Automating Manufacturing Surveillance Processes Using External Observers

by

Gauri Sharma

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

An automated assembly system is an integral part of various manufacturing industries as it reduces production cycle-time resulting in lower costs and a higher rate of production. The modular system design integrates main assembly workstations and parts-feeding machines to build a fully assembled product or sub-assembly of a larger product. Machine operation failure within the subsystems and errors in parts loading lead to slower production and gradual accumulation of parts. Repeated human intervention is required to manually clear jams at varying locations of the subsystems. To ensure increased operator safety and reduction in cycle-time, visual surveillance plays a critical role in providing real-time alerts of spatiotemporal parts irregularities.

In this study, surveillance videos are obtained using external observers to conduct spatiotemporal object segmentation within: digital assembly, linear conveyance system, and vibratory bowl parts-feeder machine. As the datasets have different anomaly specifications and visual characteristics, we follow a bottom-up architecture for motion-based and appearance-based segmentation using computer vision techniques and deep-learning models.

To perform motion-based segmentation, we evaluate deep learning-based and classical techniques to compute optical flow for real-time moving-object detection. As local and global methods assume brightness constancy and flow smoothness, results showed fewer detections in presence of illumination variance and occlusion. Therefore, we utilize RAFT for optical flow and apply its iteratively updated flow field to create a pixel-based object tracker. The tracker differentiates previous and current moving parts in different colored segments and simultaneously visualizes the flow field to illustrate movement direction and magnitude. We compare the segmentation performance of the optical flow-based tracker with a space-time graph neural network (ST-GNN), and it shows increased accuracy in boundary mask IoU alignment than the pixel-based tracker. As the ST-GNN addresses the limited dataset challenge in our application by learning visual correspondence as a contrastive random walk in palindrome sequences, we proceed with ST-GNN to perform motion-based segmentation.

As ST-GNN requires a first-frame annotation mask for initialization, we explore appearance-based segmentation methods to enable automatic ST-GNN initialization. We evaluate pixel-based, interactive-based, and supervised segmentation techniques on the bowl-feeder image dataset. Results illustrate that K-means applied with watershed segmentation and gaussian blur reduces superpixel oversegmentation and generates segmentation aligned with parts boundary. Using Watershed Segmentation on bowl-feeder image dataset, 377

parts were detected and segmented of total 476 parts present within the machine. We find that GLCM and Gabor filter perform better in segmenting dense parts regions than graph-based and entropy-based segmentation. In comparison to entropy-based and graph-based methods, the GLCM and Gabor filter segment 467 and 476 parts, respectively, of total 476 parts present within the bowl-feeder. Although manual annotation decreases efficiency, we see that the GrabCut annotation tool generates segmentation masks with increased accuracy than the pre-trained interactive tool. Using the GrabCut annotation tool, all 216 parts present within the bowl-feeder machine are segmented. To ensure segmentation of all parts within the bowl-feeder, we train Detectron2 with data augmentation. We see that supervised segmentation outperforms pixel-based and interactive-based segmentation.

To address illumination variance within datasets, we apply color-based segmentation by conversion of image datasets to HSV color space. We utilize the images, converted within the value channel of HSV representation, for background subtraction techniques to detect moving bowl-feeder parts in real-time. To resolve image registration errors due to lower image resolution, we create Flex-Sim synthetic dataset with various anomaly instances consisting of multiple camera viewpoints. We apply preprocessing methods and affine-based transformation with RANSAC for robust image registration. We compare color and texture-based handcrafted features of registered images to ensure complete image alignment. We evaluate the PatchCore Anomaly detection method, pre-trained on MVTec industrial dataset, to the Flex-Sim dataset. We find that generated segmentation maps detect various anomaly instances within the Flex-Sim dataset.

Acknowledgements

I would like to sincerely thank my research supervisor, Dr.Zelek, for his encouraging and patient guidance throughout the research project. I am extremely grateful to have an inspiring mentor who cultivated a strong curiosity-driven environment and provided thorough feedback.

I would like to sincerely thank Professor Zhao Pan and Professor Javad Shafiee for taking the time to read and provide feedback on my thesis. I greatly appreciate your time and feedback.

I would like to thank VIP Lab Students and Postdoctoral fellows: Dr. Zobeir Raisi, Dr. Georges Younes, and Dr. Mohamed Naiel for their positive and supportive feedback. I would like to thank Stan Kleinikkink and ATS Automation for their time and support towards this research project.

Dedication

To my parents, my grandmother Sarlanj, and Shahad for their love and sacrifices; and without whom this would not be possible.

Table of Contents

List of Figures

Chapter 1

Introduction

With advancements in digital manufacturing and smart factories, mass customization is becoming more and more prevalent; and the integration of reconfigurable automated systems has been increasing to optimize an end-to-end manufacturing process. Assembly is an integral part of the product lifecycle. Data illustrates that the time allocated towards product assembly accounts for 20%-50% of the total production time, and the manufacturing costs associate with 20%-30% of the total cost of a fully assembled product [99]. Therefore, product assembly has a significant impact on factors such as product delivery time, cost, quality, durability, as well as maintenance. By digitalization of the assembly line, manufacturing companies aim to optimize such assembly factors while enabling mass customization.

Across the different industries in manufacturing, the assembly of custom industrial products involves complex manufacturing processes to assemble base components of varying shapes and functions. With the rise of COVID-19, the global manufacturing industry experienced increased product demand throughout many sectors. According to the United Nations Industrial Development Organization (UNIDO), global manufacturing production increased by 9.4% in 2021. The market size within manufacturing, measured by revenues in USD, was estimated to be $434.2 billion in global healthcare sector, $952.4 billion in global aerospace sector, $2.7 trillion in global automotive sector, and $724.48 billion in global consumer electronics sector [85]. To meet the increasing demands for varied industrial products, manufacturing companies are in critical need to employ assembly systems, which enable mass production with increased operational efficiency, reduced manufacturing defects, and lower costs.

