Fault detection and classification in automated assembly machines using machine vision

Vedang Chauhan1 · Brian Surgenor1

Abstract Automated assembly machines operate continuously to achieve high production rates. Continuous operation increases the potential for faults such as jams, missing parts, and electromechanical failures of subsystems. The goal of this research project was to develop and validate a machine vision inspection (MVI) system to detect and classify multiple faults using a single camera as a sensor. An industrial automated O-ring assembly machine that places O-rings on to continuously moving plastic carriers at a rate of over 100 assemblies per minute was modified to serve as the test apparatus. An industrial camera with LED panel lights for illumination was used to acquire videos of the machine’s operation. A programmable logic controller (PLC) with a human-machine interface (HMI) allowed for the generation of faults in a controlled fashion. Three MVI methods, based on computer vision techniques available in the literature, were developed for this application. The methods used features extracted from the videos to classify the machine’s condition. The first method was based on Gaussian mixture models (GMMs); the second method used an optical flow approach; and the third method was based on running average and morphological image processing operations. In order to provide a single metric to quantify relative performance, a machine vision performance index (MVPI) was developed with five measures of performance: accuracy, processing time, speed of response, robustness against noise, and ease of tuning. The MVPI for the three MVI methods is reported along with the significance of the results.

Keywords Machine vision · Automated assembly machine · Fault detection and classification · Performance index

1 Introduction

The use of machines for automated assembly enabled modern manufacturing industries to achieve a high production throughput and to gain competitive advantages. To accomplish these goals, automated machines are operated around-the-clock. Continuous operation of an assembly machine results in wear of its various mechanisms that in turn lead to machine faults such as part jams, missing parts in the assembly, misalignments, and blockages with subsequent machine downtime. When machine failure occurs, a small portion of downtime is spent for actual repairing, while the majority of time is consumed in locating the source of the problem [1]. Even with the state of the art technology, it is hard to prevent the machines from having faults such as missing parts in assemblies, part jams, misalignments, and blockages. Early fault detection and classification can help to restore a machine to its online state in the least possible time [2]. Many researchers, from different fields, have studied and developed various fault detection and classification methods with different applications. Therefore, it is important to define some terminology for fault detection, classification, and diagnosis that are used in this research work. Detection is the first stage where a fault is recognized as soon as it occurs. Classification is the second stage that deals with detecting the type and location of the fault. Further information about the fault is obtained in the third stage where diagnosis determines the causes of the fault. This research work is concerned with the development of a
fault detection and classification system as applied to an automated O-ring assembly machine. The goal of this research work was to develop and validate a machine vision inspection (MVI) system to detect and classify multiple faults using a single camera as a sensor.

Traditional methods of fault detection in machines are based on readings from sensors (such as limit switches, proximity, potentiometers, pressure sensors, and current and voltage sensors) and monitoring them with limit checking, where a fault is detected when a sensor output exceeds either upper or lower threshold limits. More recent methods are model-based methods, where machine input and output signals are used to generate a mathematical model of a process to determine the occurrence of faults. These methods are subjected to limitations in terms of added cost due to the need for multiple sensors and detecting a fault only after it exceeds the threshold, which might take a long time. Furthermore, it may not provide proper information for fault diagnosis. Therefore, a faster automatic fault detection and classification method is needed to minimize the delay time in taking corrective action, once a fault is detected.

MVI systems are popular for finished product inspection applications, robot guidance, object tracing, etc. These systems use industrial grade cameras to acquire data in the form of images and videos. A camera can be used for continuous video acquisition of the machine’s operation and computer vision techniques could be used to develop a fault detection system that is non-intrusive, requires less processing time, and can adapt to changing operating conditions. MVI systems are of three types: PC-based, vision sensor-based, and smart camera-based. The PC-based systems use a digital camera that acquires images and sends them to a PC that runs machine vision inspection algorithms. This type of MVI systems is relatively slow due to time required to send image data to the PC. The vision sensor-based systems use a vision sensor that has its own processor and it can do a limited set of predefined inspection tasks. The vision sensor uses proprietary software for application development. The smart camera-based systems use a camera with a processor, memory, digital I/Os, and Ethernet connectivity for remote monitoring and control purposes. The smart camera provides a greater flexibility compared to a vision sensor, and it can do multiple tasks such as image acquisition, processing, and controlling other peripheral systems using its I/Os. In the context of this research, a PC-based MVI system for fault detection and classification on an automated assembly machine was designed and developed.

2 Related work

Development of a fault detection and classification system for automated assembly machines is the subject of this research paper. A number of researchers have worked in the general area of fault detection and diagnosis in machines. During the literature review, several papers were found on the topic of machine condition monitoring and fault detection in assembly automation. A selection of both the non-machine vision and machine vision based papers will be reviewed in this section.

Traditional methods of fault detection in machines are based on readings from sensors (such as limit switches, proximity, potentiometers, pressure sensors, and current and voltage sensors) and monitoring them with limit checking, where a fault is detected when a sensor output exceeds either upper or lower threshold limits. More recent methods are Petri net-based [3], model-based [1], and parameter estimation-based [4] methods. Researchers have also used signal-processing methods for fault detection purposes [5, 6]. Decision trees and artificial neural networks (ANNs) are widely used for fault detection and diagnosis in assembly automation. The ROBODOC, a decision tree-based generalized system for fault diagnosis and maintenance of automated systems was proposed [7]. The approach used DCLASS, a group technology method for decision making and classification of faults and symptoms. Demetgul et al. [8, 9] implemented two ANNs: adaptive resonance theory (ART) and backpropagation (Bp), for fault detection in a pneumatic modular production system. Both ANNs were trained using the data collected from eight different sensors for both normal and faulty operations and were able to correctly classify the faults using the trained networks. Fernando and Surgenor [10] developed a system using three grayscale sensors and two limit switches for fault detection and identification in automated assembly machines. Result showed that the rule-based system was more useful when unknown or multiple faults were presented.

Sekar et al. [11, 12] proposed an e-diagnostic approach for programmable logic controller (PLC)-based automated assembly systems. The study showed that there was no significant difference in terms of overall troubleshooting performance between an expert engineer and a novice operator. MVI based papers are discussed in the rest of the section.

It has been suggested that machine vision technology can help industry gain a competitive advantage in terms of a better product quality, high customer satisfaction, less inspection time, and improved productivity [13]. MVI systems use a digital camera and image processing software for an inspection of a product or process. Web-based remote monitoring, control and diagnosis of manufacturing processes using digital cameras, and a PLC and PC-based human-machine interface (HMI) were investigated [14].

ANNs are often used for classification and pattern recognition problems using MVI systems. A machine vision system for the detection of missing fasteners on steel stampings was proposed [15]. A neuro-fuzzy image classification algorithm was developed and compared with a threshold-based classifier. It was reported that the neuro-fuzzy classifier performed better than the threshold-based classifier. MVI systems are
often used for detection of unusual events in a scene such as event detection in crowded videos [16, 17]. Hughes et al. [18] examined the application of a video event detection method, based on spatiotemporal volumes (STVs), for fault detection in automated assembly machines. A trained model STV was modeled using a set of normal operation sequences. New STVs, for both normal and faulty operation sequences, were built and compared with the trained model STV and classified into an appropriate category using a distance measurement. Automatic detection of stamping defects in lead frames using machine vision was implemented [19]. The system was capable of detecting defects from both continuous reel and individual cut lead frames with rotational misalignment of ±10°. Shahabi and Ratnam [20] proposed a machine vision system for in-cycle monitoring of tool nose wear and surface roughness of turned parts. The maximum deviation of 10 % was recorded between the inspection using the MVI system and a stylus.

