrid approach that combines supervised and semi-supervised layers.

The task of image classification is similar to the task of novelty detection. For example, Piementel et al. (2014) used the term *novelty detection* to address the problem of anomaly detection and outlier detection, which is similar to the problem addressed by this paper. They reviewed several novelty detection techniques and grouped them into five categories. The work of this paper falls into their fourth category, which is *domain-based* novelty detection, as it deals with the semi-supervised SVM technique.

Three applications were selected to test the ability of the MV-based system to deal with the unknown class problem: (1) small plastic gears, (2) plastic push-in wire connectors and (3) metallic Indian coins. Four hybrid methods that were based on SVMs and ANNs were developed and applied to the three applications.

An objective measure of the level of difficulty from one application to the next was obtained with a survey of 14 indi-

viduals who had experience in the field of machine vision. More than 84% of respondents agreed that the gear application was the easiest to classify, mainly due to their rotational symmetry, the absence of minor internal patterns and were uniformly solid gray in color. It was further agreed that the connector application was more difficult relative to the gear application, mainly because the connectors were rectangular in shape (rotational asymmetric) and the body was transparent plastic. Finally, it was unanimously agreed that the coin application was the most difficult. The coins had different levels of wear, possessed internal patterns, were rotationally asymmetric due to the internal pattern and were similar in size between denominations.

Related work and FMS

The need for a Flexible Machine Vision (FMV) system, an MV-based system that can be implemented for different applications without extensive retuning or retraining, is a decades old issue (Wilder 1989). The fact that it is still an unresolved issue can be attributed to the complexity of the problem and the observation that multiple MV methods can achieve the same required performance, but usually only after extensive tuning. For example, Modi and Bawa (2012) compared 20 different MV methods for coin recognition and concluded they all worked.

FMV systems that have been developed to date have been only pseudo-flexible, in the sense that they were not tested on different applications, but instead they were tested on different styles of the same part for the same application. As an example, Chetima and Payeur (2012) referred to their approach as ‘automated tuning’ (with MV) and is believed to be the only paper in recent times that set out to automate the “initial tuning of a real-time vision-based inspection system”. But they only applied their binary classification system to tortillas and then retuned and retrained the system for seeded buns of similar geometry and style. As another example that made use of the word ‘adaptive’, Su and Tarn (2008) applied an Adaptive Neuro Fuzzy Inference System (ANFIS) to inspect for surface appearance defects in varistors. There were six classes of defects: back qualified, broken, cracked, front-qualified, printed and dry. The adaptive action was applied to the selection of the type of membership function for the ANN.

Wilder (1989) originally referred to the problem as the need for “adaptive sensing”. Indeed, a number of researchers have used the word “adaptive” in their work. For example, Wang et al. (2017) performed adaptive maximum margin analysis for image recognition. They proposed an adaptive maximum margin analysis for dimensionality reduction that gave the largest margin between different classes. The mathematical model involved calculating a weighting matrix

that could be adaptively calculated by solving the objective function. Schliping et al. (2014) presented an adaptive pattern recognition system that worked in real-time for the video-based analysis of soccer to identify a player’s position. There were five classes of player: outfielder team 1 and 2, goalkeeper team 1 and 2 and referee. They used an adaptive background model for automatic real-time player segmentation. They considered their model robust because experiments were conducted under different weather and ground textures conditions. Li et al. (2013) presented locally adaptive decision functions for person identity verification. For a decision function, they adaptively prepared a local threshold rule. The task was to verify if two images were of the same person. This is similar to a binary classification problem in the sense that the second image would be classified as either the same person (accept) or not the same person (reject). Li and Guo (2013) proposed an adaptive active learning algorithm for image classification with 5 and 10 classes from three public databases: MIT Urban and Natural Scene, Caltech101 and VOC 2007. They set out to select the best weighting parameter from a range of pre-defined values, thereby making it adaptive. They validated their algorithm by comparing it with four different approaches (near optimal, fixed combination, most uncertainty and random sampling) and concluded that their algorithm provided best results.

In spite of the examples in the previous paragraph, the term ‘adaptive’ is not considered appropriate for MV-based applications. In the automatic control system context, ‘adaptive’ refers to a system that continuously adapts to changes in the operating conditions of a process. The term ‘flexible’ is used instead for this work as it is considered more appropriate. ‘Flexible’ in the context of MV-based system is in line with the definition of a flexible manufacturing system (FMS). An FMS is a manufacturing system that can be changed to produce new parts types and/or can change the order of operations performed on a part, without having to make a significant physical changes to the machine (Rosati et al. 2013).

‘Flexible’ systems are changed only once as part of the set-up procedure for a manufacturing process. For example, when a switch is made to a new product type. It is in this context that the phrase flexible machine vision (FMV) is introduced. A system is combination of hardware(s) and/or software(s). Small systems or subsystems combine together and form a bigger system. An FMV system is a subsystem that use generalized hardware and developed software package that can work with different applications. The FMV system is a part of a bigger system, Flexible Manufacturing System (FMS) or Flexible Assembly System (FAS) wherein a change in application can be handled by changing some system inputs. If an MV system is not truly flexible, it reduces the efficiency of the FMS/FAS and will likely lead to a bottleneck subsystem. In the context of FAS, Rosati et al. (2013)

proposed constitutional elements, functioning principles and working cycle of a fully flexible assembly system (F-FAS). They were of the opinion that flexibility can be referred to as the ability to handle a wide variety of part types, conduct model changeovers rapidly and easily, simultaneously process multiple parts/models and quickly respond to part design changes.

Papers can be found on the subject of FMS that use MV as a component in a larger system. For example, Nerakae et al. (2016) integrated an MV system with a robotic system for a pick and place operation that assembled square, triangular and circular parts with various translational and angular positions: above, center, below, 30°, 60° and 90°. They used NI Vision Builder (VB) and LabVIEW NI Vision software. The work involved controlling movement of a robotic arm that used input from the MV system. Hosseininia et al. (2016) introduced flexible automation with MV for a porcelain edge polishing application. Specifically, MV was used to detect the position and orientation of circular and rectangular biscuits (porcelain dishes) so that a robot arm could perform the polishing operation at the correct position and orientation. Tapilouw et al. (2015) developed a white light triangulation sensor for flexible inspection system to measure surface depth profile with accuracy of 1.15 μm .

Weigl et al. (2016) improved performance of surface inspection by online active learning and flexible classifier updates. They proposed active learning as an additional component to the conventional inspection system. This component continuously updated the classifier with the help of user interaction. The user interaction involved re-labelling of samples after the predefined number of samples were classified into predicted classes. The classifier was then re-trained with a combination of samples from the initial training set and the newly labeled samples. Chen and Perng (2016) proposed an automatic inspection system to detect defects on IC molding surface. They achieved 94.2% accuracy rate by using a camera based vision system. Sun et al. (2016) implemented an MV system for inspection to detect four major defects in the manufactured using ANN. The system gave 98.5% accuracy when employed back propagation neural networks.

As will be shown in “Conclusions and future work” section, the methods reviewed above could not achieve the target accuracy of 95% with the presence of unknown images and/or meet the requirement for ease of tuning. It will be shown that the proposed hybrid approach does work with the three different small part applications under consideration.

Coin classification

The Indian coin application in this paper was identified as particularly challenging. Thus, it is considered appropriate to review papers that worked on the problem of coin recognition. For example, Cooray and Fernando (2011) described a

coin counting system that used a webcam to capture pictures of Sri Lankan coins. Fukumi et al. (1992) proposed an ANN pattern recognition system for Japanese coins which was insensitive to rotation of the image. Modi (2011) obtained 81% average recognition rate by using intensity values of 100 pixels as a feature vector input to an ANN for Indian coins. To generate the feature vector, the coin images were shrunk in size to 10×10 pixels. Modi and Bawa (2011) increased the average recognition rate to 98% when an image size of 20×20 pixels was used. The images were rotated from 0° to 360° in 5° increments. Both obverse and reverse side of Indian coins were used. It should be noted that they have used 400 features (i.e. a large number). By using a multi-level counter propagation neural network, Velu et al. (2011) obtained a 99.5% average recognition rate for Indian coins.

The pattern variability within a single class due to wear, rotationally asymmetric patterns and overlapping range of acceptable diameters makes the coin application particularly challenging. Furthermore, the introduction of counterfeit coins increases the level of difficulty still further. In this work, CAN 25 cent coins were introduced as a counterfeits due to their size similarity with the Indian coins. An attempt to use VB on this problem was only able to achieve an accuracy of 79% with a database consisting of good and medium quality coins. Furthermore, when Modi’s method was applied to the problem, the achievable accuracy was only 60% (not 98% as reported in Modi and Bawa 2011), mainly due to the introduction of a realistic database (Joshi et al. 2016). It was then suggested that a Deep Neural Net (DNN) might be able to achieve the target performance of 95% (Bianchini and Scarselli 2014; Schmidhuber 2014). However, DNN was not considered for this application for three reasons: (1) they are not transparent in the sense that finding the cause of the misclassification is difficult, (2) they require a large database that may result in overfitting and (3) they are susceptible to gross errors (Nguyen et al. 2015; Szegedy et al. 2015).

Rationale for the hybrid approach

Upon a review of the literature, it was concluded that theoretically a hybrid SVM/ANN approach that uses both supervised and semi-supervised machine learning algorithms might be able to meet the requirement that the system classify parts into multiple known classes and reject any unknown classes. The developed system must be able to learn all the classes presented in training, but it must also be able to learn to reject classes not covered in training. ANN is most commonly applied as a supervised machine learning algorithm and is unable to deal with unknown classes. SVM is most commonly applied as a semi-supervised machine learning algorithm and is unable to differentiate between multiple known classes. It is hypothesized that when used together, they should be able to handle both multiple known and

unknown classes. This paper sets out to test this hypothesis, and to determine the best combination of SVM and ANN algorithms, as applied to the problem of small parts inspection.

Experimental setup and image collection

The experimental setup is shown in Fig. 3. The main components are seen to be a camera and a ring light as mounted on a linear belt conveyor. The part (in this case a connector) appears as the bright spot directly under the camera. The conveyor was a Dorner 2200 series with two key features: (1) there were no sidewalls, which enabled flexible lighting and camera arrangements, and (2) conveyor speed range was 0.5–50 m/min. The entire apparatus was placed in an enclosure to ensure uniform lighting conditions. The natural color of the conveyor belt was yellow. A black background was applied to maximize the differentiation between the part being inspected and the conveyor surface.

The camera was a monochrome smart CCD camera (8.5 mm sensor) from NI that could provide 60 frames per second (fps) with a resolution of 640×480 pixels. This industrial camera can be used for on-line classification of moving parts as it contains a built-in processor that can run programs developed off-line in NI Vision Builder or VB. However, VB was only used for image acquisition in this application. Image processing and classification was conducted off-line using the Image Processing Toolbox in MATLAB to enable the development and testing of novel classification methods. The lens was from Kowa with a 6 mm focal length. Aside from its focal length, it was selected because it had markings on the lens for focus and aperture and a locking screw. Most lens do not have markings because they are supposed to work under fixed operating conditions. Nevertheless, given the different nature of the applications for this paper, it was thought that different settings on the lens might be required. The need to change the settings, however, did not emerge.

In preliminary experiments (Joshi et al. 2016), a diffused light produced better quality images as compared to those with a direct light. Both bright field and dark field lighting approaches were subsequently tested for the Indian coin application. Dark field provided better results as compared to bright field, as it tended to minimize the shifting of shadows generated by the internal surface pattern. With this result in hand, an industrial grade dark field ring light was obtained. Specifically, an RL1660 dark field LED-based illuminator from Advanced Illumination was used to provide uniform light in the central field of view (FOV). An orange-red color (wavelength 625 nm) was selected over a red color (wavelength 660 nm) as the smart camera's sensitivity to that wavelength was higher.

For all experiments, data was collected when the parts and conveyor were stationary. Data collection of moving parts and conveyor will be considered as future work. It was observed that if the camera fps is high enough, moving images would appear as stationary images. Specifically, the images did not begin to visibly 'blur' until speed exceeded 400 parts/min (conveyor speed 18 m/min, camera speed 55 fps). The target speed for this application was 100 parts/min. Images were saved from the smart camera directly to the computer by File Transfer Protocol (FTP) over the internet. The internal memory facility on the camera was not used.

Design of the image database

The design of the image database involved two steps: (1) setup original image database and (2) prepare conditioned image database. The basic requirement of the system is to be able to classify multiple classes and reject an unknown class. Following the practice of Wu et al. (2015), it was decided that working with four classes per application would satisfy the "multiple" requirement, to be used in both training and testing. A fifth class was created to be used for testing only. This class is referred to as the others or 'OT' class. Training for OT is not possible because it involves a wide range of images with different properties and characteristics. The OT class covers true negatives (correctly rejected), false negatives (incorrectly rejected), counterfeits and any part that does not belong to the four known classes (i.e. unknown).