Through the emergence of the fourth industrial revolution, the application of digital

technologies such as 3D printing, artificial intelligence, robotics, and the internet of things (IoT) makes it possible to use reconfigurable automated systems. Specifically, in applying these systems to perform complex assembly operations using real-time production monitoring, decentralized 3D printing facilities, real-time optimization and decision-making support. Such system attributes enable product assembly with decreased defects, costs and increased efficiency, respectively. To increase operational efficiency in mass production, the worldwide spending on digital transformation in manufacturing is forecast to reach $2.8 trillion USD by 2025[85].

With the integration of reconfigurable automated systems within the production process, manufacturing companies aim to complete the core work processes such as materials handling, milling, assembly, and inspection, as illustrated by the production process in Figure 1.1. Such examples of digital transformation are evident within the Aerospace and Healthcare manufacturing industries. Specifically, the initiative taken by Relativity Space, an aerospace manufacturing company, to enable cost-effective and modular assembly of rockets using 3D printing technology [59]. Thus reducing the time in orbital rocket launch to days as opposed to years, which previously resulted due to conventional manufacturing processes applied. In addition to the intelligence and analytics integrated within Siemens manufacturing sector, which aims to provide real-time machine health monitoring facility[23]. This is significant as the use of digital technologies in manufacturing enforces reliability with minimal parts required, speed through a faster production time, flexibility in the supply chain, and optimization.

In this thesis, the research methods are evaluated on surveillance videos of automated assembly systems, which are used to perform core assembly processes within the consumer goods manufacturing sector. Specifically, the assembly machines and part-feeding system: linear conveyor system, digital assembly and the vibratory bowl-feeder machine.

1.1 Problem Scope

An automated assembly system integrates multiple electromechanical automated devices, which perform a sequence of assembly operations to combine multiple components into a fully assembled product or a subassembly of a larger product. To reduce production cycle-time resulting in lower costs and higher rates of production, the devices are embedded as part of the following subsystems: the main assembly workstation and parts-feeding at the workstation. The workstations are the main sites within the assembly system, which perform the core manufacturing operations for the assembly of base components. In order to move parts from the initial position at specified angle intervals, the assembly

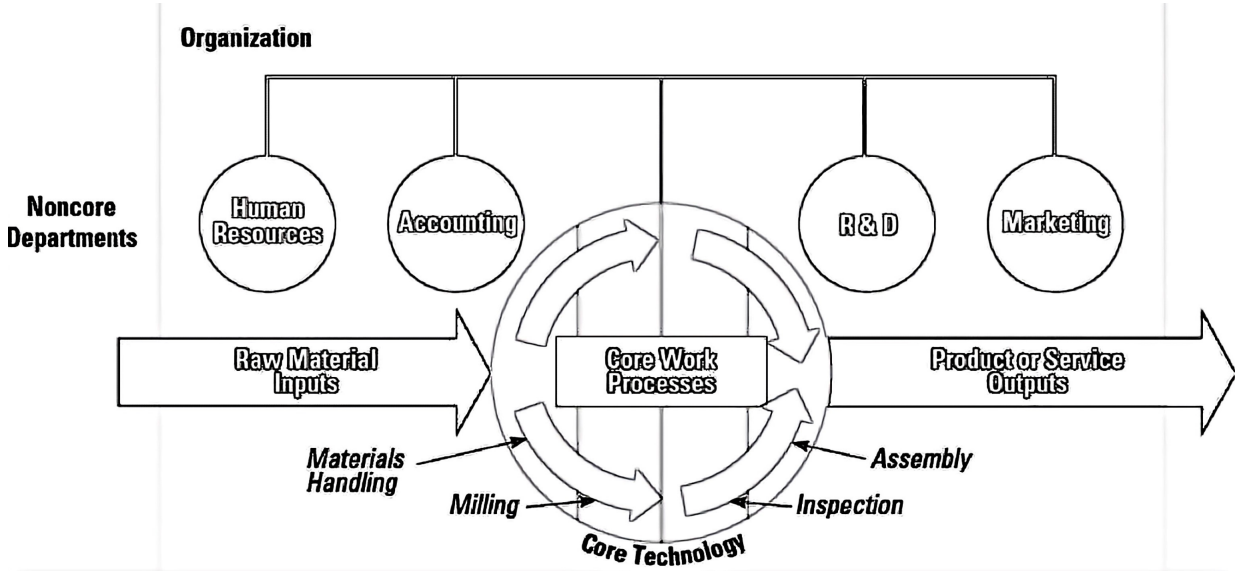


Figure 1.1: Core manufacturing operations conducted during the assembly process[25]

workstations consist of linear and rotatory indexing systems[69]. As the indexers are driven pneumatically or through servo-based systems, they play a critical role in controlling the part acceleration and decelerations. This is significant towards ensuring accurate part positioning, and facilitating smooth part transfer between two intermediate workstations. To reduce time in parts transfer between main assembly workstations, linear and rotational indexers within workstations operate in continuous, synchronous, and asynchronous motion for the transfer of parts[69]. Therefore, all moving parts regulate with shared motion characteristics within the assembly machine.

The varied motion of the transfer system is obtained through indexing applied in different configurations of the main workstations. Specifically, in configurations such as the dial-type, in-line, carousel, and single-station assembly machines. Depending on the integrated indexing system, the modular workstations perform manufacturing operations such as screw driving and dispensing, pick and place, crimping, ultrasonic welding, and pressing[69].