Xiaokun and Porikli [21] presented a novel approach to automatically detect highway traffic events, such as heavy congestion, high vehicle density with high speed, vacancy, traffic jam, etc., using a MVI system. The algorithm classified traffic event using a Gaussian mixture hidden Markov model (GMHMM). The Gaussian mixture model (GMM) is a statistical technique of clustering data using probability density estimations. The model detected events in real time with an accuracy of 94 %. Zezhi et al. [22] used Gaussian mixture models for segmentation of moving road vehicles from the colored video data acquired using CCTV. These applications had demonstrated that GMMs combined with blob analysis had the potential to solve O-ring machine fault detection problem. Transfer track jam on the machine has similarity with traffic jam on highways and missing hopper fault is similarly to vacancy on highways.

Optical flow is used for motion estimation from a video. It was first introduced by Horn and Schunck [23] with the examples of a rotating sphere and a cylinder on its axis. Patel and Shukla [24] implemented a vehicle-tracking algorithm using optical flow. The optical flow was used to segment an image frame for a vehicle detection. The segmented vehicle was then tracked, and the velocity of the vehicle was determined by calculating the distance that the object moved in a sequence of frames. This application had shown the potential that optical flow can be used for moving object detection and motion estimation. In case of the O-ring machine, carriers and O-rings followed specific path on the machine. The flow estimation of that path using optical flow technique would help in fault detection and classification. However, no publications were found that used optical flow approach for fault detection in assembly machines.

Usamentiaga et al. [25] proposed a system to detect jams in a steel processing line using a MVI system. The proposed MVI system acquired images from the line, processed them, and extracted features that were the measure of the density of the number of pieces ejected from the side trimmers. These features were used to successfully detect jams in two nozzles using the running average method. The MVI system was effective in detecting jam in the processing line during the 14 months of operation. This application had demonstrated that fundamental image processing techniques had the potential to solve the O-ring machine inspection problem. Fault detection in an automated assembly machine using three MVI methods was studied [26]. This paper is an extension of that work with the addition of fault introduction system, detailed explanation of fault detection approach, and performance measurement criteria.

3 Automated O-ring assembly machine

Fault detection and classification with a MVI system for automated assembly machines are carried out on a modified assembly machine as shown in Fig. 1. The main parts of the machine are two rotating transfer wheels (primary wheel and secondary wheel) that hold empty carriers and assemblies, a vibrating hopper as the supply of O-rings, a feed chute, two air transfer tracks (transfer 1 and transfer 2), an air knife above the primary wheel for removal of unassembled O-rings, and a vacuum system for collection of excess O-rings and assembled O-rings. The digital camera and four LED panel lights are also shown in the figure. As mentioned in the previous section, the machine was originally designed as a part of the larger machine, where single O-ring in their carriers were used for the next stage in the assembly procedure for coaxial cable. For the purposes of this research, the machine had to be modified so that O-rings could be separated from their carriers at the end of the assembly cycle and returned to the hopper. With this modification, the machine could be run in a continuous fashion. The primary wheel has 16 positions to hold carriers and assemblies. The maximum rotational speed of the machine is 6.8 RPM. Hence, the maximum rate of assembly is 108 assemblies per minute. The following paragraph explains the sequence of operations for normal assembly cycle.

The machine assembles black colored O-rings into continuously moving, white, circular carriers as shown in Fig. 2. The rate of assembly is 108 per minute at the maximum rotational speed of the machine. During operation, as the hopper vibrates, a steady stream of O-rings fall onto the primary wheel through the feed chute. The primary wheel has 16 slots to hold and transfer assemblies. As the O-rings fall from the feed chute, the aligning pins located beneath the primary wheel are raised and used to align the fallen O-ring onto the carrier. Here, a single O-ring is assembled onto the circular groove of a carrier. The primary wheel rotates at a rate of around 6.8 RPM that gives the linear velocity of 135 mm/s to the assemblies. The excess O-rings that are not picked up by a carrier
fall onto the primary wheel and are blown off into the collection bin by an air knife; where they are vacuum sucked and returned to the hopper. In the next stage of the assembly process, the carriers (each holding a single O-ring) are transferred, one at a time, to the secondary wheel via the air transfer track 1 (transfer 1). The secondary wheel rotates the carriers to the collection vacuum valve. At this point, the O-rings are vacuumed off of the carriers and returned to the hopper. The empty carriers are then returned to the primary wheel via the air transfer track 2 (transfer 2), and the cycle is repeated. With this modification the machine could be run in a continuous fashion.

The machine uses a pneumatic system for vacuum suction and for the transfer of assemblies and empty carriers. The air for the pneumatic system is supplied by multiple compressors through a manifold. There are six electro-pneumatic valves, two for transfer tracks, two for air knives, and two for vacuum valves along with two Line Vac valves for the collection vacuum. The basic sequences of operation are that (1) empty carriers appear on the primary wheel, (2) O-rings assembled onto the carriers with aligning pins, (3) assemblies transferred onto the secondary wheel, and (4) O-rings recirculated back to the hopper. Any deviation from the normal sequence should be considered as a fault in the assembly cycle. During the normal cycle, assemblies move at different speeds at different locations on the machine. The assemblies move at the highest speed (0.6 m/s) through the pneumatic transfer tracks; O-rings move at a relatively slow speed onto the primary wheel (0.09 m/s); and O-rings fall from the hopper at a moderate speed (0.33 m/s). Under ideal circumstances, the machine should continuously assemble O-rings at the desired rate (108 assemblies/min).

3.1 Machine faults

During several runs of the machine for an extended period of time, a set of commonly occurring faults was observed. These faults occurred due to friction in the tracks, lack of O-rings flow from the hopper due to low vibrations and intermittent flow of O-rings, or improper operation of the air knives due to air pressure drop. Some of these faults could have serious consequences in terms of the machine’s condition, while other faults might have resulted in the lack of assemblies. The transfer track faults (transfer 1 jam and transfer 2 jam) were serious faults because they could damage the machine if immediate corrective action was not taken. On the other hand, a lack of O-rings flow only results in missing O-rings on the carriers. This fault could have not damaged the machine but simply...
resulted in incomplete assemblies. These faults are grouped into three region of interests (ROIs) with respect to the field of view (FOV) of the camera as shown in Fig. 3. In the first stage of experimental video data collection, efforts were made to minimize the frequency of these faults. In other words, there was a need to run the machine under a normal operating cycle to get a measure of how well the machine ran under near fault free conditions. In the second stage of video data collection, an Allen Bradley MicroLogix 1400 PLC (24 VDC, 20 inputs and 12 outputs) with a Panel View C600 HMI was used to introduce the controlled faults to acquire the videos for faulty operations and to communicate the MVI system decisions to the machine.

The types and locations of the adopted set of four faults are shown in Fig. 4. The faults were as follows: transfer tracks 1 and 2 jams, the air knife fault, and the hopper fault. These faults were grouped into three ROIs: transfer track ROI, air knife ROI, and hopper ROI. Transfer track jams can occur due to low air pressure in the transfer track air nozzle, friction in the tracks itself, or unassembled O-rings being stuck in the narrow passage of the tracks. The air knife fault can occur due to inadequate air pressure that fails to remove excess unassembled O-rings from the primary wheel. The lack of a continuous flow of O-rings from the hopper can occur due to friction in the hopper chute, insufficient amount of O-rings in the hopper, or low hopper vibration amplitude. The hopper fault results in incomplete assemblies at the feed chute.