Figure 4 shows the five selected classes for the gear (top), connector (middle) and coin (bottom) applications, with the 'test only' class as the 'others' (OT) class. The five gear classes are: 40 teeth spur with 12 holes, 24 teeth spur with internal clutch, 24 teeth crown/bevel front side, 24 teeth crown/bevel back side and 16 teeth spur (as the unknown). The five connector classes are: 4 pin front side, 4 pin back side, 3 pin front side, 3 pin back side and 2 pin one side (as the unknown). The known coin classes are the reverse sides of 1 ₹, 2 ₹, 5 ₹, 10 ₹ Indian coins and the 25 cent Canadian coin (as the unknown). By convention, the "reverse" side of a coin is the side that shows a number. The parts from these three applications can all fit in an area of $50 \text{ mm} \times 50 \text{ mm}$.

Based upon previous experience with the coin application (Joshi et al. 2016), different images were used for training and testing, as this was considered more realistic. 30 physical parts were available for each class; 25 were used for training and 5 were used for testing. One original image was acquired for each part (taken at a random orientation). Figure 5 illustrates the 30 original images for a given class from each application: 24 teeth crown gear (top), 4 pin front side connector (middle) and 5 ₹ coin (bottom). There were a total of 125 images for each application ($25 \text{ samples/class} \times 4 \text{ classes}$ for training and $5 \text{ samples/class} \times 5 \text{ classes}$ for testing). For

Fig. 3 Experimental setup for three applications

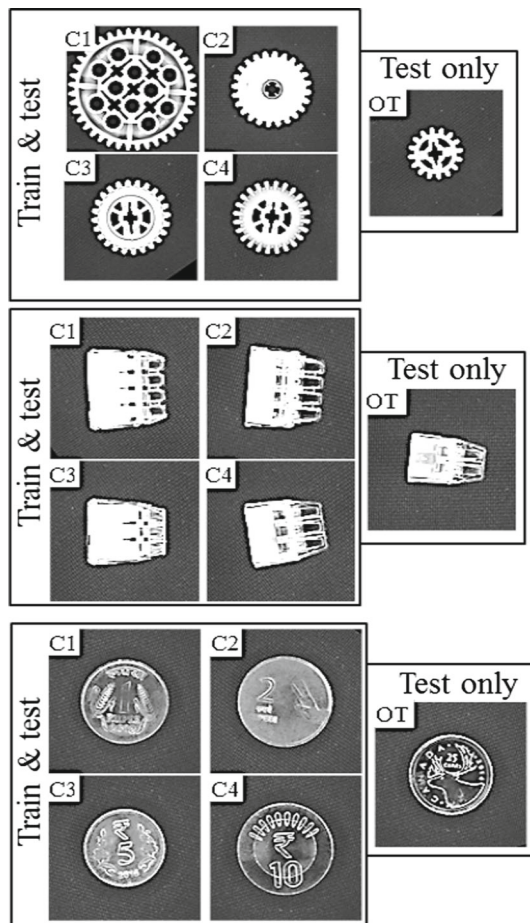
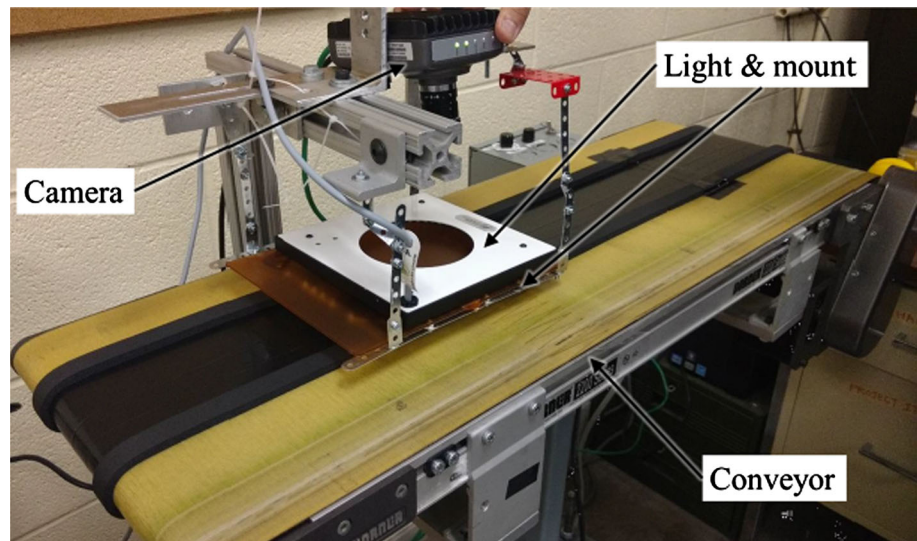


Fig. 4 The five classes for the three applications (Brightness and sharpness increased by 50% for this paper)

details on the rationale for this database design, refer Joshi (2018).

The FOV of the camera was larger than size of the parts. Due to the random nature of their placement, each part was

not necessarily in the center of the FOV when its image was taken. However, the system did guarantee that the part would be in the FOV. The size of each original image was 640×480 pixels. This size is too large for speedy analysis and it also contains irrelevant background information. For faster analysis, a smaller centered image was adopted. Thus, as the image conditioning (IC) step, the following actions were taken on each original image:

1. Cropped to reduce size
2. Translated to center part in image
3. Rotated by 18° increments to generate 20 versions of each original image

This third IC step results in a total of 2000 images for training (100×20) and 500 images for testing (25×20), for each application. The final size of the images is a user input. The following guidelines can be used for size selection:

- Largest part must fit within a 480×480 pixel square with a minimum clearance (black background) of 15 pixels around the outer boundary of the part in the image. If the part does not fit in this square, increase the working distance between camera and the part, followed by focal adjustments, to get the part inside the 480×480 square.
- There can be too much clearance. There can be too little clearance. If the clearance is more than necessary, the size of the part will be small and the system will have less information about the part. If the clearance is less than necessary, cropping might add '0's to the image (depends on the dimension given by user). This is not desired, as it might change the shade of the background.
- Conditioned images will be square only. Therefore, user needs to enter only one dimension of the conditioned image. Maximum value user can enter is 480. Value beyond

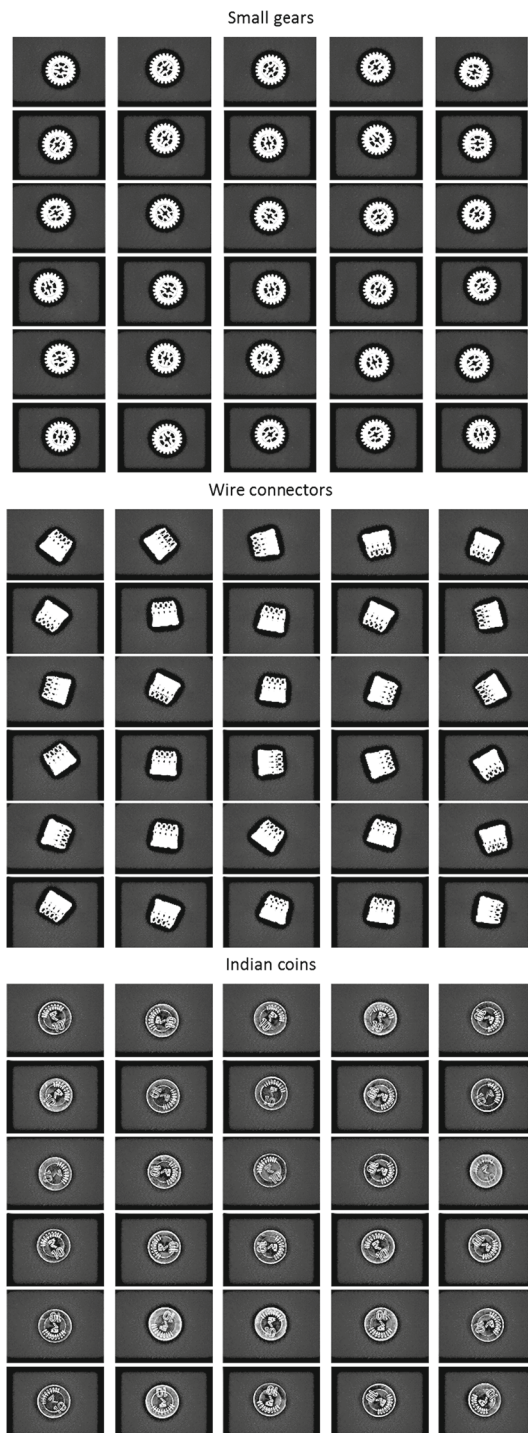


Fig. 5 Sample of 30 original images from a single class for each of the three applications. (Brightness and sharpness increased by 50% for this paper)

480 is not desirable. If the user enters value more than 640, system will not generate square conditioned image because it cannot make square beyond 640 that is original maximum dimension of the source image.

- A value less than 100 pixels is not preferred as it will encourage smaller parts or in the case of the big part will be cropped (part will not be fully visible). Therefore, minimum value set to 100.
- Default value of the dimension is 350 pixels. However, this will not guarantee reasonable performance as selection of this value depends on the largest part from the original images.

Once a user enters the dimension, system will start conditioning training and testing images. This procedure will take time depending on the number of images and class. After completion of IC, FlexMVS will prompt for another inputs from the user for feature extraction purposes. Figure 6 displays 20 conditioned images from a single original image for three applications when user provided default value of 350 pixels as the size of conditioned features. These conditioned images will be used to extract features.

Feature selection and extraction

Feature selection and extraction are critical steps because the value of the feature is the basis for the classification decision. For the best performance, features should be non-redundant, consistent within the class and encapsulate important details of a part. There are two types of features: global and local. Global features are extracted from the whole image and represented by a single feature vector. Local features are calculated from the different points of interest within the image. There are several local feature detectors available, two of the most popular are SIFT (scale invariant feature transform) and SURF (speeded up robust features). Once the points of interest are identified, information from them and their surroundings can be extracted and converted in high dimensional feature vectors. Local features are often computationally expensive.

Feature selection

Features were selected after considering the common properties of the parts being studied, such as size and shape. ‘Color’ is an important feature for a given application when it is the significant differentiator between classes. However, in the three applications considered for this work, color was not a significant differentiator. Therefore, color-based features were not selected. This does not mean that the selected features were not influenced by the color. Intensity-based features are dominated by shades of grey, which is a measure of color.

Tuytelaars and Mikolajczyk (2008) in their survey of feature detectors, provide a few guidelines for feature selection. One important guideline addressed was the level of

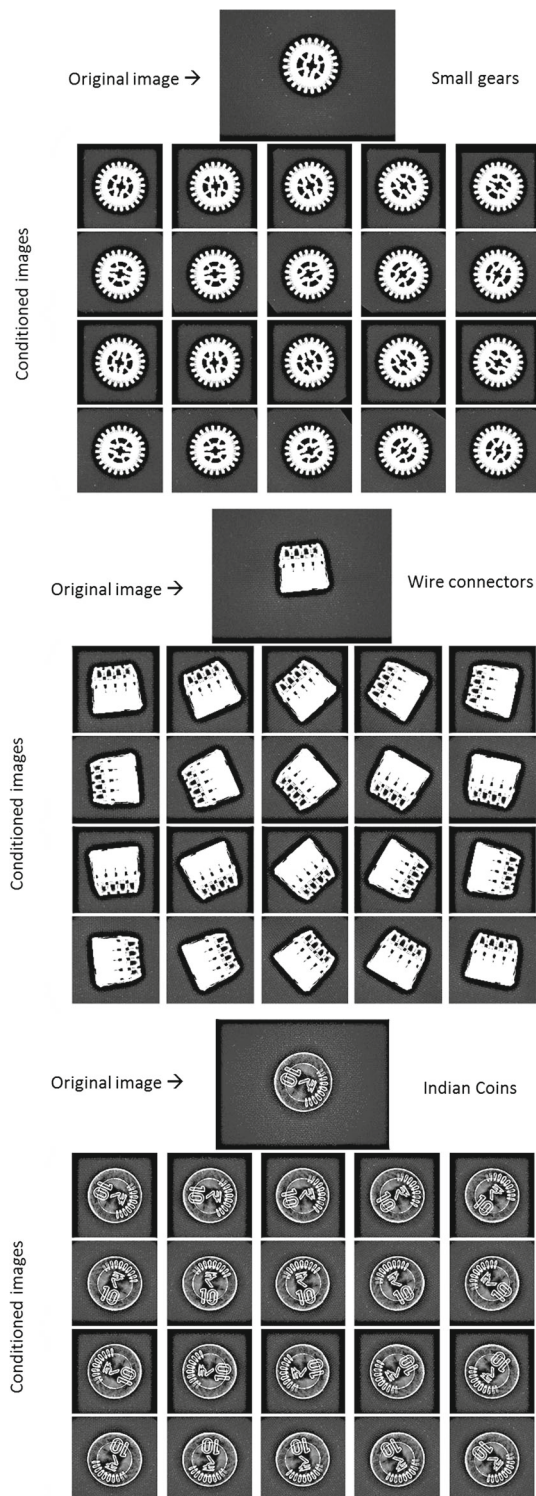


Fig. 6 Sample of 20 conditioned images from an original image for each of the three applications. (Brightness and sharpness increased by 50% for this paper)

invariance. As the level of invariance is increased, the discriminative power of a feature is decreased. This implies that the level of invariance should be as low as possible. On the

other hand, a low level of invariance cannot compensate for observed variability.