Within the dial-type configuration, the base partare is indexed around a circular dial, as illustrated in Figure 1.2. The rotatory configuration operates in a synchronous and continuous motion to index parts, which are positioned on the outer periphery of the

dial. In manufacturing operations, the dial-type configuration is applied to add and fasten components at workstations surrounding the outer periphery of the dial. The in-line type configuration consists of workstations arranged in a linear sequence, as illustrated in Figure 1.2, to perform assembly operations such as metal-cutting. The in-line configuration enables parts transfer using continuous, synchronous, and asynchronous motion. The carousel configuration integrates the circular configuration of dial-type and linear configuration of in-line to transfer parts using continuous, synchronous, and asynchronous transfer. The single-station configuration consists of a stationary base part system, in which robotic manipulators are used to deliver base parts and transfer completed assemblies in linear sequence. Through the different physical configurations, the linear and rotational indexers operate in synchronous and asynchronous motion for machine operation and part transfer. In normal operation, the parts are transferred between intermediate workstations with specified orientation and movement[69]. To facilitate parts-feeding at workstations, the

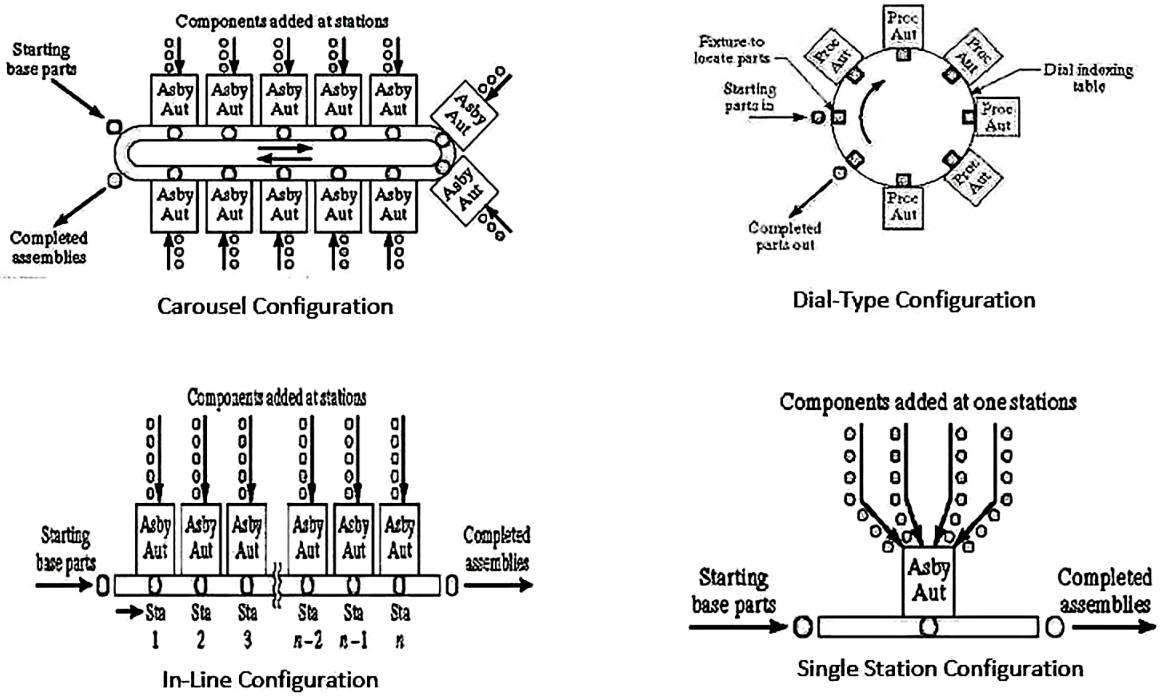


Figure 1.2: Illustration of various assembly machine configurations[14]

parts-feeding systems such as a hopper and a vibratory bowl-feeder are used for single part feed into the assembly workstations. The hopper serves as the container in which parts are loaded in bulk. During the initial parts-feeding within the hopper machine, the parts orientation is randomly arranged. The vibratory bowl-feeder consists of different shelf levels with variations in the accumulation of parts,as illustrated in Figure 1.3. The design of the bowl-feeder shelves enables decreased accumulation of parts towards the bowl-feeder exit and controls the orientation of parts within the outer shelf. Additionally, the design of the parts-housing located within the electromechanical devices within the main assembly workstations and the parts-feeding system enable the accurate placement of parts, with different shapes and sizes, into a specified orientation[69].

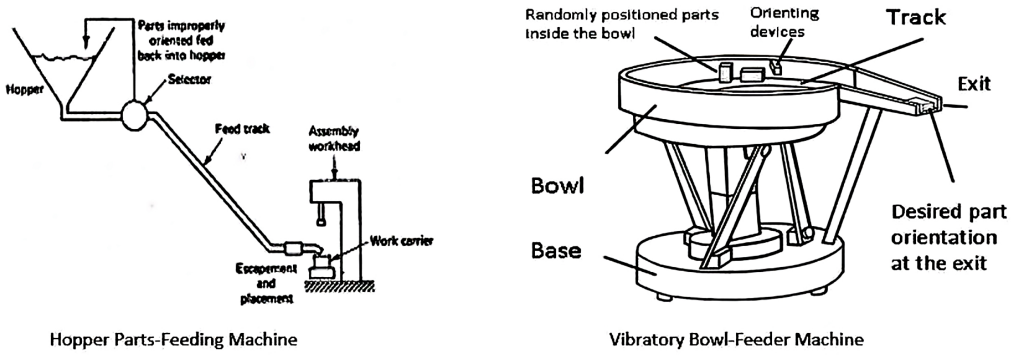


Figure 1.3: Illustration of parts-feeding machines within automated assembly systems[14]

1.2 Problem Motivation

During the normal operation of main assembly workstations and parts-feeding system, machine operation failure within the subsystems and errors in parts loading lead to increased cycle-time. Specifically, factors such as mechanical failure and increased sensor noise in motion control systems affect algorithmic objective resulting in minimization of fault-tolerance and an increase in the total time allocated in assembly[69]. Additionally, slower production rate is caused by variations in parts loading such as missing parts, new part-type insertion of varying shapes, part orientation change, and misaligned parts fit-

ting within the housing. An example of exact part alignment within indexer housing is illustrated in Figure 1.4.