4 MVI system and data collection

The elements of a MVI system are as follows: camera with lens, lighting, image processing hardware, computer with image processing software, and a PLC for communicating the decision. The first step in the design of a MVI system is the selection of application specific camera lens and lights. The quality of the original images is very important for any MVI system. There are two ways to get high quality images: one way is a hardware fix, where the camera-lens and light settings are optimized to enhance the required features in the image; the second way is a software fix, where various image pre-processing techniques are used to improve the quality of the image. The efforts required for image enhancement after image acquisition depend greatly on the quality of the original images. Even with the help of state of the art image enhancement algorithms, it is hard to improve the quality of poorly acquired images. Selection of camera-lens and lights requires several factors to be considered. The most important point to remember is that MVI systems are application specific, no single camera-lens, and light combination can be used to solve all MVI problems.

4.1 Camera and lens selection

The camera and lens combination is selected first in the design of the MVI system. The camera selection factors are shown in Fig. 5.

The factors considered are nature of the digital interface, required resolution, sensor format, colors and type of sensor, and a frame rate. Commonly preferred digital interfaces for industrial machine vision solutions are IEEE Firewire, Gigabit Ethernet, and USB 2.0/3.0. IEEE Firewire has advantages such as low CPU load and low jitter, but it has limitations of low bandwidth (32 MB/s for IEEE 1394a and 64 MB/s for IEEE 1394b). Gigabit Ethernet can support a long cable length (up to 100 m) and has an easy infrastructure setup for multi-camera use. However, it has a high CPU load and price. USB 2.0/3.0 has become a standard in the consumer market, and a lot of hardware supports it. Compared to USB 2.0, USB 3.0 offers nine times more bandwidth (up to 350 MB/s), better error management, higher power supply, and longer cable lengths (up to 8 m). USB cameras have less CPU load compared to Gigabit Ethernet. Therefore, a USB 3.0 digital interface was preferred for the O-ring machine MVI camera.
Sensor resolution for MVI cameras can be broadly categorized into three segments: (1) less than 1.0 MP, (2) between 1.0 and 2.0, and (3) more than 2.0 MP. The higher resolution images look better but they need more processing time by the image processing software. A camera with more than 2.0 MP resolution would be appropriate for the project. Sensor format along with resolution decides the quality of image. A bigger sensor is better, but it is more expensive as well. It was decided that a sensor with size 1/1.8" would fit the project requirement.

A frame rate should be at least 10 times the rate of assembly. The O-ring machine assembles at a maximum rate of 108 assemblies/min or 1.8 assemblies/s. Therefore, the frame should be at least 18 fps. The nearest available standard frame rate was 30 fps, and it was selected for the project. The next step after the camera selection was setting the appropriate camera resolution, mounting the correct size lens and fixing the camera at a proper working distance. A single camera was used to detect multiple faults in the MVI system. The required FOV for the camera is 710 mm × 381 mm. To achieve this FOV, a camera was mounted at the appropriate working distance (WD; distance from the front of the lens to the object under inspection) above the machine such that it can view this entire area. The WD is calculated using Eq. (1). It depends on the focal length (FL) of the lens, maximum dimension of the FOV, and corresponding camera sensor size (7.18 mm).

\[
WD = \frac{\text{focal length} \times \text{FOV}}{\text{sensor size}}
\]  

On the O-ring assembly machine, the WD was limited to a maximum of 1200 mm. A WD above this value caused interference in the FOV by the hopper. A lens with a short focal length is required to achieve the required WD.
length caused distortions at the edges of the image. A suitable focal length was judged to be between 12 and 16 mm. A lens with 8.5 mm focal length and 2/3″ format size was selected because it was a larger format lens for the smaller format camera which gave an equivalent focal length of 12 mm. The use of a larger format lens for the smaller format camera helped to reduce distortions in images at the corners because only the central portion of the view was captured by the camera sensor. Although, this lens and camera combination was not able to view the full FOV, it did cover the area of the machine that was needed to detect and classify the faults. In conclusion, the WD based on the available camera sensor sizes and lens focal lengths was calculated to be 1.1 m for a maximum FOV value of 710 mm.

Considering all the above factors and the project requirements, a Grasshopper3 2.8 MP Color USB3.0 Vision camera with 1/1.8″ sensor and a resolution of 1928 × 1448 at 26 fps from Point Grey Research Inc. was used as the MVI system camera.

4.2 Light selection

Light from the environment is undesirable for machine vision applications as it varies throughout a day. It is a common requirement for a MVI system to give accurate and precise results at any time irrespective of changes in the environment light [27]. Therefore, MVI systems need controlled lights which can accentuate required features and minimize undesired features. The good performance of a vision application requires non-varying lighting conditions. Kopparapu [28] presented an approach to obtain uniform illumination on the scene being imaged using several light sources. The use of correct lighting is crucial for the application related to surface parameter estimation with machine vision. Specular reflections from metallic surfaces can also significantly degrade the quality of the images.

![Fig. 6 Light selection factors](image-url)

Light selection is dependent upon the part size, part color, surface features, geometry, inspection environment, and the system needs. A full understanding of the immediate inspection area and the application requirements is the prime step to be performed. Once a rigorous lighting analysis is done based on the above factors, a decision can be made to select a light solution from available options. Five factors, as shown in Fig. 6, play an important role in the light selection based on a given application [29].

Due to flexibility, an extremely long lifespan, stability, low heat generation, and cost-effectiveness, LEDs are the first choice. Diffuse lighting minimizes glare and sensitivity to the surface angles on the parts. The O-ring machine had metallic surfaces that caused specular reflections. Hence, a diffuse lighting technique positioned as dark-field light was preferred for the illumination. A TIFFEN circular polarizer with size 40.5 mm was used to further minimize the bright reflections from the shiny wheel surfaces on the O-ring machine. By considering the above factors, four Aputure Amaran AL-528 LED panel lights with adjustable intensity were selected as the lighting source for the MVI system.

The automated assembly machine had both dull (plastic) parts and shiny reflective (aluminum wheel) surfaces. The specific challenges were the small size of the O-rings, the influence of the environmental light, the holes in the carrier, and the shiny nature of the surface of the metallic primary and secondary wheels. Using the guidelines in the previous section, LED panel lights were selected for illumination. The position of the lights and camera relative to the FOV was the next important factor to be considered. Hence, an algorithm was tested with different lighting positions. Once a camera view and the lens focal length were set, tests were conducted under six different lighting conditions, namely, (1) ambient light, (2) diffuse light, (3) bright-field light with a diffuser, (4) bright-field light with a polarizer, (5) dark-field light with a diffuser, and (6)
dark-field light with a polarizer. A foam board was fixed under the wheels to get the effect of backlight for O-rings detection though the holes in the carriers. This resulted in the O-rings clearly appearing as black circles on a white background. Carriers and O-rings detection from images acquired under an ambient light and dark-field light with a polarizer are shown in Fig. 7. The effectiveness of the illumination was measured in terms of a detection ratio. It is the ratio of the number of objects detected from the image to the actual number of objects present in the image. The ratio was calculated for both carriers and O-rings separately. The results of the tests are presented in the form of a bar chart shown in Fig. 8. The best results (100% detection ratio for the carriers and 50% detection ratio for the O-rings) were obtained with the dark-field light and the polarizer.