When the number of selected features is large, the non-relevant features can negatively impact the training of the model. Blum and Langley (1997) first pointed out that non-relevant features reduce the rate of learning and require more training to reach a given accuracy. More recently, Chetima and Payeur (2008) used 82 features to decide whether to accept or reject a sample. To remove non-relevant features, they employed four different feature selection methods. One of their methods known as RELIEF reduced the number of features from 82 to 10. Dash and Liu (2003) focused on inconsistency-based feature selection in order to minimize the number of features. In one of their datasets termed as Splice, they reduced the number of features from 60 to 9. Hua et al. (2005) demonstrated that for a sample size of 200, perceptron and linear/polynomial SVM-based linear/nonlinear model would have an optimal feature size in between 10 and 30. In this paper, the sample size is 500. Thus, the 14 features used in this paper is not considered a large number and is consistent with the range recommended by Hua et al. and as used by others.

Out of the 14 features considered in this work, the first 5 are global in that they work with the whole image. These global features are: average intensity (AVIN), black to white pixels ratio (BWR), circularity (CIRC), diameter (DIAM) and frequency weighted intensity (KAVG). The remaining 9 features are local in that information from only the pixels surrounding the individual points of interest are used. However when combined, these 9 features become global because every pixel in the conditioned image was utilized to get their values. Individually, these 9 features (labelled I1 to I9) are intensity values of the local image (3×3 pixels).

These features are selected considering the three applications at hand. Some features are dominant for one application while other features are dominant for another application. For example, DIAM is dominant for coins whereas BWR is dominant for connectors. It was found that the combination of these 14 features was able to provide the target performance for all three applications.

In summary, in order to generate the conditioned image database, the user had to provide the following inputs:

- (1) Size of conditioned image in pixels
- (2) Size of largest part in pixels
- (3) Size of smallest part in pixels

Feature extraction

With the conditioned image database in place, the next step is to start extracting features from the training image dataset. The system will ask the user to select class wise conditioned training images for labelling purposes wherein the user can

specify the class name. A feature vector is prepared for each image. If there are 500 images of a given class in the training dataset, 500 feature vectors prepared. The same procedure employed for all classes. Thus, for a training dataset with four classes, 2000 (500 feature vectors \times 4 classes) feature vectors prepared. After all the feature vectors are generated, the system will ask the user to select class wise conditioned *testing* images for labelling purposes, wherein the user can *again* specify the class name. However, in this case, change in the class name will only be in the 5th (OT) class. The testing dataset labelled in order to enable calculation of the performance measures.

The calculated feature values are continuous in nature. An 8-bit discretization procedure applied to reduce complexity and ensure that when input to the classifier, all feature values were discrete in nature with a range of 1–256. This approach is analogous to having a histogram of 256 bins. A higher than 8-bit discretization would not significantly reduce discretization error. A lower than 8-bit discretization may not provide sufficient range for the feature. For a given feature, discretization of the *training* features is straightforward. The image with the lowest feature value is assigned to the 1st bin and the image with the highest feature value is assigned to the 256th bin. The intermediate feature values are scaled and assigned to their bins, respectively. However, discretization of the *testing* features is not as straightforward.

When discretizing testing features, information must be retrieved from the training images for the range of the feature values. The Eq. (1) is used to interpolate the discretized value of a feature for a testing image:

$$F_{1D}^{test} = Round \left\{ 1 + \left(F_{1C}^{test} - F_{1Cmin}^{train} \right) \left(\frac{255}{F_{1Cmax}^{train} - F_{1Cmin}^{train}} \right) \right\} \quad (1)$$

where F_{1D}^{test} = discrete value of the test image for feature F_1 , F_{1C}^{test} = continuous value of the test image for feature F_1 , F_{1Cmin}^{train} = minimum value of continuous feature F_1 from training dataset of all classes and F_{1Cmax}^{train} = maximum value of continuous feature F_1 from training dataset of all classes. The Round function rounds to the nearest integer, with decimal point 5 rounded up.

If the discrete (interpolated) value of a feature for a testing image is outside the range of the training discrete feature dataset, it will be clipped to the minimum or maximum value of the training discrete feature dataset. Once all the discrete values are obtained for the training and testing datasets, the next task is to normalize these in between 0 and 1. The normalizing procedure is the same for both: training and testing datasets. The following equation is used to calculate the normalized value of a discrete feature:

$$F_{1N} = F_{1D} * 0.00390625 \quad (2)$$

where F_{1N} is the normalized value of the feature F_1 , F_{1D} is the discretized value of the continuous feature F_1 and 0.00390625 (= 1/256th) is the resolution obtained for an 8-bit normalization inspired from 8-bit discretization. By using Eq. (2), normalized features range from [0.00390625, 1] in steps of 0.00390625. The output of this procedure is class wise normalized feature values. These normalized values are used in the next step for developing hybrid models and thereby predicting the class for a test image.

The normalized values of the 14 features for the images in the training dataset for the gear application are plotted by class in Fig. 7. The top two plots show the actual feature values for all 2000 images of the four classes. The actual value plots illustrate the degree of features overlap between classes. The bottom two plots depict the median feature values for the four classes. The median value plots illustrate the degree of separation between features by class, as well as the target feature values for a test image to be considered as one of the classes. From Fig. 7 actual value plots, one can compare the feature values between classes. For example, the I5 value of C1 is always less than the I5 value of C2 and the BWR value of C1 is always less than the BWR values of the other classes. The utility of the median value plots will become more apparent when dealing with the more difficult connector and coin applications, where the degree of overlap between classes with the actual value plots becomes more evident.

As highlighted in Fig. 7, I5 is the best feature for the gear application as it provides the clearest differentiation between the 4 classes. There is a fair degree of overlap of the values with the other features. This does not mean that I5 is the only feature that should be used. All features become important in the testing phase with the introduction of the OT class. For an OT part to get accepted in the testing stage, it must have values of all 14 features within an acceptable range. Figure 7 can also be used to identify the least effective features. For example, DIAM is the worst feature because according to the actual values plots, there is a high degree of overlap between classes C2, C3 and C4 (highlighted in Fig. 7). Its closest competitor is CIRC that also has overlap between classes C2, C3 and C4. Thus, the prediction is that both CIRC and DIAM will be the least effective features for this application.

Figure 8 gives the normalized values of the 14 features by class of the training image dataset for the connector application. As highlighted in the figure, BWR is the best feature for this application as there is no overlap between classes (from actual value plots). There is clear overlap for the other 13 features. However, median value plots can help for finding least effective features. In Fig. 8, the least effective feature is still DIAM (highlighted) as according to both (actual and median value) plots. There is considerable amount of overlap between C1 and C2, and C3 and C4. Its closest competitor is CIRC, where the median value plot shows considerable

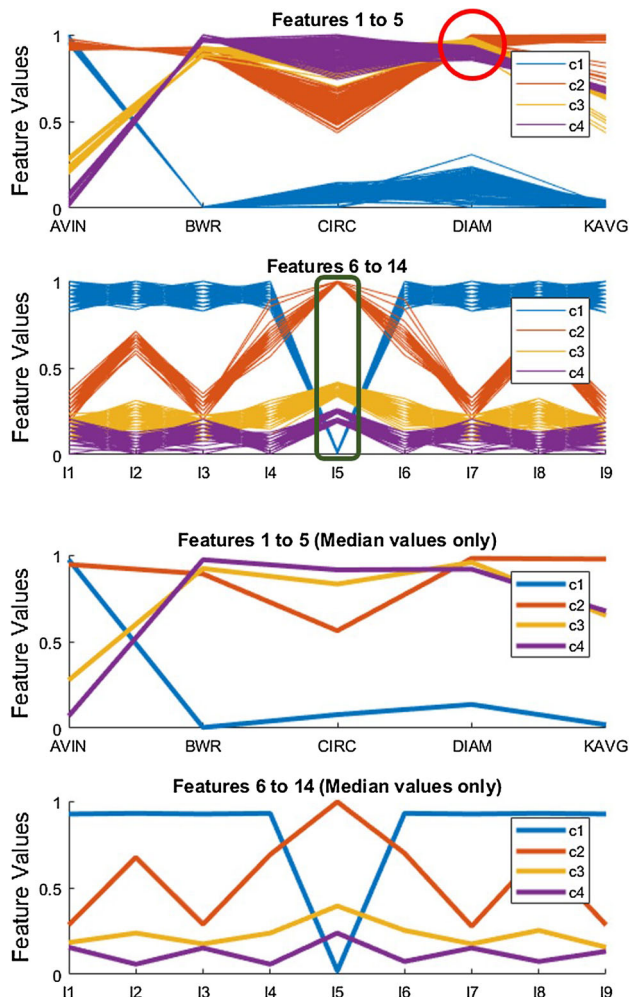


Fig. 7 Gear feature values for training image dataset by class (with I5 in green and DIAM in red) (Color figure online)

amount of overlap between Class 1 and 3 but Class 2 and 4 were ambiguously separated.

An analysis was carried out from both (actual and median value) plots for comparison between any possible combinations of only two classes. It turned out that out of 14 features, 12 features help in discriminating between C4 versus C1; 4 features help in discriminating between C1 versus C2 and C3 versus C4. On this basis, two predictions can be made about the connector application: (a) C1 versus C4 will be the easiest to differentiate between and (b) C1 versus C2 and C3 versus C4 will be the hardest to differentiate between. This meant that the system will be more confused between C1 versus C2 and C3 versus C4 relative to the C1 and C4.

Figure 9 gives the normalized values of the 14 features by class of the training image dataset for the coin application. The fact that there is not one feature that can clearly differentiate between the classes confirms the difficult nature of this application. It is the combination of these features that provided the necessary differentiation between classes in the

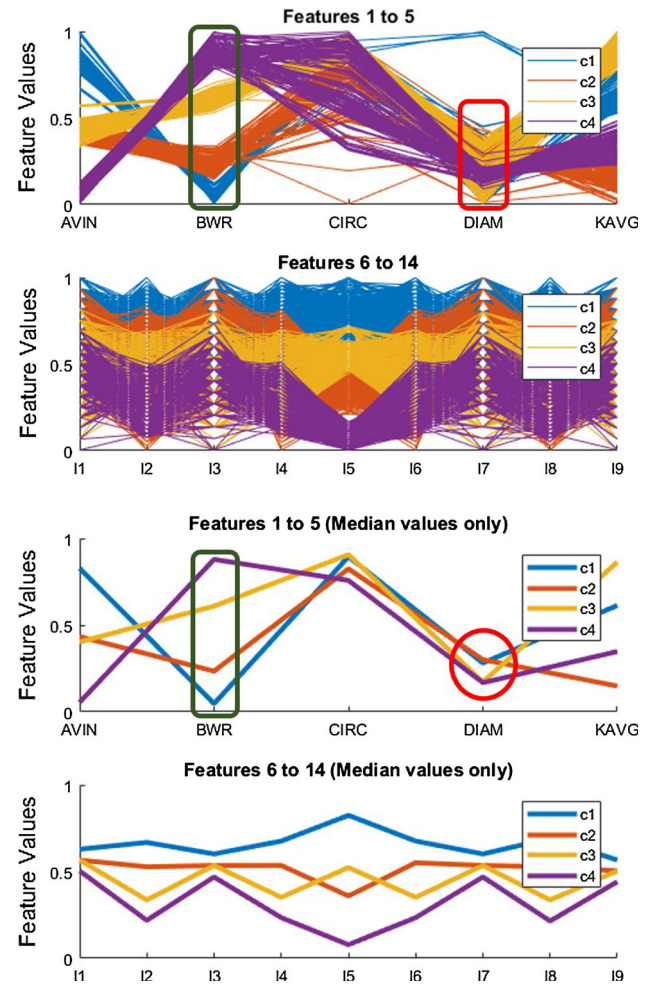


Fig. 8 Connector feature values for training image dataset by class (with BWR in green and DIAM in red) (Color figure online)

testing stage. However, it is possible to make some predictions for the coin application from this figure.

The best features for the coin application need to be determined from the median value plots of Fig. 9, as the actual value plots are seen to overlap considerably for all features. The most effective features as seen to be AVIN and I5, as the two features whose median values are distinctly different (highlighted in Fig. 9). By contrast, KAVG and I9 are two features whose median values are very close and provide the least degree of discrimination (also highlighted in Fig. 9).