Therefore, the decrease in production rate over time within workstations leads to the accumulation of parts thus causing part jams within various locations of the subsystems. Repeated human intervention is required to manually clear jams within different subsystems' locations. This is significant as the accumulation of parts over time results in lower production throughput and decreased machine operator safety. To ensure a reduction in cy-

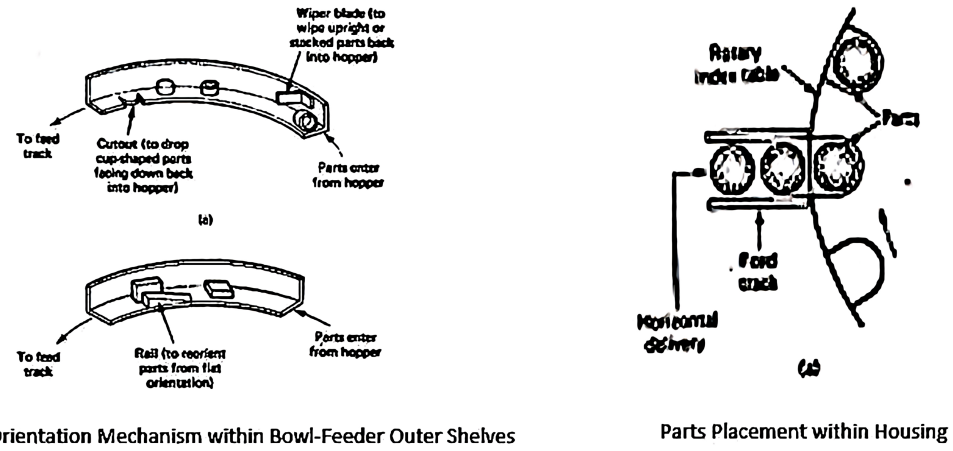


Figure 1.4: Illustration of part placement within indexer housing and orientation devices within bowl-feeder machine[14]

cle time and increased operator safety, automated visual inspection systems play a critical role towards recognition, monitoring, and providing real-time alerts of the spatiotemporal irregularities within the workstations and parts-feeding systems. For this reason, a time series analysis is performed to extract spatial and temporal characteristics of moving parts in real-time. The extracted characteristics are analyzed for comparison with normal behavior part characteristics within main assembly workstations and parts-feeding systems.

The visual surveillance can be integrated as part of the in-line production process, or an externally placed monitoring device located outside of main workstations within the automated assembly systems, as illustrated in Figure 1.5[44]. Therefore, the surveillance videos

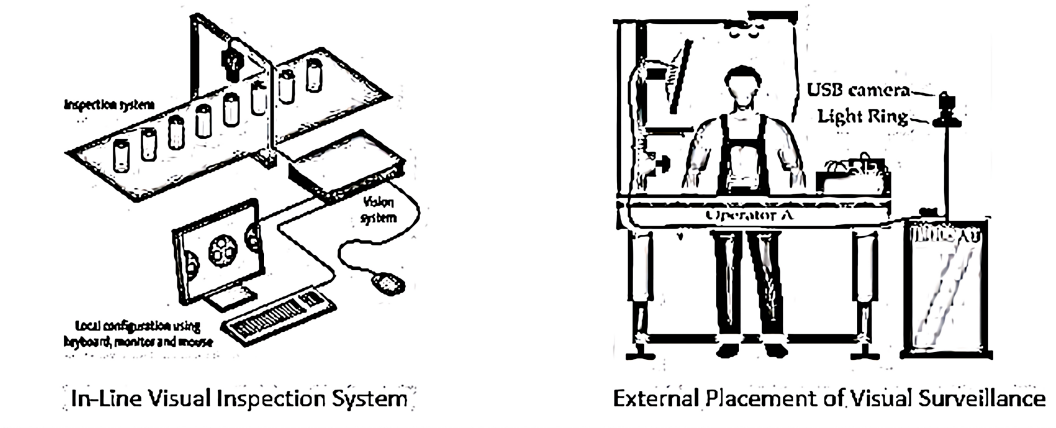


Figure 1.5: Illustration of in-line and externally placed visual inspection system[1]

can consist of parameters such as varying anomaly specifications and multiple regions of interest captured from various camera viewpoints. The placement of an automated visual inspection system outside the location of main assembly workstations and parts-feeding systems captures external disturbances such as repeated machine operator intervention. This is significant as such external disturbances introduce limitations in the visual characteristics such as occlusion, illumination variation, and lower image resolution.

Current works within the automated inspection systems focus on visual surveillance as part of the in-line manufacturing process. The objective of existing in-line visual surveillance is to detect surface defects within manufacturing parts of varying textures. Within the pre-existing manufacturing datasets, the region of interest consists of a zoomed-in focus on the different part classes within the manufacturing assemblies such as tablets, metal bolts, carpets, etc[11]. Therefore, minimal external disturbances exist within the dataset such as machine operator intervention and limitations within the visual characteristics.