As a final setup for MVI system, four LED panel lights were placed around the machine to obtain a dark-field lighting effect for the wheel surfaces and with the foam board under the wheels to obtain a backlight effect for the holes in the carriers. The camera was mounted at 1.1 m above the wheel surfaces and had the polarizer above the lens. Figure 1 illustrates the adopted positions for the four LED panel lights.

5 Fault detection and classification approach

Fault detection with a MVI system for automated assembly machines is carried out on a modified O-ring assembly machine. A set of four individual faults has been selected for testing purposes. The MVI system should detect and classify these faults from their video data only. The locations of the faults along with their corresponding three ROIs are shown in Fig. 3. The camera was mounted 1.1 m above the surface of the machine in order to cover the entire machine surface within the FOV of the camera. The machine was run for normal operation, and videos were recorded. The faults were introduced by the HMI-PLC, and videos were recorded for each faulty condition. During video acquisition of the data for the faults, the machine was first run under normal operation for up 200 frames and then the fault was introduced. The 200 frames recorded one full rotation of the primary and secondary wheels. The cycle of assembly events was repeated after one full rotation. The frame number at the event of the fault was recorded. The video datasets were acquired with the high-speed USB 3.0 camera, and MATLAB was used for the video data acquisition.

The video data acquisition was performed for both normal and faulty operation of the machine. For each condition, five videos were recorded. Both normal and fault condition videos were recorded for each of the faults (transfer tracks 1 and 2, air knife and hopper). Therefore, a total of 7 distinct types of operating conditions were recorded as shown in Table 1. Three videos were acquired for normal operation for each ROI, and four videos were acquired for each ROI under faulty operation. For better representation of the machine’s operating conditions, five videos were acquired per condition. As a result, a total 35 (i.e., 7 × 5) videos were acquired. All video files were processed by each fault detection method. Each video file had an average of 600 frames that resulted in around 3000 frames (images) for one operating condition of the machine. Hence, 35 video files were enough for training and testing with the MVI methods.

Three MVI-based methods are developed for detection and classification of the machine’s operating conditions. These methods are as follows: (1) fault detection with GMMs and blob analysis, (2) fault detection with optical flow estimation, and (3) fault detection with foreground running average area. All methods are tested with each video file for their ability to detect and classify the adopted set of faults. However, only one transfer track normal operation and one transfer track jam 1 results are used for the methodology explanation. For all methods, GUI was built in MATLAB as shown in Fig. 9.
Detailed explanation of the methods is given in the following subsections.

5.1 Method 1 fault detection with GMMs and blob analysis

This method uses the foreground detector system object to monitor the machine condition while it is in operation and classifies the operating condition into a normal operation or a faulty operation. To overcome the limitations of a single Gaussian per pixel in the background model, Stauffer and Grimson [30] presented a mixture of adaptive Gaussians per pixel as the background model. The model provides a description of both foreground and background pixels. Foreground represents moving objects (in case of the O-ring machine foreground contains O-rings and carriers), while the background represents stationary objects (such as hopper and machine frames of the O-ring machine) from a video. Hence, the approach of an adaptive Gaussian mixture model, where each pixel in the scene is modeled by a mixture of $K$ Gaussian distributions, works better for moving object detection. The approach compares a video frame to a background model to determine whether individual pixels are part of the background or the foreground. Based on this information, it computes the foreground mask. Once computed, it performs background subtraction and detects foreground objects in an image taken from the overhead stationary camera. The three steps for background detection are as follows: construction of a GMMs, parameter update, and background estimation.

5.1.1 Construction of GMMs

If $I$ is the image sequence and $(x_0, y_0)$ represent the location of a pixel, then the history of the pixel value $X_t$ is obtained by Eq. (2)

$$
\{X_1, \ldots, X_t\} = \{I(x_0, y_0, t) : 1 \leq i \leq t\}
$$

The recent history of each pixel is modeled by a mixture of $K$ Gaussian distributions. In most cases, $K$ is set to be between 3 and 5. The probability of observing a certain pixel value $X_t$ at time $t$ can be written as Eq. (3), where $\omega_{i,t}$ is an estimate of the weight for $i$th Gaussian component, $\mu_{i,t}$ is the mean of the Gaussian component, and $\eta$ is the Gaussian probability density function (Eq. (4)). To simplify computations, the co-variance matrix is assumed to be diagonal and of the form $\Sigma = \sigma_i^2 I$.

This assumes that the red, green, and blue pixel values are independent and have the same variances.

$$
P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \times \eta(X_t, \mu_{i,t}, \sum_{i,t})
$$

Using the history of pixels from a set of frames, each pixel value in the scene is characterized by a mixture of Gaussians. If the history is considered as stationary, the classification of foreground and background clusters can be obtained by maximizing the likelihood of the observed data using EM. The history of pixel in real application is not stationary, and it varies with time $t$. Hence, at each frame at a time $t$, two problems must be solved simultaneously: (1) assignment of new observed values to the best matching distribution and (2)

| Table 1 Summary of video data set |
|-------------------------------|
| ROI          | Condition         | Number of video files |
| Transfer track | Normal operation  | 5                      |
| Transfer track | Transfer 1 jam (fault F1) | 5                      |
| Transfer track | Transfer 2 jam (fault F2) | 5                      |
| Air knife     | Normal operation  | 5                      |
| Air knife     | Air knife fault (fault F3) | 5                      |
| Hopper        | Normal operation  | 5                      |
| Hopper        | Hopper fault (fault F4) | 5                      |
| Total         |                   | 35                     |

![Fig. 8 Detection ratio vs lighting conditions](image-url)
estimating the updated model parameters. The parameters are updated using online K-means approximation.

5.1.2 Parameters update

Each new pixel value is compared with the existing K Gaussian distributions, until a match is found. If a pixel value is within 2.5 standard deviations of a distribution, then it is considered as a match. Once the match is found, the parameters are updated by the following Eqs. (5)–(8).

\[ \omega_k^t = (1-\alpha)\omega_k^{t-1} + \alpha \times M_k^t \]  
\[ \mu_t = (1-\rho)\mu_{t-1} + \rho \times X_t \]  
\[ \sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho (X_t-\mu_t)^T (X_t-\mu_t) \]  
\[ \rho = \alpha \times \eta(X_t|\mu_k,\sigma_k) \]

where \( \alpha \) is the learning rate (range 0 to 1), \( \rho \) is the second learning rate, \( M_k^t \) is 1 for the model that matched and 0 for other models. The \( \mu \) and \( \sigma \) parameters are updated only when the current pixel value matches one of the distributions. In case of no match found, the distribution is replaced with a new distribution using the current value of the pixel as its mean, high initial variance, and low prior weight.

5.1.3 Background estimation

The K distributions are ranked as per their fitness value which is the ratio of the peak amplitude to the standard deviation, \( \omega_k/\sigma_k \). The variance of the moving object is expected to stay larger than the background pixel. Hence, it is assumed that the background has the higher and more compact distributions. As a result of ranking, a new distribution is generated with the most likely distributions on top and the less probable ones on the bottom. The estimated background \( B \) is used to compute the foreground mask:

\[ B = \arg\min_b \left( \sum_{k=1}^b \omega_k > T \right) \]  

The first \( b \) distributions, in ranking order, are used for modeling the background, where \( T \) is a threshold which is the measure of the minimum portion of the data that should be accounted for by the background model. The estimated output from a frame is a set of pixels having higher probability that belong to the foreground. The foreground mask contains pixels that represent both the actual moving objects and the noise due to random motions. Smoothing spatial filters are used on the estimated foreground for noise reduction before the blob analysis.