Classification methods

As the difficulty of an application increases, it will demand more conservative and stricter ways to get the target performance. Ideally, the number of False Positives (FPs) and False Negatives (FNs) should be minimized. However, in most part-manufacturing applications, FNs are preferred over FPs,

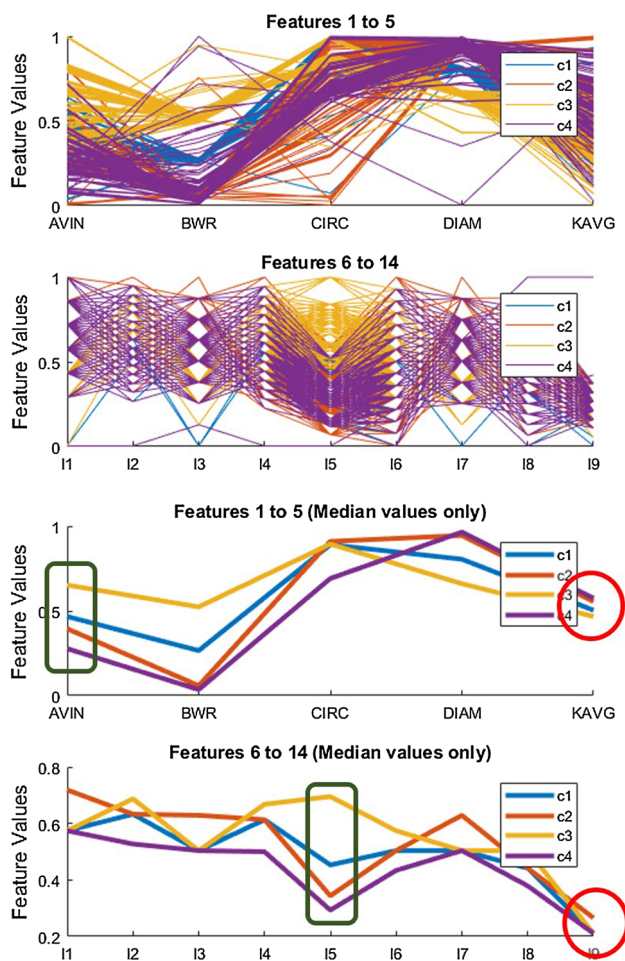


Fig. 9 Coin feature values for training image dataset by class (with AVIN and I5 in green, KAVG and I9 in red) (Color figure online)

because one can always recycle incorrectly rejected good parts (FNs), but one cannot permit incorrect acceptance of faulty parts (FPs). Therefore, the target was to get 0% FPs along with accuracy of more than 95%.

Four classification methods were developed and tested for the three applications. All methods are hybrid in the sense that they use both supervised and semi-supervised machine learning algorithms, in order to meet the requirement that the system be able to classify parts into multiple (known) classes and reject any unknown class. The developed system must learn all the classes presented in training, but must also learn to ‘reject’ classes not covered in training. The reject class is a subset of the OT class, which is defined earlier in “[Design of the image database](#)” section, includes a range of possibilities (i.e. FNs, TNs, counterfeits). Although, it is not possible to train for the OT class, one must still be able to test and classify images as OT.

In the procedure of classification, the system requires labelled images that the user has provided in the feature extraction step. Details of the four hybrid methods will

be covered in next paragraphs. All four methods were implemented by using a combination of MATLAB’s Image Processing, Statistics and Machine Learning, Neural Network and Computer Vision toolboxes. Each method was a combination of two of three different machine-learning algorithms: SSVM, USVM and SANN; where SSVM stands for supervised SVM, USVM stands for semi-supervised SVM and SANN stands for supervised ANN.

Method M5: USVM-SSVM classification

SVM is a classification algorithm that aims to maximize the distance between class boundaries (Vapnik et al. 1996). With labelled training images as input, the SVM algorithm builds a model to predict the class of an unlabeled test image dataset. Method M5 uses SVM for both supervised and semi-supervised machine learning, with the semi-supervised SVM being applied before the supervised SVM, hence the designation USVM–SSVM.

The first layer of M5 uses semi-supervised SVM to identify images belonging to the OT class. This is implemented by combining the training datasets of four classes into one class and preparing a temporary single class called ‘accept’. The semi-supervised classifier will learn this one class ‘accept’ and make OT the second class. The semi-supervised SVM classifier then calculates the classification score for each image in the test dataset. Once the classification score is obtained, a decision on whether the test image belongs to the ‘accept’ class or the OT class can be made based upon the value of the classification score. If the classification score is negative, the image is classified as OT as it is an outlier for the ‘accept’ class. Otherwise, the image is classified as being in the ‘accept’ class. The second layer of M5 is implemented only if the result from the first layer was ‘accept’.

The second layer of M5 uses a supervised SVM classification algorithm for the four known classes. This involves training a number of binary SVM classifiers to reduce the problem from multi-class to binary class. The actual number of binary classifiers will be discussed in “[Results and discussion](#)” section. The prediction provided by the second layer is taken as the decision for the test image that passed in the first layer.

There is a difference between learning a single class (with low pattern variation) and learning number of classes altogether as a single class (with high pattern variation). When the pattern variation is high, the ability of the system to recognize the OT class is compromised. Therefore, if the first layer of M5 incorrectly predicts the image as one of the known classes, then the second layer will classify that unknown image into one of the four classes, as it is a supervised layer. Thus, it is predicted that M5 will have non-zero FPs.

Method M6: SSVM-USVM classification

This method is similar to M5 as it also uses two SVM algorithms, except that their order is reversed, hence it is designated as SSVM–USVM. Thus, the first layer is the application of the supervised SVM classifier to the four known classes. Specifically, a single multi-class SVM classifier (composed of multiple binary classifiers) learns from the training dataset and classifies images from the test dataset into one of the four known classes. This means that an image that belongs to the OT class gets classified into one of the four known classes by the first layer. Based on its training, the second layer will be able to correct this mistake.

The second layer of M6 sets out to validate the prediction of the multi-class classifier in the first layer. As the number of known classes is four, four semi-supervised SVM classifiers are trained with images from their respective classes. For example, if the prediction for an image in the first layer is C1, then the image will undergo a validation step with the semi-supervised SVM classifier for C1 in the second layer. The image will be classified as C1, only if it passes this second layer. Otherwise, the image will be classified as OT. Classes C2, C3 and C4 are handled in a similar fashion. The first layer of M6 classifies OT images as one of the four known classes due to its supervised nature. However, the second layer will ‘catch’ that image as OT because it does not align with any of the images in the training dataset.

Every test image will be classified as one of the known classes by the first layer of M6, as it is a supervised layer. The second layer is strict in classifying images into one of the known classes as it contains four binary classifiers with low pattern variations. Because of this, it is predicted that M6 will have zero FPs.

Method M7: USVM-SANN classification

Supervised Artificial Neural Networks (SANNs) are widely used for classification problems. SANNs are inherently adaptive as they can map any input–output continuous relation provided that they are given a sufficient number of hidden neurons and a properly designed training dataset (Nielsen 2015). However, they can only work with known classes. In method M7, in order to enable SANN for an application that has an unknown class, semi-supervised SVM is applied before SANN, hence it is given the designation USVM-SANN.

This method is similar to M5 in the sense that SVM is used for the unsupervised learning of the unknown or OT. The difference between M5 and M7 is that instead of using SVM for supervised learning, M7 uses SANN. The first layer in M7 uses semi-supervised SVM to check for images in the OT class. If the classification score is negative, the image is classified as OT, as it is an outlier for the ‘accept’ class.

Otherwise, the image is classified to be in the ‘accept’ class. The second layer of M7 is implemented only if the result of this first layer is ‘accept’.

The second layer for M7 is SANN learning. Only the images that pass in the first layer are considered in the second layer. SANN will predict the class of the image based on the training with the four known classes. This means that irrespective of the original class of the image, SANN will classify the image into one of the four known classes. As a consequence, it is predicted that M7 will have non-zero FPs.

There are two possibilities for an FP occurring with M7: (1) an OT image is classified as a known image by the first layer and the second layer (SANN) classifies it as one of the known four classes; and (2) first layer correctly classifies an image as a known class image, however, the second layer classifies the image into an incorrect known class (i.e. classifies C1 as C4). This implies that M7 will have non-zero FPs.

Method M8: SANN-USVM classification

This method is similar to M7 as it also uses an SANN in combination with a USVM, except that their order is reversed, hence it is designated SANN-USVM. Thus the first layer is the application of a multiclass SANN classifier to provide an initial prediction of an image’s class. Every image from test dataset will go through the first and second layers. In the second layer, one of the four semi-supervised SVM classifiers will be used to determine if the initial prediction was correct. Prior to this, four semi-supervised classifiers are trained to predict a known class or an OT class. These classifiers are strict in classifying a test image into the known class. The four classifiers correspond to each of the four known classes.

In the second layer, one of the four available classifiers is selected based on the initial prediction by the SANN multiclass classifier. For example, consider a case where a test image’s initial prediction from first layer is C3. Then in the second layer, the semi-supervised SVM classifier designed for C3 will be applied. If the image is actually from C3, it will pass the second layer and the final prediction will be C3. On the other hand, if it does not pass second layer, it is classified as OT. To illustrate the nature of M8 further, the training and testing procedures for M8 are given in Figs. 10 and 11, respectively. A subroutine named ‘Get vectors’ used in training is included in Fig. 10.

A key difference between M7 and M8 is the use of binary semi-supervised SVM classifier/s with low (strict, as in M7) and high pattern variation (not strict, as in M8). This implies that similar to M6, M8 will not provide any FPs. Performance differences between M6 and M8 will be dependent on the individual capability of the supervised multi-class classifiers of SVM (M6) and ANN (M8).

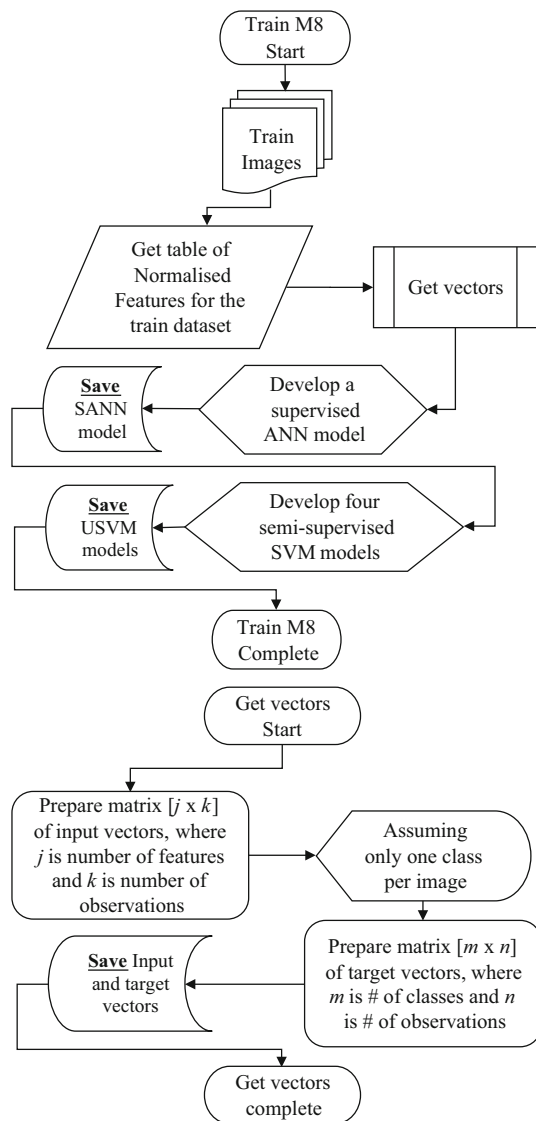


Fig. 10 Flowchart explaining training procedure of M8

Performance measures

The simplest and most popular performance measure is the accuracy. In the context of this paper, it is defined as the correct number of predictions divided by the total number of predictions. FPs and FNs are incorrect classifications. True Positives (TPs, correctly accepted) and True Negatives (TNs, correctly rejected) are considered correct classifications. Variations on accuracy with similar inputs include positive predictive value (correctly accepted out of total accepted), true positive rate (correctly accepted out of total positive) and true negative rate (correctly rejected out of total negative) (Sokolava and Lapalme 2009). For this paper, two performance measures will be used: (1) percentage accuracy and (2) percentage of FPs. As stated in “Classification meth-

ods” section, the target performance is 95% accuracy with 0% FPs.

Results and discussion

As a benchmark of performance, Table 1 summarizes the results and parameters for the coin application with two conventional non-hybrid methods, when tested with an image database that excluded unknown images. The first method used an SSVM (designated M1). The second method used an SANN (designated M2). As mentioned earlier, Modi and Bawa (2011) reported 97.7% accuracy for Indian coins using an ANN, with a database of only known images (Experiment E1-R in Table 1). Chauhan et al. (2017) repeated Modi’s experiment with the same parameters and ANN setup, but with a higher quality image database, and achieved 100% accuracy, with both non-hybrid SSVM and SANN (E2-R in Table 1). When Chauhan’s experiment was repeated, but with 4 instead of 14 classes and 14 instead of 400 features, the accuracy was still 100% for both SSVM and SANN (E3 in Table 1).

In order to obtain quantitative measure of the negative impact of unknown images, Table 2 repeats experiment E3 from Table 1, with the same parameters, but with unknown images introduced to the database, that is the database developed for this paper. The third method used the Supervised K Nearest Neighbor (SKNN, designated M3) approach with the value of K set to 1. In M3, the developed model compares the feature vector of the test image with the feature vectors of the training images. The model would then find an image from the training dataset whose feature vector was the nearest to the test image. Finally, the model would assign the class of that image from the training dataset to the test image.