1.2.1 Problem Statement

In this study, surveillance videos are obtained of the digital system with modular assembly blocks, linear motor-based conveyance, and vibratory bowl-feeder parts-feeding machine. The research objective is to detect part jams by providing real-time alerts of spatiotemporal irregularities within varying regions of the automated assembly machines. By performing

a time series analysis of spatiotemporal part characteristics, the parts accumulation can be detected early on before a part jam occurs, and thereby decreasing production cycle-time.

1.3 Thesis Contribution

To conduct time-series analysis of spatiotemporal characterisics for anomaly detection, spatiotemporal part segmentation is performed on the obtained surveillance videos. Specifically, a bottom-up architecture is followed to perform motion-based and appearance-based segmentation by evaluating computer vision techniques and deep learning-based models. Image preprocessing methods are applied to address limited visual characteristics within the datasets. Additionally, spatiotemporal characteristics are analyzed for anomaly detection using traditional methods and deep learning-based models. Based on the methods evaluation on various manufacturing datasets, we show:

- Segmentation of current and previous moved parts using a pixel-based tracker. Based on method evaluation, the tracker generated different colored segments to visualize current and previously moved parts. Within the bowl-feeder dataset, the tracker differentiated the outer-shelves foreground parts with frequent motion and accumulated parts within base-shelf in different colored segments.

- Automatic initialization of Space-time Graph Neural Network(ST-GNN) model using appearance-based segmentation to generate a first-frame annotation mask. Based on evaluation of appearance-based segmentation methods, supervised segmentation outperformed pixel-based and interactive-based segmentation techniques. With supervised model training and evaluation applied on the bowl-feeder dataset, the Detectron2 model generated segmentation mask on foreground parts with increased boundary mask IoU alignment .

- Robust affine image registration applied with correspondence selection to transform synthetic manufacturing images, captured from multiple camera viewpoints, into one coordinate system. Registered images increased accuracy in spatial feature extraction and anomaly detection.

Chapter 2

Background

2.1 Obtained Surveillance Videos

An automated assembly system performs a sequence of assembly operations to build a full assembled product. To reduce production cycle-time,the devices are embedded as part of the following subsystems: main assembly workstation and parts-feeding at workstation. Failure within the machine operation of the subsystems and errors in parts loading lead to increased cycle time. For this reason, automated visual inspection systems play a critical role in monitoring and in providing real-time alerts of irregularities within the workstations and the parts-feeding systems.

2.1.1 Dataset Characteristics

Within automated assembly systems, visual surveillance can be integrated as part of in-line production process or as an externally placed monitoring device located outside of main assembly workstations. In this study external observers are used to obtain surveillance videos of modular digital assembly blocks, linear motor-based conveyance, and vibratory bowl-feeder machine.

The modular design of digital assembly system aims to assemble bike-lights using its rapid speed matching (RSM) robotic arm and transfer components under two hundred strokes per minute. The electronic cam-driven system synchronizes motion between modular assembly blocks for faster component transfer. Therefore, in normal operation, the

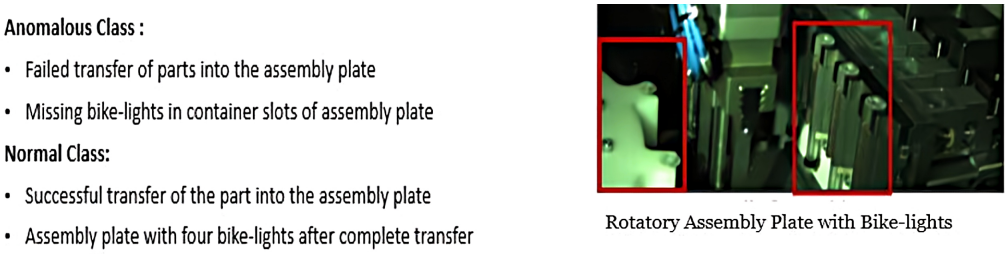


Figure 2.1: Normal Operation of the Bike-light digital assembly

coordinated multi-axis motion control enables synchronous transfer of bike-light base component from a linear escapement to a dial-type part feeder configuration, where the RSM robotic arm transfers component into the fixtures of assembly pallet within a single station configuration. For each assembly module, the complete transfer of parts consists of one-part placement within linear escapement, two components placed in dial-type feeder and four components placed in assembly pallet, as illustrated in Figures 2.1 and 2.2. Due to parts-feeding errors and technical failures in assembly system, the anomalies consist of missing circular-shaped components within the various assembly modules. The linear

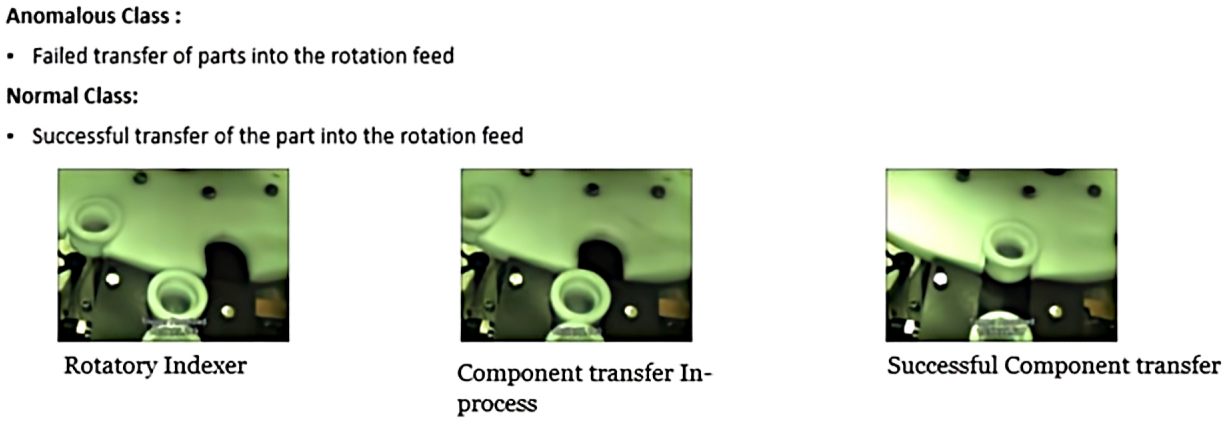


Figure 2.2: Normal Operation of rotational indexer within the Bike-light digital assembly