The next step after foreground detection was blob analysis of the image frame. Blob analysis is a fundamental image processing technique based on the analysis of consistent image regions. It groups image pixels into a set of regions where each region has connected pixels within it. The blob analysis computes statistics for the connected regions in a binary segmented image. It is a way of representing each blob by a number of characteristics such as centroid, area of the blob, bounding box, perimeter, and major and minor axes lengths for ellipses and label. For the O-ring machine application, blob area and bounding box of blobs for
each frame are used as the features. The main steps of method 1 algorithm are given in Fig. 10.

A sample image and plot for normal operation through the transfer track region are shown in Figs. 11 and 12, respectively. The ROI is shown as a rectangle in the frame. The foreground blob area was measured in pixels and plotted w.r.t. the frames as a solid line in the plot. The estimated threshold from the first few training frames was plotted as a dashed line in the plot.

It can be seen in Fig. 12 that the blob area is not a constant w.r.t number of frames; this is due to non-uniform flow of the carriers through the ROI as a consequence of fluctuations in the air pressure and friction. The frame rate of video acquisition was higher than the rate of movement of carriers. Therefore, not all frames had carriers visible in them. Some frames have only background and no carriers in them. Since method 1 measures only foreground area, it does not plot any value for the background. Hence, the blob area for frames without carriers is zero. At some point, the blob area in a particular frame exceeds the threshold, but it is only for a single frame due to a temporary jam in the track. The algorithm is trained not to classify this false jam as a transfer jam.

The fault growth and detection for transfer track 1 jam with method 1 is shown in Table 2.

- Figure A: A carrier reached at the middle of the track in normal operation sequence.
- Figure B: The carrier was ready to engage in the slot of the secondary wheel. The fault was introduced using the HMI-PLC.
- Figure C: The carrier did not engage due to the absence of air pressure as a result of fault introduction. The blob area dropped due to lack of the motion of the carrier since figure B.
- Figure D: Transfer jam fault detected by method 1 and a high value (more than the threshold) of blob area were obtained due to more number of carriers in the track.
- Figure E: Continue running the machine past the fault resulted in the transfer track full of the carriers. The maximum value of blob area was obtained due to the presence of more carriers in the track.

![Fig. 10 Method 1 algorithm steps](image1)

![Fig. 11 Transfer track ROI and normal operation frame](image2)

![Fig. 12 Normal operation blob area feature plot with method 1](image3)
The above events are marked and shown in the fault signature plot. In practice, the method stopped processing the video file as soon as it detected the fault. However, to get an idea of how bad a fault could have gone, the blob area till the end of video was plotted.

### 5.2 Method 2 fault detection with optical flow

Optical flow is a basic motion detection technique in video analysis. Optical flow finds applications in many areas of motion analysis from a video such as: motion-based
segmentation, stereo from motion, video compression, traffic flow monitoring, and people and vehicle tracking. Optical flow was introduced by Horn and Schunck [23] for computing motion between two video frames. Optical flow for a pixel between two video frames is shown in Fig. 13. It is a 2D vector that represents motion in horizontal and vertical directions. For each pixel, it is plotted as a line joining the pixels current position with its previous position. Method 2 uses optical flow for motion detection and fault classification from a video. It calculates the optical flow for a normal operation sequence, which is a measure of the movements of foreground objects. The optical flow density (OFD) is computed from the optical flow of the video. The OFD, the sum of the absolute values of optical flow vectors, is a single number that represents the motion in a frame. During a normal operation sequence, the OFD stays within limits (between zero and the threshold). The OFD of the normal operation sequence is considered as the threshold (reference value). For each frame, the OFD is determined and compared with the threshold. If the current frame OFD exceeds the threshold and keeps on increasing with respect to time, then that is the indication of more number of objects in the frame. The method then keeps track of the OFD for next few frames, and if the trend continues, the machine condition is considered as a fault. A high degree of pixel movement is either due to the motion of a single large object that covers the whole of the region or it can be due to a number of small closely spaced objects. Hence, the OFD is a direct indicator of the degree of motion in a video. Optical flow theory and further background on this approach can be found in MATLAB [31].

The optical flow for each frame of a video is calculated using the above steps by implementing a code in MATLAB. The optical flow is output as horizontal and vertical components in complex forms. The OFD is the absolute value of optical flow as determined for each frame. The OFD is used as a feature to detect and classify faults from videos with method 2. The steps of method 2 algorithm are given as a flowchart in Fig. 14. The first step is to calculate optical flow with the Horn-Schunck method by assuming brightness constancy and smoothness constraint. The maximum number of
iterations to perform in the iterative computation is set to 10. The optical flow is output in the form of horizontal and vertical components in the complex form, where the real part is motion in horizontal direction and the complex part is motion in vertical direction. The OFD is the sum of these absolute values for a given frame. The OFD is directly proportional to the number of moving objects in a frame. The reference value (threshold) of OFD is estimated from the first 200 training frames (one full rotation of the wheels) of a video file. For a normal operation, OFD stays within the limits (0 and threshold).

In the second step, OFD of the current frame is computed and compared with the threshold. If it stays within limits between zero and the threshold, the operation is classified as a normal. The steps are repeated until the end of the video or a fault, whichever occurs first. Figure 15 depicts a full plot of OFD vs number of frames for the normal operation through the transfer track region. It shows that for normal operation, the OFD stays within limits.

If the OFD of the current frame exceeds the threshold, then the method keeps track of the OFD for the next five consecutive frames. If the OFD of the subsequent frames keeps on increasing, then the condition is detected as a transfer jam. The limit of five consecutive frames is selected for fault confirmation because more than five frames added delay in fault detection, while less than five frames misclassified normal operation as a jam.

Table 3 shows important events and fault signature for transfer track 1 jam detection with method 2.

- Figure A: A carrier reached at the end of the track in normal operation sequence.
- Figure B: The carrier was ready to engage in the slot of the secondary wheel. The fault was introduced using the HMI-PLC.
- Figure C: The carrier did not engage due to the absence of air pressure as a result of fault introduction. The OFD reduced due to lack of the motion of the carrier since figure B.
- Figure D: Transfer jam fault detected by method 2 and high density of the OFD were obtained due to more number of carriers in the track.
- Figure E: Continue running the machine past the fault resulted in the transfer track full of the carriers. The OFD reduced due to less movement of the carriers in the track.

The above events are marked and shown in the fault signature plot. In practice, the method stopped processing the video file as soon it detected the fault. However, to get an idea of how bad a fault could have gone, the OFD till the end of video was plotted.

5.3 Method 3 fault detection with running average

A fast and efficient technique for background estimation [25] is implemented for method 3. The method uses preprocessing steps, prior to background estimation, such as spatial Gaussian filter for noise removal and histogram equalization. It reads each image frame and implements the abovementioned preprocessing steps. The machine was operated under a low lighting condition (overhead lights off) to minimize the effect of ambient light. Moreover, to avoid specular reflections from the shiny wheel surfaces, dark field lighting was preferred for video data acquisition. As a result of low light illumination, the video was sensitive to noise. Hence, the first step is the filtering of each frame with a spatial Gaussian filter of size $5 \times 5$ to minimize the effect of noise.