The fourth method used the Supervised Bag of Words (SBOW, designated M4) approach. For M4, the vocabulary for visual words was prepared by K means clustering of extracted SURF features from training images, where the value of K was 500. In the training of M4, a model was developed considering the frequency of these 500 visual words from the training dataset. The test image in M4 gets classified based on the frequency of the 500 visual words. Note that M4 is the only method of the 8 methods examined that did not use the features from the designed generic feature library (of 14 features).

For the coin application (E6), the accuracy drops from 100 to 69% for SSVM, 76% for SANN, 71% for SKNN and 66% for SBOW. Thus, the impact is seen to be significant. The accuracy for the gear (E4) and the connector (E5) applications is 80% for both, which is still well below the target accuracy of 95%. Note that for the connectors, M4 dropped the accuracy to a low of 78% with E5. It is acknowledged that if the number of unknown images doubles (reducing

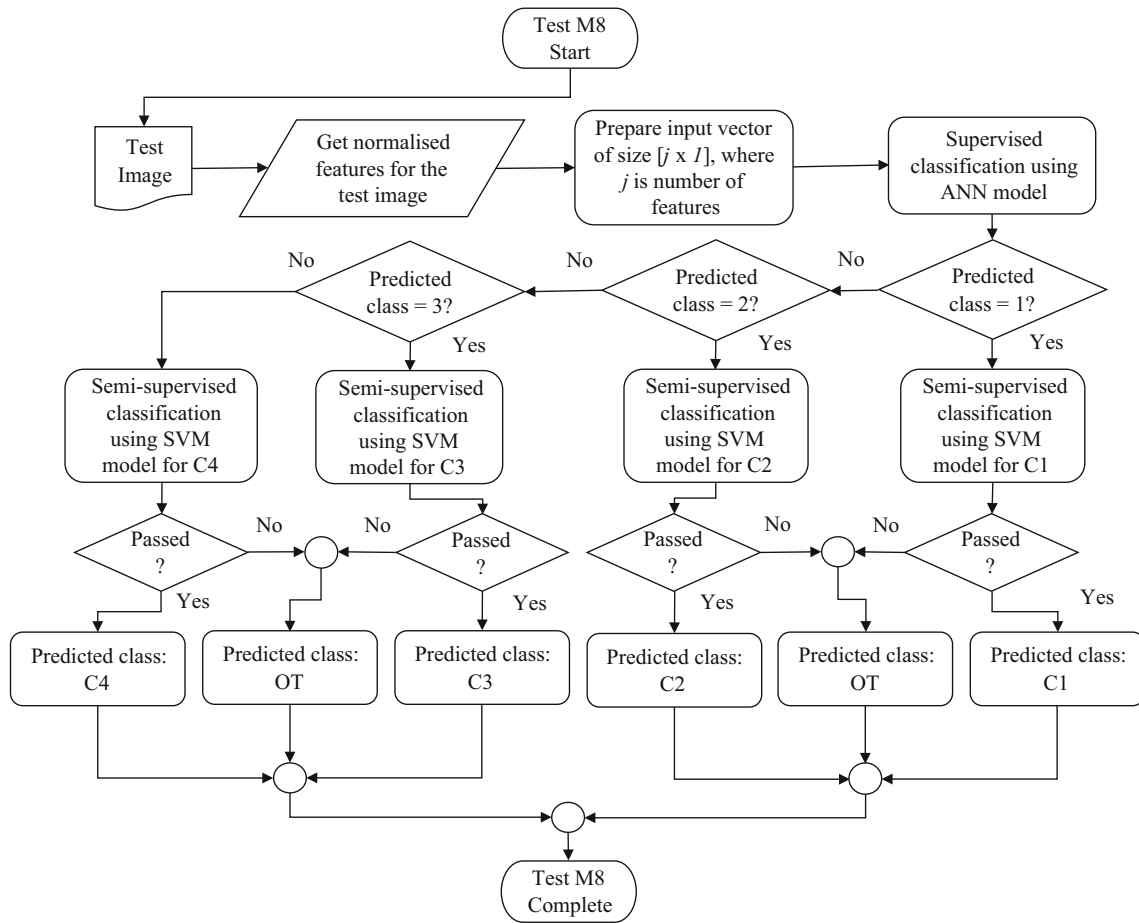


Fig. 11 Flowchart explaining testing procedure of M8

Table 1 Benchmark SSVM and SANN results with partial database (excludes unknown images, OT class) of classes as shown in Fig. 4, where suffix ‘-R’ in experiment number denotes experiment from the reference paper

Exp. #	Application	M1 (non-hybrid) SSVM		M2 (non-hybrid) SANN		Parameters		
		Acc. (%)	FPS. (%)	Acc. (%)	FPS. (%)	Features	Classes	Images per class
E1-R	Coins, Modi and Bawa (2011)	-	-	97.7	-	400	14 (train & test)	342 (train) and 18 (test)
E2-R	Coins, Chauhan et al. (2017)	100	0	100	0	400	14 (train & test)	342 (train) and 18 (test)
E3	Coins (this paper)	100	0	100	0	14	4 (train & test)	500 (train) and 100 (test)

the number of known images, to keep the same size of test dataset) the accuracy would drop to 60%. For experiments E3 to E6, SSVM was multiclass with 6 binary classifiers using the ‘one versus one’ (OVO) method. SANN was multiclass with 14 neurons in the input layer, 10 neurons in the hidden layer and 4 neurons in the output layer. For further details, see Modi and Bawa (2011) and Chauhan et al. (2017). SKNN was multiclass with one classifier. SBOW was mul-

ticlass with one classifier. No FNs were reported for any of the experiments as is seen in an examination of Tables 1 and 2.

Table 3 summarizes the results with the four hybrid classification methods described in “Classification methods” section. A total of 18 experiments were conducted (E7 to E24). Experiments E7, E8 and E9 were conducted with default user inputs: 350 pixels as the conditioned image size,

Table 2 Conventional methods results with full database (includes unknown images) of classes as shown in Fig. 4

Exp. #	Application	M1 (non-hybrid) SSVM		M2 (non-hybrid) SANN		M3 (non-hybrid) SKNN		M4 (non-hybrid) SBOW	
		Acc. (%)	FPs. (%)	Acc. (%)	FPs. (%)	Acc. (%)	FPs. (%)	Acc. (%)	FPs. (%)
E4	Gears	80	20	80	20	80	20	80	20
E5	Connectors	80	20	80	20	80	20	78	22
E6	Coins	69	31	76	24	71	29	66	34

Parameters used for E4, E5 and E6 are same as parameters used for E3 features = 14, classes = 4 (train) & 5 (test), images per class = 500 (train) & 100 (test)

Table 3 Hybrid SSVM/SANN results with full database (includes unknown images) of classes as shown in Fig. 4

Exp. #	Application	Image Size (square pixels)	Smallest part size (pixels)	Largest part size (pixels)	M5 (hybrid) USVM–SSVM		M6 (hybrid) SSVM–USVM		M7 (hybrid) USVM–SANN		M8 (hybrid): SANN–USVM	
					Acc. (%)	FPs. (%)	Acc. (%)	FPs. (%)	Acc. (%)	FPs. (%)	Acc. (%)	FPs. (%)
E7	Gears	350	160	340	100.0	0.0	99.0	0.0	100.0	0.0	99.0	0.0
E8	Connectors				80.0	20.0	98.8	0.0	80.0	20.0	98.8	0.0
E9	Coins				69.4	30.6	89.2	0.0	76.0	24.0	96.2	0.0
E10	Gears	345	160	340	100.0	0.0	99.6	0.0	100.0	0.0	99.6	0.0
E11		355	160	340	100.0	0.0	100.0	0.0	100.0	0.0	100.0	0.0
E12		350	150	330	100.0	0.0	97.8	0.0	100.0	0.0	97.6	0.0
E13		350	170	350	100.0	0.0	98.4	0.0	100.0	0.0	98.4	0.0
E14		350	150	350	100.0	0.0	97.8	0.0	100.0	0.0	97.8	0.0
E15	Connectors	345	160	340	80.0	20.0	98.8	0.0	80.0	20.0	98.8	0.0
E16		355	160	340	80.0	20.0	99.8	0.0	80.0	20.0	99.8	0.0
E17		350	150	330	80.0	20.0	100.0	0.0	80.0	20.0	100.0	0.0
E18		350	170	350	80.0	20.0	98.8	0.0	80.0	20.0	98.6	0.0
E19		350	150	350	79.6	20.0	97.6	0.0	79.6	20.0	98.0	0.0
E20	Coins	345	160	340	68.4	31.6	88.0	0.0	75.6	24.4	96.8	0.0
E21		355	160	340	70.0	30.0	89.4	0.0	73.8	26.2	94.4	0.0
E22		350	150	330	69.4	30.6	89.2	0.0	75.6	24.4	97.2	0.0
E23		350	170	350	69.4	30.6	89.2	0.0	77.6	22.4	97.0	0.0
E24		350	150	350	69.4	30.6	89.2	0.0	75.8	24.2	97.4	0.0
-	Average performance				83	17	96	0	85	15	98	0

160 pixels as the size of the smallest part and 340 as a size of the largest part. The remaining experiments varied the user inputs to study their effect. The observation from Table 3 is that M8 (ANN-USVM) provided the best performance with

an average accuracy of 98% and zero FPs. As shown in table, changing the user inputs for M8 from ± 2 to ± 6% causes the accuracy to range from 94% (E21) to 100% (E17). Finally, the default user input results give the gear, connector and

Table 4 Confusion matrices of experiments E7, E8 and E9 by method M8 (SANN–USVM)

Gears E7		Predicted classes					Subtotal
	Class	C1	C2	C3	C4	OT	
Actual classes	C1	95	0	0	0	5	100
	C2	0	100	0	0	0	100
	C3	0	0	100	0	0	100
	C4	0	0	0	100	0	100
	OT	0	0	0	0	100	100
Accuracy: 99%		FPs: 0%					Total: 500
Connectors E8		Predicted classes					Subtotal
	Class	C1	C2	C3	C4	OT	
Actual classes	C1	100	0	0	0	0	100
	C2	0	100	0	0	0	100
	C3	0	0	100	0	0	100
	C4	0	0	0	94	6	100
	OT	0	0	0	0	100	100
Accuracy: 98.8%		FPs: 0%					Total: 500
Coins E9		Predicted classes					Subtotal
	Class	C1	C2	C3	C4	OT	
Actual classes	C1	100	0	0	0	0	100
	C2	0	82	0	0	18	100
	C3	0	0	99	0	1	100
	C4	0	0	0	100	0	100
	OT	0	0	0	0	100	100
Accuracy: 96.2%		FPs: 0%					Total: 500

coin application accuracies as 99.0, 98.8 and 96.2%. This is consistent with the belief that the gear application was the least difficult and the coin application was the most difficult.

Table 3 can also be used to confirm the FP predictions from “Classification methods” section. It was predicted that M5 will have non-zero FPs. The results for M5 confirm that FPs are present for the connector and coin applications. For M6, it was predicted that there will be zero FPs. Results for M6 for all experiments confirm that prediction. The prediction for M7 was that it will have non-zero FPs. Results for M7 confirms this prediction for the connector and coin applications. M8 was predicted to provide zero FPs. Results for M8 for all experiments are in agreement with this prediction.

Table 4 gives the results for E7, E8 and E9 with M8 in the form of a confusion matrix (CM), which tabulates the actual classes versus predicted classes. The actual classes are the ones labelled by the user. The class assigned by the classifier to an image is the predicted class. The performance of the classifier is reported by three measures: accuracy, percentage of FPs and FNs. FNs can be obtained by subtracting accuracy plus FPs from 100. In a confusion matrix, the diagonal values are TPs and TNs, the off diagonal values excluding

last column (OT) are FPs, and the off diagonal values in the last column (OT) are FNs. The information from the CM can provide insights into why a method performed the way it did.

In the Table 4, the CM for E7 (gears) shows that 5 images of C1 were classified as OT by the system. The CM for E8 (connectors) shows that 6 images of C4 were classified as OT by the system. The CM for E9 (coins) shows that 18 images of C2 were classified as OT by the system. This is because those images did not provide feature values that were in the acceptable range for the classifier. The same reason is applicable to the three applications. The CM also indicates which class for a given application is the most difficult to classify. For example, CM for E9 in Table 4 illustrates that class C2 was the most difficult to classify as it registered the maximum number (18) of FNs.