motor-based conveyance is an assembly workstation with a carousel track configuration and affixed magnetic assembly carriers. By controlling the activation of stationary elec-

tromagnetic coils embedded within straight and curved track segments, the assembly carriers adapt an asynchronous motion for transferring base components to serial assembly workstations[69]. In normal operation, each assembly carrier transports base component of same shape and with a constant component orientation, as illustrated in Figure 2.3. In this instance, rectangular-shaped objects are transported by the conveyor system.Main anomalies in system consist of absence of parts, changed component orientation, and new part-type insertion of different shape. Additional anomaly instances include machine operator intervention and temporal deviations in assembly carrier movement. The vibratory

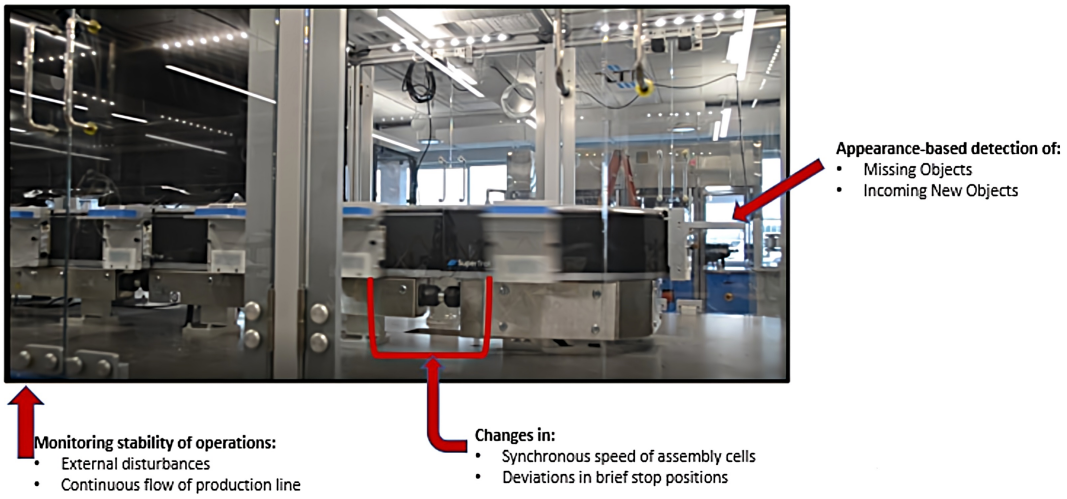


Figure 2.3: Normal and anomalous spatio-temporal behavior of the conveyance system

bowl-feeder has different shelf levels, which consist of variations in the accumulation of parts.The design of the bowl-feeder shelves enables decreased accumulation of parts towards the bowl-feeder exit and controls the orientation of parts within the outer shelf[69]. In normal operation, the parts within the bowl-feeder outer-shelf exit at a particular orientation. During anomalous instances, a part jam occurs during the bowl-feeder outer exit due to change in part-type orientation, as illustrated in Figure 2.4.

As the anomalies consist of spatial deviations within the assembly machine parts, a time series data analysis is performed to spatially segment varying regions of interest within the assembly machine videos. To detect spatiotemporal irregularities within the assembly machine videos, the extracted features from spatiotemporal segmentation are compared with the normal behavior characteristics of the spatiotemporal features.

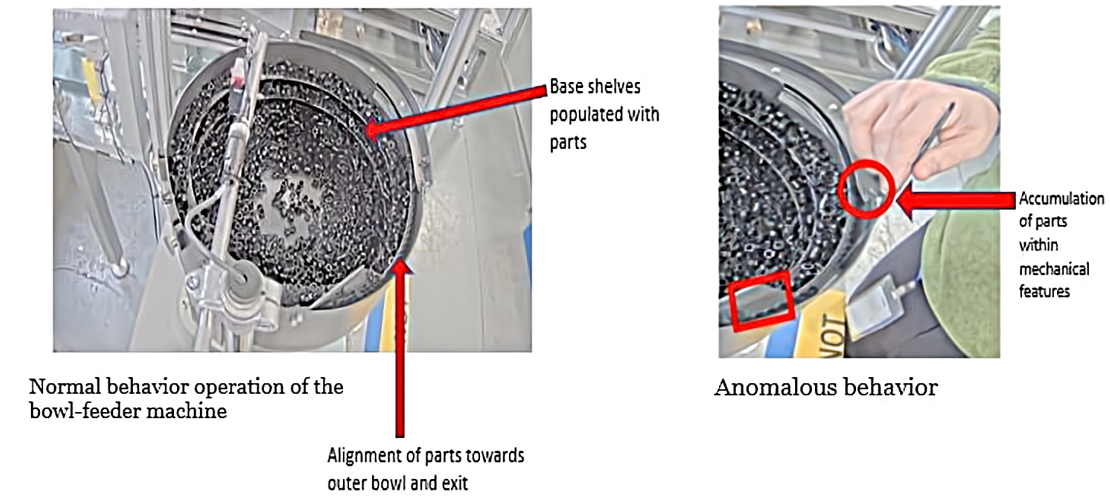


Figure 2.4: Normal and anomalous spatio-temporal behavior of the vibratory bowl-feeder machine

Based on the dataset characteristics outlined in Figure 2.5, each dataset consists of varying parameters such as appearance-based anomalies of different shapes, parts with synchronous or asynchronous motion, videos consisting of multiple camera viewpoints, changing locations of anomalies within the subsystems, and varying texture of foreground objects. Specifically, the appearance-based anomalies consist of missing parts such as rectangular-shaped objects in assembly cells of the conveyor system, circular-shaped objects in digital assembly, and parts accumulation within the outer-shelf region of the vibratory bowl-feeder exit.