The second preprocessing step is histogram equalization. A histogram of a digital image is a plot of the number of pixels having a particular gray level vs the number of gray levels in the image. Dividing each pixel’s number of occurrences by the total number of pixels, a normalized histogram is obtained. It gives the probability of occurrence for each gray level in the image. There is a direct correlation between image appearance and its histogram. For example, a dark image has its histogram concentrated on left side of the gray level values. Similarly, an image with high contrast and brightness has the histogram that occupies the entire grayscale range, and it has an equal distribution. Therefore, modifying image histogram changes its appearance. For this method, as mentioned previously, the frames are acquired in a low lighting condition. Hence, the frames have histograms that do not cover the entire gray scale span and do not have equal gray level distributions. Histogram equalization is the gray level transformation function that uses gray level probability values and tries to redistribute gray level values such that the resulting image has a histogram that spans the full range and it has nearly equal (not always) probability of occurrences for each gray level. The histogram equalization

![Fig. 15 Normal operation optical flow plot with method 2](image-url)
function is derived from cumulative probability distributions of gray values in the image. This function is applied to each frame of the video file.

The third step of the method is background estimation by image averaging and calculating the running average. In this step, a background frame is obtained by averaging the last 20
The running average method with a filter size of 20 is used to estimate the background from the video. A filter size less than 20 resulted in improper estimate of the background and higher than 20 caused delays in processing. The background at any time $t$ was estimated using Eq. (10). The background estimation is dynamic, in the sense that the estimated background is updated after every 20 frames.

$$B_t = \left( \sum_{i=1}^{20} I_i \right) / 20$$

The method is very fast and has low memory requirements compared to advanced background estimation methods.

The fourth step is foreground, $F$, detection by background subtraction using:

$$F_t = |I_t - B_t|$$

where $I_t$ is the frame at time $t$, $B_t$ is the estimated background, and $F_t$ is the resulting foreground image.

The foreground contains the moving objects, such as carriers and assemblies and noise (unwanted details). The image is then segmented using thresholding to obtain a binary frame. As a result of thresholding, only black and white pixels are available in the image. Moreover, the moving objects may not be full but they can be broken due to uneven lighting. A morphological closing operation with a disk shape structuring element of size 9 is applied to fill holes and smooth contours. The side effect of the closing operation is the results of narrow gaps. This is rectified by a morphological opening operation with a disk shape structuring element of size 3.

The resulting image is considerably noise free and it contains only moving object details. The resulting image has white pixels as assemblies and black pixels as the background. The area of foreground (moving) objects is calculated by counting the number of white pixels in the image. The area value is stored in a vector, and the running average area of the last seven frames is computed. Finally, a threshold (maximum area) is estimated for a normal operation from the first 200 frames. The area trace is a noisy signal, while the running average area trace is much smoother. Running average area is used as a feature to classify the operating condition of the machine. The main steps of the method 3 algorithm are shown in Fig. 16.

As shown in the flow chart, the current running average area is computed and compared with the threshold. If it stays within limits between zero and the threshold, the operation is classified as a normal. The steps are repeated till the end of a video file or a fault, whichever occurs first. Figure 17 depicts the plot of running average area vs the number of frames for the normal operation through the transfer track region. It shows that for normal operation, the running average area stays within the limits (0 and threshold).

If the area of the current frame exceeds the threshold, then the algorithm keeps track of the area for the next seven consecutive frames. If the area of the subsequent frames keeps on increasing, then the condition is detected as a transfer jam.

The limit of seven consecutive frames is selected for fault confirmation because more than seven frames added delay in fault detection while less than seven frames misclassified normal operation as a jam. The jam is then classified as either transfer 1 or transfer 2 jam by locating the coordinates of the white pixels in the ROI. A GUI, similar to previous methods, was designed in MATLAB for method 3. The machine operating conditions are indicated by five indicators, one for a normal operation and four for the faults.
Table 4 shows important events and fault signature for transfer 1 jam detection with method 3.

- Figure A: A carrier reached at the middle of the track in normal operation sequence.
- Figure B: The carrier was ready to engage in the slot of the secondary wheel. The fault was introduced using the HMI-PLC.
- Figure C: The carrier did not engage due to the absence of air pressure as a result of fault introduction. The running avg. area dropped due to lack of the motion of the carrier since figure B.
- Figure D: Transfer jam fault detected by method 3 and a high value (more than the threshold) of running avg. area were obtained due to more number of carriers in the track.
- Figure E: Continue running the machine past the fault resulted in the transfer track full of the carriers. The maximum value of running avg. area was obtained due to the presence of more carriers in the track.

The above events are marked and shown in the fault signature plot. In practice, the method stopped processing the video file as soon as it detected the fault. However, to get an idea of how bad a fault could have gone, the running avg. area till the end of video was plotted.

6 Performance measurements

Fault detection and classification with three MVI methods were tested for seven operating conditions of the machine out of which three conditions were normal and four were faulty. All conditions were grouped into three ROIs: transfer track ROI, air knife ROI, and hopper ROI. The ROIs were processed one at a time, and the parameters of the three MVI methods were tuned for individual regions at first. Setting the parameters only for a specific ROI resulted in the perfect classification for that ROI. In the first stage of the algorithm development, tuning parameters for each method were adjusted to correctly classify only one fault condition at a time. Once, the method worked for one fault, the parameters were modified and the method was tuned to detect and classify the remaining faults with the same parameters. At this stage, the methods got separated in terms of their ability to classify the faults. As a result of the fixed parameters, the accuracy of classification dropped for method 2 significantly and for method 3 marginally.

In order to better compare the performance of the three methods, a novel machine vision performance index (MVPI) was developed. The MVPI is based on five measures of performance: accuracy of a method to classify the given machine condition, number of frames processed in a given time, speed of response, robustness against noise, and ease of implementation or tuning efforts. These five parameters are selected because a MVI system needs time to process a video file and display frames; it requires a certain number of frames to see a fault, it can be sensitive to noise, and it can be application specific. This means that when a MVI system developed for one application is applied to a different application, it needs to be retuned. The MVPI is the weighted sum of these parameters. It is hard to develop a single MVI method that could be the best in all five measures. For example, a MVI method involving complex steps can give the highest accuracy but it can be slow in terms of the processing speed of the algorithm. Hence, the MVPI serves as a more comprehensive way of comparing the performance of different MVI methods. The next few paragraphs explains the performance parameters in more detail. These parameters are determined for all three methods and are then normalized and multiplied by weights to obtain the MVPI. Accuracy is assigned a higher weight compared to other parameters because it is of prime consideration.

6.1 Parameter $p_1$: measure of accuracy in fault detection

Accuracy in fault detection and classification is the measure of the number of correct classifications. The accuracy is calculated based on the inspection results of 35 video files.

6.2 Parameter $p_2$: number of frames processed in 10 s

This parameter depends on the time taken by a method to process a single frame of the video file. A method with less time to process a single frame can process more number of frames in 10 s, and it can handle high frame rates for video file acquisition. For data acquisition, a frame rate of 30 fps was used but a higher fps may be required if the machine is required to operate at a higher speed. High fps provides better visualization in the case of a fault diagnosis because it can...
replay the motion with finer movement of the parts with respect to time. Hence, this parameter is an important consideration when selecting the fps of the camera relative to the machine speed.