Figures 12, 13 and 14 show feature plots obtained from the test datasets for gears, connectors and coins, respectively. The actual value plots in each figure shows feature values of 500 images of the test dataset of an application. Even though the median value plots look to be the same for C1 to C4 in training and testing (Figs. 7 vs. 12, Figs. 8 vs. 13 and Figs. 9 vs. 14), they are not exactly the same due to the adopted

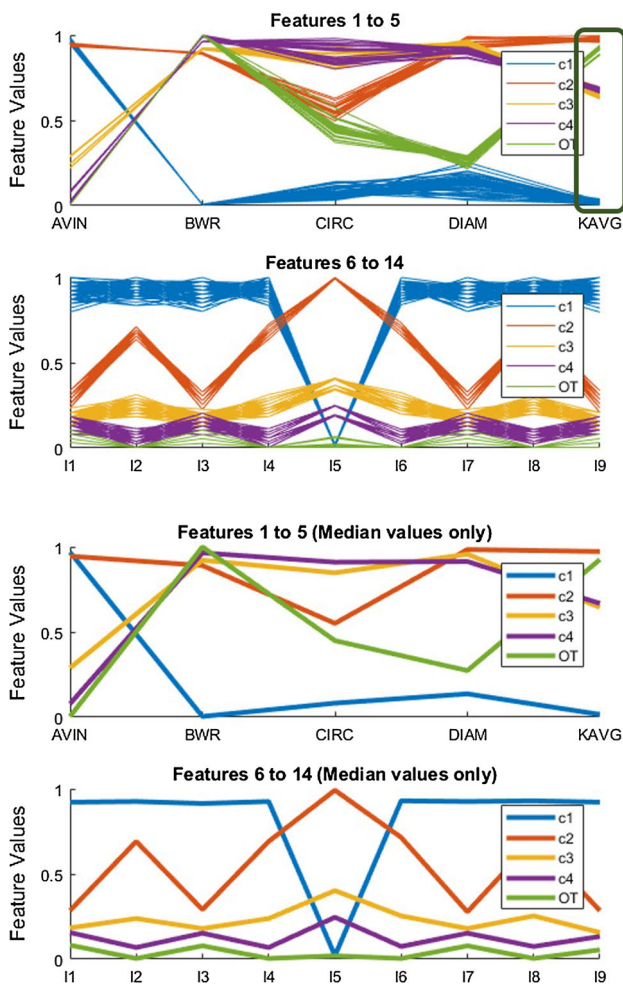


Fig. 12 Gear feature values for testing image dataset by class (with feature KAVG in green) (Color figure online)

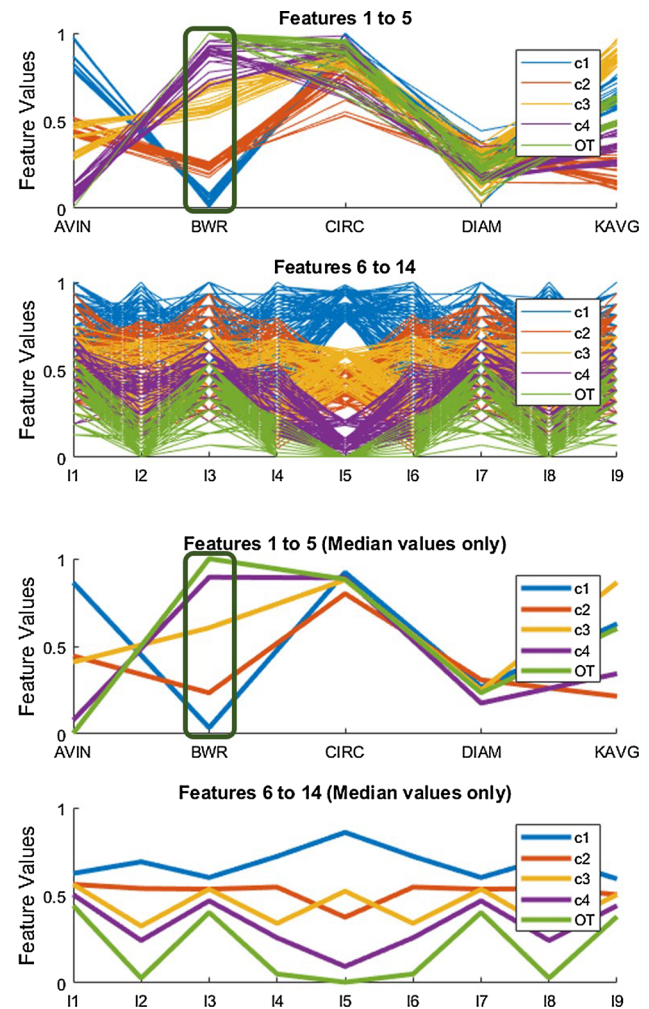


Fig. 13 Connector feature values for testing image dataset by class (with feature BWR in green) (Color figure online)

DTT strategy for the setup of the training and testing datasets (Chauhan et al. 2017). As per the discussion in “Classification methods” section, it is not possible to fully test the OT class. Thus, the fifth class prepared for OT should not be considered as the only OT. For example, any non-Indian coin or Indian coins with unknown pattern would be considered as OT for the coin application.

One can see that different features can become important when the OT class is introduced. For example, with the gear application, I5 was the best discriminating feature in training (Fig. 7, highlighted in green for actual value plots); but KAVG effectively discriminates OT from C1, C2, C3 and C4 in testing (Fig. 12, highlighted in green for actual value plots). Interestingly for the connector application, the important feature in training was BWR (Fig. 8, highlighted in green for both: actual and median value plots), the same feature effectively discriminates OT from C1, C2, C3 and C4 in testing (Fig. 13, highlighted in green for both actual and median value plots).

For the coin application, the important features in training were AVIN and I5 (Fig. 9, highlighted in green for median value plots); but BWR and DIAM were the important features that could effectively discriminate OT from C1, C2, C3 and C4 in testing (Fig. 14, highlighted in green for median value plots).

The general observation from this discussion is confirmation that features that are important for one application, may not be important for another application. Thus, for an MV system to be successful and flexible, the feature set must be comprehensive. For the three applications considered in this paper, the selected set of 14 features was able to do the job because they combined both geometrical and statistical measures. Moreover, these features were easy to calculate in comparison with advanced features such as SIFT and SURF. Based on these observations, the likelihood that the proposed system will work for other applications is considered high. The system has the following basic constraints for a given application: (1) the conditioned image should be between

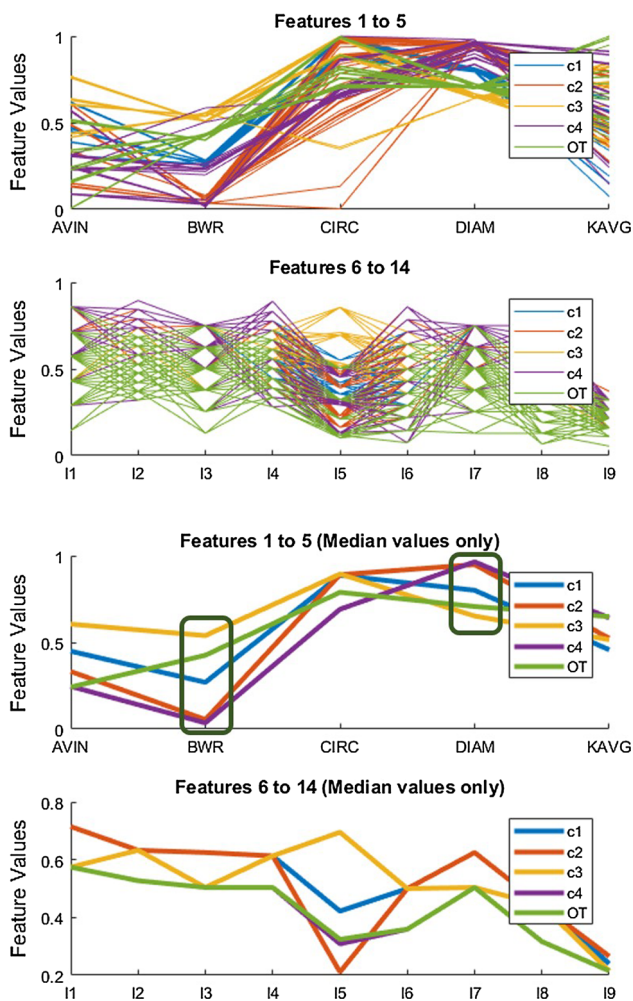


Fig. 14 Coin feature values for testing image dataset by class (with features BWR and DIAM in green) (Color figure online)

100 × 100 to 480 × 480 pixels square and (2) the part must fit inside the conditioned image with a minimum 15 pixel clearance around the outer boundary of the part, as explained in “Design of the image database” section. The size of the part in the image depends upon the working distance and the focal length of the lens. For the applications covered in this paper, this meant a part that fit within a 3 × 3 cm square with 10 cm working distance between the camera lens (of 6 mm focal length) and the part. There is no direct restriction on the physical size of the part. However, the discriminating details of the part should be visible, as dictated by the selection of an appropriate combination of camera, lens and lighting.

Comparison of non-hybrid and hybrid methods

Of the eight methods studied, the first four (M1 to M4) were non-hybrid (supervised) methods and the remaining four (M5 to M8) were hybrid (semi-supervised and supervised) meth-

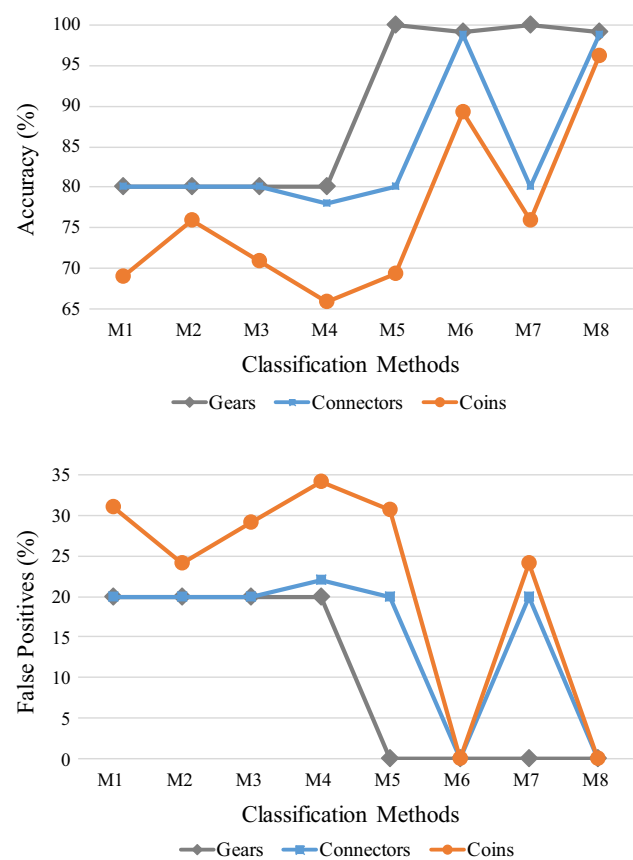


Fig. 15 Comparison between non-hybrid (M1 to M4) and hybrid (M5 to M8) methods for classes of three applications as shown in Fig. 4

ods. Figure 15 provides a comparison of the performance of the eight methods, as applied to the three applications. The first observation is that the hybrid methods provide better accuracy with fewer false positives than the non-hybrid methods. In particular, M8 (hybrid, SANN-USVM) is seen to have the best combined performance by achieving the target of zero FPs and exceeding the target accuracy of 95%. By contrast M4 (non-hybrid, SBOW) had the worst combined performance (65% accuracy with non-zero FPs). The second observation is that the best performing non-hybrid method was M2 (SANN), as compared to M1, M3 and M4. The third observation is that the worst performing hybrid method was M5 (USVM-SSVM), as compared to M6, M7 and M8.

Discussion of hybrid methods

All four possible combinations for a two layered hybrid method based on SVM and ANN were studied. First, M5 was tested and was found unable to achieve target performance for any of the three applications. Second, M6 was tested, and was found unable to achieve target performance for the coin application. At this point, M5 was tested to see if SANN could provide better results than SSVM. The answer was no,

M7 was also unable to achieve target performance. Finally, M8 was tested and found to be the only method that could achieve target performance for all three of the applications.

M8 succeeded for two reasons: (1) low intraclass variation because the USVM for M8 is trained from only one class, whereas the USVM for M5 and M7 is trained from four classes (high intraclass variation) and (2) SANN performed better than SSVM. The second reason is consistent with the results achieved by other researchers, for example Antkowiak (2006) and Ren (2012).

Speed of machine vision-based inspection systems

Two factors limit the speed of machine vision-based inspection systems: (1) time taken by the camera to acquire an image and (2) time taken by the system to classify that image. The speed for image acquisition is a hardware limitation, for a given camera. The speed of classification is a software limitation, for a given computer. For example, a 60 fps camera takes 0.02 s to acquire an image. The inspection speed of FlexMVS was found to be 400 parts/min, or 0.15 s to classify a part. Thus, with this camera, the system takes 0.17 s to acquire and process a single image. The speed of the system could be increased by reducing the number of features, but this would compromise the accuracy and flexibility of the system.

For this type of application (inspection of small parts), achievable speeds with human inspectors has historically been shown to be in the order of 30 parts/min or 2 s/part (Drury 1973). This speed is not only significantly lower than that achievable with machine vision, but the accuracy with human inspection varies from 70 to 90% and is highly dependent on the level of operator fatigue.

Inspection speed is a relative measure. Target speed in the context of medical component manufacturing is on the order of 130 parts/min or 0.46 s/part (ATS automation, 2018). Target speed in the context of high speed assembly machines is on the order of 1000 parts/min or 0.06 s/part. The limitations in the assembly case are the physical limits as to how fast a part can be moved (Shafer, 1999). Thus, the target speed for machine vision inspection applications can range from 100 parts/min (0.6 s/part) to 1000 parts/min (0.06 s/part). With this as background, a speed of 500 parts/min or 0.12 s/part seemed to be a reasonable target for the flexible machine vision system. FlexMVS currently operates at an inspection speed of 400 parts/min. This can be improved by optimizing the algorithms used for the classification methods.