2.2 Motion-based Segmentation

This section outlines motion-based segmentation techniques such as moving object detection and temporal segmentation. Through the application of motion-based segmentation, foreground parts can be segmented based on shared motion characteristics. This is significant towards addressing limitations in spatial part characteristics and segmentation.

Videos	Data-Type	Anomalies (Appearance-based)	Moving objects	Challenges	Anomaly Locations	Camera View	ROI Extracts
Dataset 1	Conveyor Belt	Missing objects from assembly cells	Synchronous movement of assembly cells	• Illumination changes • Movement of Camera • Occlusions	Assembly cells (upper surface)	Side	3-4 objects
Dataset 2	Bike-light Assembly Line	Missing bike-lights at rotary wheel and assembly plate	• Robotic manipulator • Rotary wheel • Assembly plate	• Occlusion • Segmentation masks • Multiple Moving objects	• Rotary wheel • Assembly plate	Side	4 objects (assembly plate) – fragments 2 objects (rotary wheel)
Dataset 3	Bowl-Feeder	Accumulation of parts at bowl exit and upper shelve of bowl-feeder	Rubber rings at different bowl shelves	• Human intervention • Inconsistent frame rates	• Bowl exit • Upper shelve	Top	ROIs specified (Small Objects)

Figure 2.5: Specifications of varying dataset characteristics and anomalous behavior within different assembly machines datasets

2.2.1 Moving Object Detection

To detect moving objects in real-time, the local optical flow method of Lucas Kanade is evaluated, which estimates the sparse motion between two consecutive frames using the corner features extracted, as illustrated in Figure 2.7. To solve for the pixel displacement between consecutive frames, the Lucas Kanade method assumes brightness consistency,in which the pixel brightness intensity with respect to changes in pixel position over time remains the same, as illustrated in Figure 2.6. The application of the local method attenuates noise through sparse feature extraction and increases detection in presence of limited visual characteristics.

Although the Lucas-Kanade method attenuates noise to increase the accurately detect moving objects, the sparse feature extraction causes a decrease in the number of object detection. To increase real-time detection of moving objects, the global Gunnar-Farneback optical flow method is applied, which estimates the dense motion between two consecutive frames based on polynomial expansion. The method models image intensity by approximating the pixel local neighborhood using a quadratic polynomial f_1 as defined in Equation 2.1, in which, the polynomial variables, 1, x^2, y^2, x, y, xy, represent the pixel values, and the coefficients A, b, c represent as the symmetric matrix, vector, and scalar values, respectively. To show pixel motion, a second new signal f_2, with a global displacement of d,

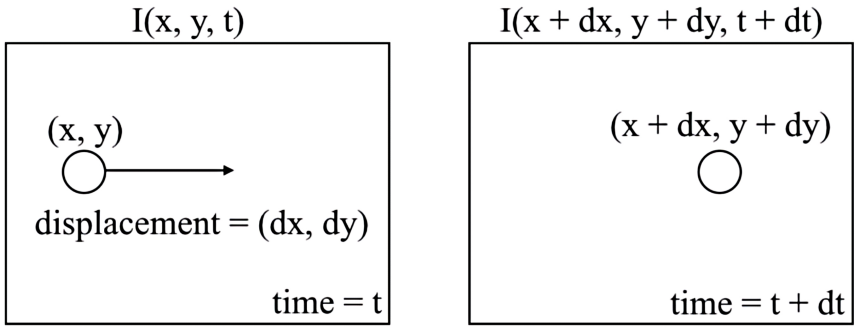


Figure 2.6: Optical flow computation to solve pixel displacement in consecutive frames[26]

is used to represent the consecutive frame, as outlined in Equation 2.2. This is significant as the method assumes constant brightness intensity in pixel values in order to equate the coefficients in Equations 2.1 and 2.2 and solve for the pixel displacement, as outlined in Equations 2.3.

$$f_1(\mathbf{x}) = \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + \mathbf{b}_1^T \mathbf{x} + c_1 \tag{2.1}$$

$$f_2(\mathbf{x}) = f_1(\mathbf{x} - \mathbf{d}) = (\mathbf{x} - \mathbf{d})^T \mathbf{A}_1 (\mathbf{x} - \mathbf{d}) + \mathbf{b}_1^T (\mathbf{x} - \mathbf{d}) + c_1 \tag{2.2}$$

$$\begin{aligned} \mathbf{A}_2 &= \mathbf{A}_1 \\ \mathbf{b}_2 &= \mathbf{b}_1 - 2\mathbf{A}_1 \mathbf{d} \\ c_2 &= \mathbf{d}^T \mathbf{A}_1 \mathbf{d} - \mathbf{b}_1^T \mathbf{d} + c_1. \end{aligned} \tag{2.3}$$

As limitations in visual characteristics such as discontinuities in surface illumination, reflectance, and occlusion cause pixel intensity variation in consecutive frames, the assumptions to compute optical flow with traditional local and global methods would not be met. Therefore, fewer moving object detections can result due to decreased number of pixel features extracted and tracked in presence of pixel intensity variation. In order to increase accuracy in real-time detection of moving parts, the Recurrent All-Pairs Field Transforms (RAFT) deep network architecture is applied for optical flow.