6.3 Parameter $p_3$: measure of speed of response

This parameter is measured indirectly by measuring how fast a method detects the fault. Each MVI method needs a certain

**Table 4** Fault events and signature for transfer track 1 jam detection with method 3

| A: Normal frame (200th frame) | B: Fault introduction frame (300th frame) |
|-------------------------------|------------------------------------------|
| Carrier in the middle of the track (normal operation) | Carrier ready to engage in the slot |

| C: Fault visible frame (325th frame) | D: Fault detected frame (357th frame) |
|-------------------------------------|-------------------------------------|
| Carrier did not engage (running avg. area dropped as no motion since B) | Jam detected, running avg. area exceed the threshold due to more carriers |

| E: Fault's growth if undetected (430th frame) |
|---------------------------------------------|
| Transfer track full of carriers (running avg. area maximum) |

Running avg. area plot for transfer track 1 jam
number of frames to see a fault after it was introduced. The number of frames required to detect the fault varied from a fault to fault and therefore an average value of the frames was used in the calculation. An MVI method that detects a fault sooner is in a better position to take necessary corrective action to prevent further damage. The parameter is scaled by dividing the number of frames required by each method with the frame rate of video file acquisition (i.e., 30 fps) and subtracting the time from 2 s. The time of 2 s was selected as the maximum available time to communicate the decision before serious damage could occur on the machine. The maximum time available was measured in practice by continue running the machine even after the fault. The sooner the MVI system communicates the decision, the lower will be the damage to the machine. Parameter $p_3$ is the time available to communicate the decision.

6.4 Parameter $p_4$: measure of robustness against noise

Noise can affect the quality of images and consequently the response of a MVI system. Noise in images may result due to poor lighting, noise in the image sensor itself, or noise in the transmission and reception of the image data. To study the effect of noise on the MVI methods, all video files were corrupted with Gaussian noise of 0 (zero) mean and 0.01 variance. The unit of noise is pixel intensity. This is the source of noise in the images. All noisy video files were processed by the three MVI methods and accuracy results were recorded. The parameter $p_4$ is the ratio of accuracy with noisy video dataset to accuracy without noisy video dataset.

6.5 Parameter $p_5$: measure of ease of implementation and tuning

MVI solutions are application specific and their tuned parameters depends on the type of camera, lens, and lighting. Some parameters in an algorithm are fixed, while others are tunable. Hence, changing either the camera, lens, and lighting for the same application, or changing to different application, requires retuning of the parameters. Method 1 has five tunable parameters. Method 2 has four tunable parameters. Method 3 has three tunable parameters. The parameters are shown in Table 5. The ease of implementation and tuning is inversely proportional to the number of tunable parameters. The parameter $p_5$ is calculated by multiplying the inverse of the number of tunable parameters with a scaling factor for each MVI method. The MVPI is the weighted sum of these five performance parameters and is calculated using Eq. (12).

$$W_a \times p_1 + W_b \times p_2 + W_b \times p_3 + W_b \times p_4 + W_b \times p_5 \times 100 \over (W_a + 4 \times W_b)$$

(12)

where $W_s (W_a = 100$ and $W_b = 50$) are the weights which determined the relative importance of the performance measure. $W_a$ is double that of $W_b$, as accuracy is more important than the other parameters. The weight of 2 was selected for illustrative purposes. As long as $W_a$ was greater than $W_b$, it was found that the relative ranking of the three methods did not change.

7 Results and discussion

The inspection results of individual performance parameters are given below:

Accuracy ($p_1$) Method 1 classified 35 out of 35 video files correctly. Method 2 classified 21 out of 35 video files correctly. Method 3 classified 33 out of 35 video files correctly. Hence, accuracy of method 1 is 100 %. Method 2 provided 21 correct classifications out of 35. Therefore, its accuracy is 60 %. Method 2 misclassified 14 video files with 6 false positives (FP; operation is faulty and MVI method classifies it as normal) and 8 false negatives (FN; operation is normal and MVI method classifies it as faulty). Method 3 classified 33 out of 35 videos correctly with 1 FP and 1 FN result. Hence, the accuracy for method 3 is 94 %. The accuracy results are illustrated in Fig. 18.

Number of frames processed in 10 s ($p_2$) Methods 1, 2, and 3 processed 900, 783, and 531 frames, respectively. The number of frames results are illustrated in Fig. 19. Parameter $p_2$ is obtained by dividing these numbers by 1000 frames.

| Sn. | Method 1 | Method 2 | Method 3 |
|-----|----------|----------|----------|
| 1   | No. of frames used for background estimation | Optical flow components type | Running average filter size for background estimation |
| 2   | Initial variance for GMMs | Subsampling size for optical flow field | Filter size for running average area calculation |
| 3   | Minimum blob area to be detected | Scaling of flow fields | No. of frames used to estimate a threshold |
| 4   | Maximum blob area | No. of frames used to estimate a threshold | |
| 5   | No. of frames used to estimate a threshold | | |
Measure of speed of response ($p_3$) This is the time available to communicate the decision. Method 2 proved to be the best with 0.84 s available compared to method 1 (0.80 s) and method 3 (0.57 s). The results are shown in Fig. 20.

Measure of robustness against noise ($p_4$) This is the ratio of accuracy with noisy video dataset to accuracy without noisy video dataset. It was observed that the accuracy for noisy video files with method 1 was 54% (19 correct classifications out of 35), while with methods 2 and 3 were 14% (5 correct classifications out of 35). Hence, methods 1, 2, and 3 have the values of parameter $p_3$ as 0.54, 0.24, and 0.14, respectively, as shown in Fig. 21.

Measure of ease of implementation and tuning ($p_5$) Method 1 has five tunable parameters. Method 2 has four tunable parameters. Method 3 has three tunable parameters. The parameter $p_5$ is calculated by multiplying the inverse of the number of tunable parameters with the scale of 2 for each MVI method. Hence, for method 1, it is 0.4. For method 2, it is 0.5. For method 3, it is 0.7. Figure 22 illustrates these results.

Using Eq. (12) and the values from Table 6, the MVPI is calculated for each method. The MVPI for method 1 is 77, method 2 is 58, and method 3 is 64. These results are illustrated in Fig. 23. Method 1 achieved the highest MVPI. Method 2 had the lowest MVPI. The MVPI gives the relative ranking of the methods based on the five assigned performance measures.

8 Conclusions

MVI system with a single camera was designed to detect and classify a variety of faults on an automated assembly machine using CV techniques. Three MVI methods based on GMMs, optical flow, and foreground running average area were developed. The methods are adaptive in the sense that they learn the normal behavior of the machine from a set of initial frames in the video. They do not require separate training and testing datasets. The inherent value of a camera-based system
compared to traditional sensor-based systems was explained. For example, during a video playback for a transfer track jam fault, it was discovered that one of the reasons for the jam was a loose O-ring that obstructed the flow of assemblies through the narrow path. Another reason of transfer track jam was primary and secondary wheel binding. The MVPI for performance evaluation and relative performance comparison among multiple MVI methods were developed and applied to the task of fault detection and classification. The MVPI was calculated by the weighted average of the performance parameters. The conclusions of the evaluation using the MVPI are as follows: (a) single method was not the best in all performance criteria, (b) the accuracy was significantly reduced for all MVI methods with noisy input data, and (c) the GMM-based method had the highest overall performance result.