Conclusions and future work

This paper presented a novel solution to the problem of small parts classification when there are unknown class

images. A hybrid SVM/ANN approach was taken that combined supervised and semi-supervised layers. Four hybrid classification methods were implemented and tested: (1) SSVN–USVM, (2) USVM–SSVM, (3) USVM–SANN and (4) SANN–USVM. A software program known as FlexMVS was developed to illustrate the hybrid approach to three different small part classification applications: (1) solid plastic gears, (2) clear plastic wire connectors and (3) metallic Indian coins. The ability of the system to work with these different applications while requiring only three user inputs, with a fixed image conditioning process and a constant number of features, is provided as evidence of the flexibility of FlexMVS. Flexibility in a MV-based system is important, in order that users can change application with minimal re-tuning of the system. The robustness of the system was demonstrated by its ability to reject unknown class images. The four methods were trained with four classes and tested with five classes, where the 5th class was considered as the unknown class. It was found that SANN–USVM gave the best results with an accuracy of over 95% for all three applications. Future work will involve further testing with different small part applications whose geometric characteristics are not as pronounced, to further confirm the flexibility and robustness of the system.

Finally, it should be noted that the image library and database used in this study has been made publically available for others who are conducting research in machine vision. <http://my.me.queensu.ca/People/Surgenor/Laboratory/Database.html> is the address at which access to the image library and database can be found.

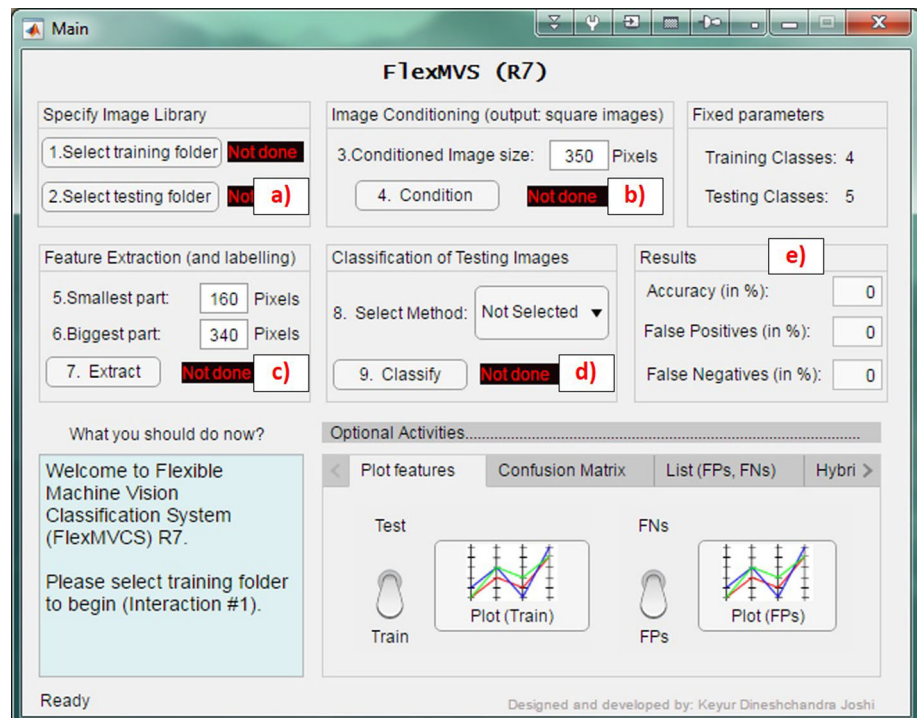
Appendix: FlexMVS software overview

FlexMVS is the software developed for small part classification using the hybrid SVM/ANN approach documented in this paper. It is written in MATLAB 2017a. The results presented in this paper were generated with FlexMVS. This appendix sets out to illustrate its ease of use, list the required actions to generate a classification run and outline the options available to the user.

The main graphical user interface (GUI) of the system is shown in Fig. 16. As keyed to the labels in Fig. 16, the required actions by the user to setup a run are:

- (a) Select path of the training and testing folder, which contains the original image database as explained in “Design of the image database” section.
- (b) Input size of the conditioned images, as per the guidelines in “Design of the image database” section. A subsequent press of the ‘Condition’ button will initiate the conditioning step and the original image database will be replaced with a conditioned image database.

Fig. 16 Main GUI of FlexMVS, with required actions labelled



- (c) Input size of the smallest and largest part, again following the guidelines in “[Design of the image database](#)” section. A subsequent press of the ‘Extract’ button will result in a prompt to the user to enter labels, which in turn will initiate the feature extraction step, as outlined in “[Feature selection and extraction](#)” section.
- (d) Select one of the four classification methods. A subsequent press of the ‘Classify’ button will initiate the classification step, as outlined in “[Classification methods](#)” section.
- (e) Once classification is complete, the results are displayed: i.e. values of accuracy, FPs and FNs. A user can select another classification method and press the ‘Classify’ button again to get a new set of results.

After completing a classification run, the user has options to investigate the results in more detail. The four options are selected by tabs given in the bottom half of Fig. 16, labelled:

1. Plot Features
2. Confusion Matrix
3. List FPs & FNs
4. Hybrid-Model

The sub window for the ‘Plot Features’ tab appears in Fig. 16. There are two switches that appear as toggle icons. The first toggle is to select training or testing dataset features as the plot. The second toggle is to select FPs or FNs as the plot. The feature plot for training and testing will be similar

to that shown in Figs. 7 and 12, respectively. The plot of FPs and FNs will be similar to the feature plots, but with fewer classes (misclassified). For example, Fig. 17 shows plots of 3.8% FNs (from CM of E9 in Table 4) for the coin application. Comparing median value plots of Fig. 17 with the median value plots of Fig. 9 clearly confirms that feature values of class C2 and C3 do not align with the median value plots of training images and hence, they were misclassified into OT class.

If the ‘Confusion Matrix’ tab is selected, the sub-window shown in Fig. 18 will appear. Pressing the ‘Compute and Show’ button will display the confusion matrix for the current run, in a format similar to Table 4.

If the ‘List FPs and FNs’ tab is selected, the sub-window shown in Fig. 19 will appear. If the ‘Show FNs’ button is pressed then a list would appear as shown in the Fig. 19, that provides name of the FNs images. If there are no FNs, a message would appear stating that ‘*No False Negatives Found: Try False Positives instead. You may have 100 percentage accuracy. OR You need to execute Classification step first*’. Similar message would appear, if there were no FPs in the classification run and the user pressed ‘Show FPs’ button.

If the ‘Hybrid-Model’ tab is selected, the sub-window shown as Fig. 20 will appear. A press of the ‘Prepare Hybrid-Model’ button will initiate the model building process. When the preparation step is complete, a press of the ‘Test Hybrid-Model’ button will open the GUI shown as Fig. 21. The user must then select a test image by pressing the ‘Browse’ button. The selected image will then appear, as shown in Fig. 22.

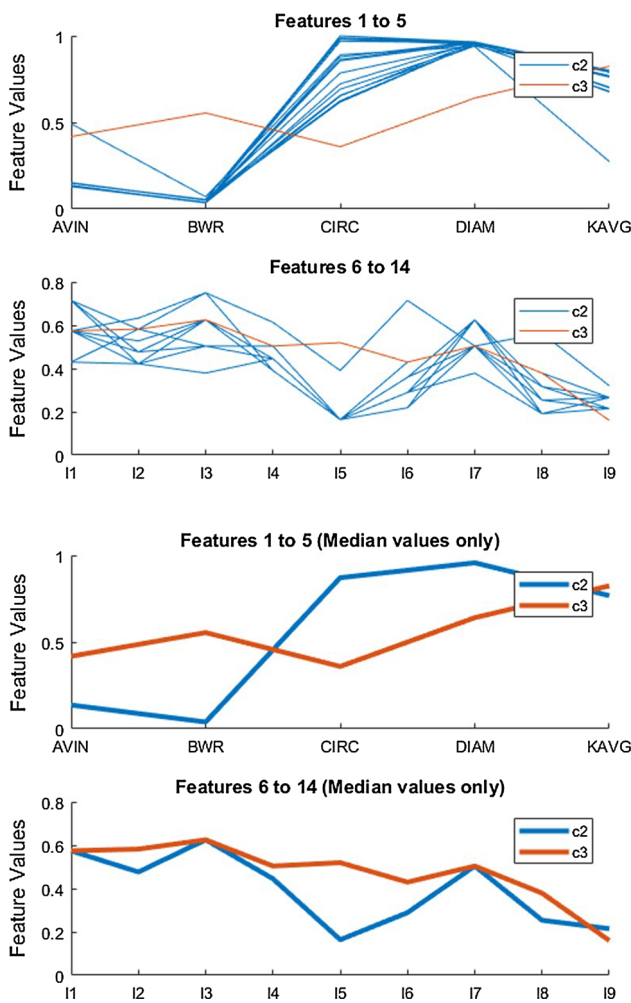


Fig. 17 FNs plot obtained from ‘Plot Features’ utility

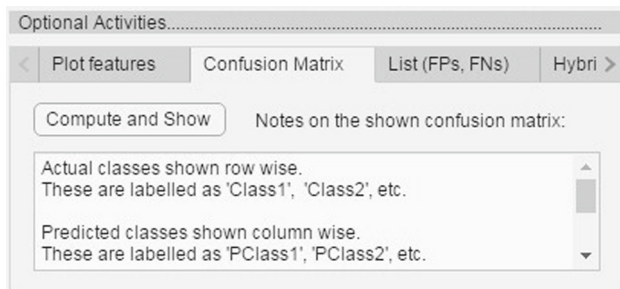


Fig. 18 Sub-window under confusion matrix tab

When the ‘Predict’ button is pressed, FlexMVS will apply the hybrid-model’s algorithms to the selected test image and will provide a decision regarding the class of the image. For example, Fig. 22 shows the selected coin image as Class C3 because its features were in line with the training features of the Class C3. In the example of Fig. 23, the selected ‘non-Indian (Canadian)’ coin image was assigned to OT class

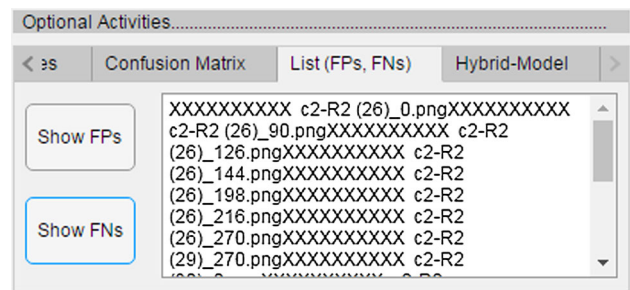


Fig. 19 Sub-window under List FPs and FNs tab

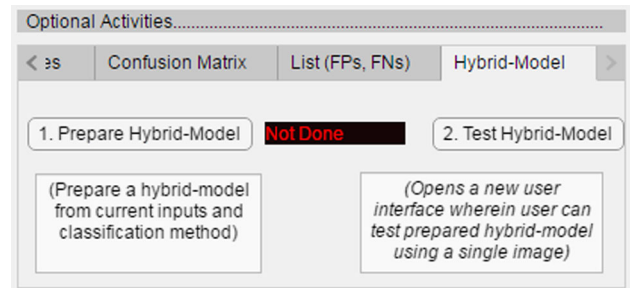


Fig. 20 Sub-window under hybrid-model tab

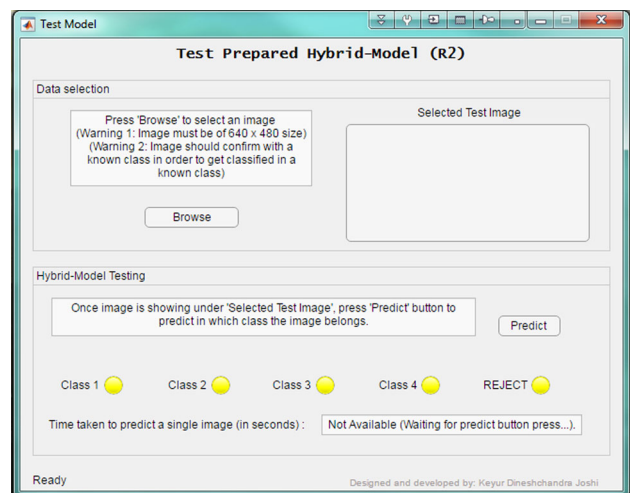


Fig. 21 GUI for test hybrid-model before image selection

because in this example, FlexMVS had been trained on the Indian coin database.

As a final point, Figs. 22 and 23 shows the time taken to classify the single test image (0.14776 s in the case of the Indian coin and 0.14969 s in the case of Canadian coin example). The time of 0.15 s to classify a single part image is equivalent to an inspection rate of 400 parts per min. This provides a baseline for the minimum production rate.