Development of the MVI system using a USB 3.0 camera and four LED lights is not only beneficial for the O-ring assembly machine inspection but it can also be used for other applications such as rotary bowl feeder inspection, jam detection on transfer lines, and missing component detection in assemblies. The GUI-based MVI algorithms and MVPI can be used by other researchers in the field of machine vision to evaluate their own MVI applications. However, the MVI algorithms will need to be tuned for these applications. The tuning will include adjusting the number of training frames, the initial variance for GMMs, the density of optical flow, threshold estimation, and the running average filter size. In addition to algorithm tuning, hardware tuning such as setting the camera focus, WD, aperture, and light intensity will also be necessary for any new application.

The developed MVI methods have the following limitations:

- The methods are able to detect and classify only faults with visual characteristics.
- The methods need faults to build up for certain frames after their introduction.
- The methods process only one ROI at a time. As a result, they are unable to monitor and detect simultaneous occurrences of multiple faults.
- The methods learn normal operation sequence from a set of initial frame. Hence, the machine should run fault free during initial frames of video acquisition.
- The accuracy of inspection has dropped significantly with noisy data.
- MVI-based methods are sensitive to variations in lighting. They require consistency in light during training and testing phases.
- If the system is to be used to monitor a machine that runs continuously for a long period of time, the threshold values should be updated at a regular interval of time.

Future work can include expansion of the system such that it can see simultaneously look for multiple faults on the machine. The MVI methods were able to detect and classify only known faults because the parameters were tuned for those faults. An unknown fault will have a different characteristic, and it will require retuning the parameters of the MVI methods. Therefore, the limitation of the existing methods was their inability to handle unknown faults. Future work should look at what is required for the system to handle unknown faults. Deep neural networks (DNNs) could be used to detect unknown faults with their complex structure composed

| Table 6 | Performance parameters and MVPI |
|---------|--------------------------------|
| Parameter | Method 1 | Method 2 | Method 3 |
| $p_1$    | 1.00     | 0.60     | 0.94     |
| $p_2$    | 0.90     | 0.78     | 0.53     |
| $p_3$    | 0.75     | 0.84     | 0.57     |
| $p_4$    | 0.54     | 0.24     | 0.14     |
| $p_5$    | 0.40     | 0.50     | 0.70     |
| MVPI     | 77       | 59       | 64       |
of multiple processing layers and multiple non-linear transformations [32].

References

1. Holloway L, Krogh B (1990) Fault detection and diagnosis in manufacturing systems: a behavioral model approach. Proc. of Rensselaer’s Second International Conference on Computer Integrated Manufacturing 252–259
2. Boothroyd G (2005) Assembly automation and product design, 2nd edn. Florence: Marcel Dekker, New York
3. Viswanadham N, Johnson T (1988) Fault detection and diagnosis of automated manufacturing systems. Proc of the 27th IEEE Conference on Decision and Control 3:2301–2306
4. Konrad H (1996) Fault detection in milling, using parameter estimation and classification methods. Control Eng Pract 4(11):1573–1578. doi:10.1016/0967-0661(96)00172-4
5. Wang W, McFadden P (1996) Application of wavelets to gearbox vibration signals for fault detection. J Sound Vib 192(5):927–939
6. Dalpiaz G, Rivola A (1997) Condition monitoring and diagnostics in automatic machines: comparison of vibration analysis techniques. Mech Syst Signal Process 11(1):53–73
7. Patel S, Kamrani A, Orady E (1995) A knowledge-based system for fault diagnosis and maintenance of advanced automated systems. Comput Ind Eng 29(1–4):147–151
8. Demetgül M, Tansel I, Taskin S (2009) Fault diagnosis of pneumatic systems with artificial neural network algorithms. Expert Syst Appl 36(7):10512–10519
9. Demetgül M, Unal M, Tansel I, Yazıcıoğlu O (2011a) Fault diagnosis on bottle filling plant using genetic-based neural network. Adv Eng Softw 42(12):1051–1058
10. Fernando H, Surgenor B (2015) An unsupervised artificial neural network versus a rule-based approach for fault detection and identification in an automated assembly machine. Robotics Comp Integrated Manuf. doi:10.1016/j.rcim.2015.11.006
11. Sekar R, Hsieh S, Wu Z (2011) Remote diagnosis design for a plc-based automated system: 1-implementation of three levels of architectures. Int J Adv Manuf Technol 57(5–8):683–700
12. Sekar R, Hsieh S, Wu Z (2013) Remote diagnosis design for a plc-based automated system: 2-evaluation of factors affecting remote diagnosis performance. Int J Adv Manuf Technol 65(5–8):1091–1109
13. Malamas E, Petakis E, Zervakis M, Petit L, Legat J (2003) A survey on industrial vision systems, applications and tools. Image Vis Comput 21(2):171–188
14. Yao A (2005) Design and implementation of web-based diagnosis and management system for an FMS. Int J Adv Manuf Technol 26(11–12):1379–1387
15. Killing J, Surgenor B, Mechefske C (2009) A machine vision system for the detection of missing fasteners on steel stampings. Int J Adv Manuf Technol 41(7–8):808–819
16. Ke Y, Sukthankar R, Hebert M (2007) Event detection in crowded videos. IEEE 11th International Conference on Computer Vision (ICCV) 1–8
17. Benezeth Y, Jodoin P, Saligrama V, Rosenberger C (2009) Abnormal events detection based on spatio-temporal co-occurrences. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2458–2465
18. Hughes K, Fernando H, Szklínyk G, Surgenor B, Greenspan M (2014) Video event detection for fault monitoring in assembly automation. Int J Intell Syst Technol Appl 32(11):1201–1210
19. Shahabi H, Ratnam M (2009) In-cycle monitoring of tool nose wear and surface roughness of turned parts using machine vision. Int J Adv Manuf Technol 40(11):1148–1157
20. Xiaokun L, Porikli F (2004) A hidden Markov model framework for traffic event detection using video features. Proc of the Int Conference Image Process (ICIP) 5:2901–2904
21. Zezhi C, Pears N, Freeman M, Austin J (2014) A Gaussian Mixturemodel and support vector machine approach to vehicle type and colour classification. Intelligent Trans Syst (IET) 8(2):135–144
22. Horn B, Schunck B (1981) Determining optical flow. Proc. International Society for Optics and Photonics, Technical Symposium East 319–331
23. Patel E, Shakla D (2013) Comparison of optical flow algorithms for speed determination of moving objects. Int J Comp Appl 63(5):32–37
24. Usamentiaga R, Molloja J, Garcia D, Bulnes F, Perez J (2013) Jam detector for steel pickling lines using machine vision. IEEE Trans Ind Appl 49(5):1954–1961
25. 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
26. Miles BC, Surgenor BW, Killing J (2008) Effect of lighting on the performance of a machine vision system. Proc. of 18th Conference on Flexible Automation and Intelligent Inspection (FAIM), Skövde, Sweden
27. Kopparapu SK (2006) Lighting design for machine vision application. Image Vis Comput 24(7):720–726
28. Shahabi H, Fernando H, Surgenor B (2014) Effect of illumination techniques on machine vision inspection for automated assembly machines. Proc. of The Canadian Society for Mechanical Engineering (CSME) International Congress, Toronto 1–6
29. Stauffer C, Grimson WE (1999) Adaptive background mixture models for real-time tracking. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2:1–7
30. MATLAB (2014) Image acquisition, processing and computer vision toolboxes. The Mathworks, Inc., U.S.
31. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Proc. of 25th Annual Conference on Advances in Neural Information Processing Systems (NIPS) 1097–1105