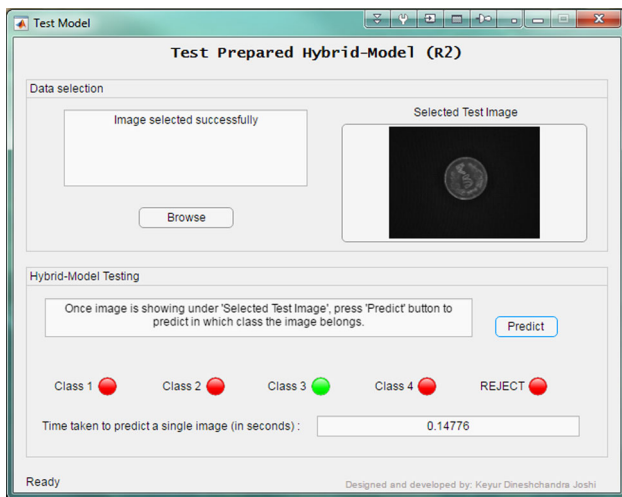


Fig. 22 GUI for test hybrid-model after ‘Predict’ is pressed

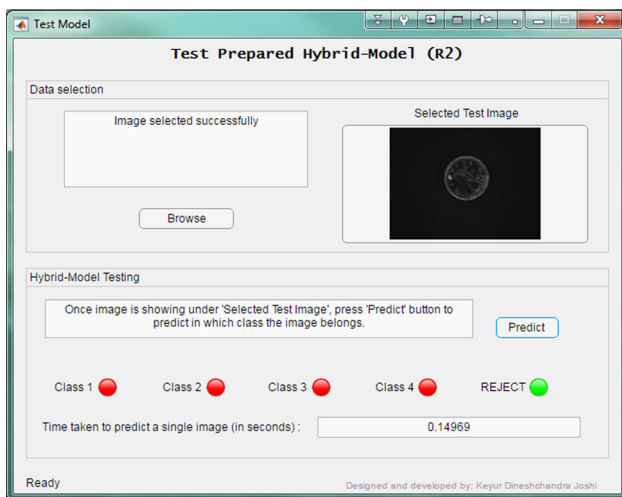


Fig. 23 GUI for test hybrid-model of an OT class

References

- Abdullah, M., Aziz, S., & Mohammad, A. (2000). Quality inspection of bakery products using a color-based machine vision system. *Journal of Food Quality*, 23(1), 39–50.
- Akhtar, A., Khanum, A., Khan S.A., & Shaikat, A. (2013). Automated plant disease analysis (APDA): Performance comparison of machine learning techniques. In *11th International conference on frontiers of information technology* (pp. 61–65). Islamabad, Pakistan.
- Antkowiak, M. (2006). Artificial neural network vs. support vector machines for skin diseases recognition. *Master Thesis, Department of Computing Science, Umea University, Sweden*.
- Atherton, T. J., & Kerbyson, D. J. (1999). Using phase to represent radius in the coherent circle hough transform. *Image and Vision Computing*, 17, 795–803.
- ATS automation (2018). High speed vision inspection success story. resource. https://www.atsautomation.com/en/Life_Sciences/Success_Stories/High_Speed_Vision_Inspection.aspx Accessed 26 Feb 2018.
- Batchelor, B. G. (2012). *Machine vision for industrial applications. machine vision handbook, Chapter 1* (pp. 1–59). London: Springer.
- Bianchini, M., Scarselli, F. (2014). On the complexity of shallow and deep neural network classifiers. In *European symposium on artificial neural networks, computational intelligence and machine learning (ESANN)* (pp. 371–376). Bruges, Belgium.
- Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97, 245–271.
- Cao, L., Xiang, M., Feng, H., & Wang, Y. (2015). Size sorting and measurement system of safety belt pin based on machine vision. *Applied Mechanics and Materials*, 741, 709–713.
- Chauhan V., Joshi K. D., Surgenor B. (2017) Machine vision for coin recognition with ANNs: Effect of training and testing parameters. In: G. Boracchi, L. Iliadis, C. Jayne, A. Likas (eds.) *Engineering applications of neural networks*. EANN 2017. Communications in computer and information science (vol 744). Springer, Cham.
- Chauhan, V., & Surgenor, B. (2015). A comparative study of machine vision based methods for fault detection in an automated assembly machine. *Procedia Manufacturing*, 1, 416–428.
- Chen, S. H., & Perng, D. B. (2016). Automatic optical inspection system for IC molding surface. *Journal of Intelligent Manufacturing*, 27(5), 915–926.
- Chetima, M. M., Payeur, P. (2008). Feature selection for a real-time vision-based food inspection system. In *IEEE international workshop on robotic and sensors environments* (pp. 120–125). Ottawa, Canada
- Chetima, M. M., Payeur, P. (2012). Automated tuning of a vision-based inspection system for industrial food manufacturing. In *Instrumentation and Measurement Technology Conference (I2MTC)* (pp. 210–215) Graz, Austria.
- Cooray, T., Fernando, S. (2011). Visual based automatic coin counting system. In: SAIM research symposium on engineering advancements (pp. 55–58). Malabe, Sri Lanka.
- Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence*, 151, 155–176.
- Drury, C. G. (1973). The effect of speed of working on industrial inspection accuracy. *Applied Ergonomics*, 4(1), 2–7.
- Fukumi, M., Omatu, S., Takeda, F., & Kosaka, T. (1992). Rotation-invariant neural pattern recognition system with application to coin recognition. *IEEE Transactions on Neural Networks*, 3(2), 272–279.
- Hosseininia, S., Khalili, K., Emam, S. (2016). Flexible automation in porcelain edge polishing using machine vision. In *Procedia Technology, and 9th international conference interdisciplinarity in engineering (INTER-ENG 2015)* (vol. 22, pp. 562–569). Tirgu-Mures, Romania.
- Hua, J., Xiong, Z., Lowey, J., Suh, E., & Dougherty, E. (2005). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8), 1509–1515.
- Huang, S., & Pan, Y. (2015). Automated visual inspection in the semiconductor industry: a survey. *Computers in Industry*, 66, 1–10.
- Joshi, K. D. (2018). A flexible machine vision system for small parts inspection based on a hybrid SVM/ANN approach. *Doctoral Thesis, Department of Mechanical and Materials Engineering, Queen's University, Canada*.
- Joshi, K. D., Chauhan, V. and Surgenor, B. (2016). Real time recognition and counting of Indian currency coins using machine vision: a preliminary analysis. In *Proceedings of the canadian society for mechanical engineering international congress*. Kelowna, Canada.
- Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6), 1153–1160.

- Kim, T. H., Cho, T. H., Moon, Y. S., & Park, S. H. (1999). Visual inspection system for the classification of solder joints. *Pattern Recognition*, 32(4), 565–575.
- Leemans, V., Magein, H., & Destain, M. F. (2002). On-line fruit grading according to their external quality using machine vision. *Biosystems Engineering*, 83(4), 397–404.
- Li, Z., Chang, S., Liang, F. (2013). Learning locally-adaptive decision functions for person verification. In: *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3610–3617). IEEE Computer Society, Portland, USA.
- Li, X., Guo, Y. (2013). Adaptive learning for image classification. In *IEEE conference on computer vision and pattern recognition* (pp. 859–866). IEEE Computer Society, Portland, USA.
- Malamas, E., Petrakis, E., Zervakis, M., Petit, L., & Legat, J. (2003). A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 21(2), 171–188.
- Modi, S. (2011). Automated coin recognition system using ANN. Master's Thesis, Thapar University, Computer Science and Engineering Department, Patiala, India.
- Modi, S., & Bawa, S. (2011). Automated coin recognition system using ANN. *International Journal of Computer Applications*, 26(4), 13–18.
- Modi, S., & Bawa, S. (2012). Image processing based systems and techniques for the recognition of ancient and modern coins. *International Journal of Computer Applications*, 47(10), 1–5.
- Nashat, S., Abdullah, A., Aramvith, S., & Abdullah, M. Z. (2011). Support vector machine approach to real-time inspection of biscuits on moving conveyor belt. *Computers and Electronics in Agriculture*, 75(1), 147–158.
- Nerakae, P., Uangpairoj, P., Chamniprasart, K. (2016). Using machine vision for flexible automatic assembly system. In *Procedia computer science, and 20th international conference on knowledge based and intelligent information and engineering systems (KES2016)* (Vol. 96, pp. 428–435), York, UK.
- Nguyen, A., Yosinski, J., Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Conference on Computer Vision and Pattern Recognition* (pp. 427–436). Boston, USA.
- Nielsen, M. A. (2015). *Neural networks and deep learning. Electronic book*. Oxford: Determination Press.
- Niklaus, P., Ulli, G. (2015). Automated resistor classification. Group Thesis, Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory, Zurich, Switzerland.
- Nilsback, M. E., Zisserman, A. (2006). A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition* (Vol. 2, 99. 1447–1454). IEEE Computer Society, New York, USA.
- Otsu, N. (1979). A Threshold selection method from grey-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1), 62–66.
- Park, H. K. (2015). Machine vision system for effective semiconductor package sorting. *Advanced Science and Technology Letters*, 120, 408–411.
- Penaranda, J., Briones, L., & Florez, J. (1997). Colour machine vision system for process control in ceramics industry. *Lasers and Optics in Manufacturing*, 3, 182–192.
- Piementel, M. A., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99, 215–249.
- Ren, J. (2012). ANN vs. SVM: which one performs better in classification of MCCs in mammogram imaging. *Knowledge-Based Systems*, 26, 144–153.
- Rosati, G., Facciao, M., Carli, A., & Rossi, A. (2013). Fully flexible assembly systems (F-FAS): A new concept in flexible automation. *Assembly Automation*, 33(1), 8–21.
- Schlipfing, M., Salmen, J., Tschentscher, M., & Igel, C. (2014). Adaptive Pattern Recognition in Real-Time Video-Based Soccer Analysis. *Journal of Real-Time Image Processing*, 19, 1–17.
- Schmidhuber, J. (2014). Deep learning in neural networks: An overview. University of Lugano and SUPSI, The Swiss AI Lab IDSIA, Switzerland.
- Schoonahd, J. W., Gould, J. D., & Miller, L. A. (2007). Studies of Visual Inspection. *Ergonomics*, 16(4), 365–379.
- Shafer, D. A. (1999). *Successful assembly automation. Defining the assembly process* (pp. 47–76). Dearborn: Society of Manufacturing Engineers.
- Shen, H., Li, S., Gu, D., & Chang, H. (2012). Bearing defect inspection based on machine vision. *Measurement*, 45(4), 719–733.
- Sokolava, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45, 427–437.
- Su, J. C., & Tarnq, Y. S. (2008). Automated Visual inspection for surface appearance defects of varistors using an adaptive neuro-fuzzy inference system. *International Journal of Advanced Manufacturing Technology*, 35, 789–802.
- Sun, T. H., Tien, F. C., Tien, F. C., & Kuo, R. J. (2016). Automated thermal fuse inspection using machine vision and artificial neural networks. *Journal of Intelligent Manufacturing*, 27(3), 639–651.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan D., Vanhoucke V., Rabinovich, A. (2015). Going deeper with convolutions. In *Conference on computer vision and pattern recognition (CVPR)*, 1–9, Boston, USA.
- Tapilouw, A., Chang, Y., Liu, H., Wang, H., Tai, H. (2015). White light triangulation sensor for flexible inspection system. In: *9th international conference on sensing technology*, IEEE (pp. 765–768). Auckland, New Zealand.
- Tessier, J., Duchesne, C., & Bartolacci, G. (2007). A Machine vision approach to on-line estimation of run-of-mine ore composition on conveyor belts. *Minerals Engineering*, 20, 1129–1144.
- Tuytelaars, T., & Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3), 177–280.
- Vapnik, V., Steven, E., & Smola, A. (1996). Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems*, 9, 281–287.
- Velu, C., Vivekanandan, P., & Kashwan, K. (2011). Indian coin recognition and sum counting system of image data mining using artificial neural networks. *International Journal of Advanced Science and Technology*, 31, 67–80.
- Wang, Q., Ma, L., Gao, Q., Li, Y., Huang, Y., & Liu, Y. (2017). Adaptive maximum margin Analysis for Image Recognition. *Pattern Recognition*, 61, 339–347.
- Weigl, E., Heidl, W., Lughofer, E., Radauer, T., & Eitzinger, C. (2016). On improving performance of surface inspection systems by online active learning and flexible classifier updates. *Machine Vision and Applications*, 27, 103–127.
- Wilder, J. (1989). Industrial applications of machine vision. In *Issues on machine vision, CISM courses and lectures No. 307, international centre for mechanical sciences* (pp. 311–339).
- Wu, W., Wang, X., Huang, G., & Xu, D. (2015). Automatic gear sorting system based on monocular vision. *Digital Communications and Networks*, 1(4), 284–291.
- Yan, M., Surgenor, B. (2011). A quantitative study of illumination techniques for machine vision based inspection. In *ASME 6th Manufacturing science and engineering conference (MSEC2011-50178)* (pp. 281–288). Oregon, USA.