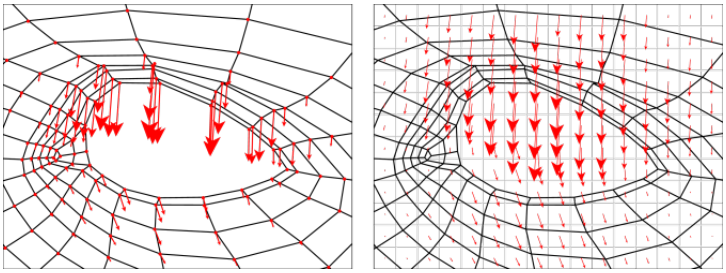


Figure 2.7: Sparse flow field(left) and dense flow field(right)[97]

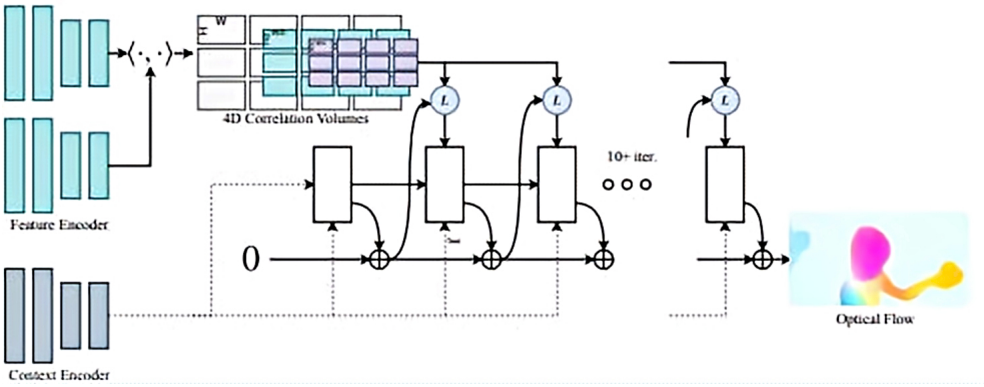


Figure 2.8: Model architecture: RAFT deep learning-based optical flow[88]

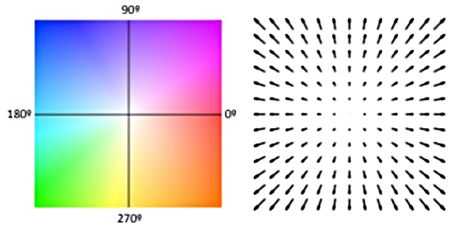
To estimate a dense displacement field, the RAFT method maps each pixel in image $I1$ to its corresponding pixel in image $I2$. To formulate a robust optical flow approach, the RAFT architecture consists of the following modules: feature extraction, computing visual similarity, and iterative updates, as illustrated in Figure 2.7[88][7]. Specifically, the feature extractors consist of a feature encoder network and context network, whose architecture is based on the convolutional neural network (CNN) consisting of 6 residual blocks.

The convolution layers extract low-level features with higher filter sizes in initial layers and high-level features with lower filter sizes in latter layers. With the application of different filter sizes, patch-based feature extraction attenuates noise in image regions with surface illumination. Therefore, a higher number of moving object detection can result in presence of limited visual characteristics. The visual similarity module aims to calculate

the correlation between all-pairs feature vectors of the two feature maps, f_1 of I_1 and f_2 of I_2 [88]. The correlation volume is computed by taking the dot product between all pairs of the feature vectors, as shown in Equation 2.4.

$$C_{ijkl} = \sum_{h} g_\theta\left(I_1\right)_{ijh} \cdot g_\theta\left(I_2\right)_{klh} \tag{2.4}$$

To detect objects with large displacement, a 4-layer correlation pyramid (C_1, C_2, C_3, C_4) is constructed by pooling the two latter dimensions of the correlation volume with multi-scale filters. The update iterator produces an update flow direction,which is applied to the current flow estimate[88][26]. In order to compute the flow direction update, the update operator uses an optimized GRU cell, in which each iteration of GRU takes the concatenation of flow, correlation, and context features as inputs. The flow prediction outputted by the GRU cell is upsampled and visualized using the optical flow color wheel illustrated in Figure 2.8, in which the flow magnitude is represented by varied colors and flow direction is represented by the intensity of the varied colors.



Visualization of the flow field, proposed by Baker et al(2007). Left side: Color code visualization of flow direction illustrated with different colors and flow magnitude illustrated with color intensity. Right side: Arrow visualization

Figure 2.9: Optical flow field visualization with variation in colors and intensity[7]

2.2.2 Temporal Segmentation

The ST-GNN model visualizes each video frame as a graph representation, in which the nodes represent the image patches, the blue edges, as illustrated in Figure 2.9, locally correspond to image patches in time, and the query node represents the initial image patch in the first frame, and the target node is set as the initial node as well[37]. Due to the challenge of the limited ground truth labels in label propagation for video object

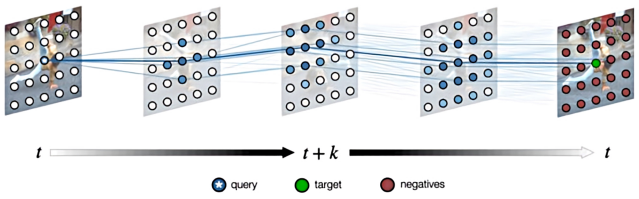


Figure 2.10: Representation of video as a space-time graph[37]

segmentation, the model applies cycle-consistency in learning visual correspondence across query and target node across space-time. Specifically, the aim of the random walker is to associate the nodes across space-time.

2.3 Appearance-based Segmentation

The appearance-based methods segment objects based on spatial characteristics, such as object shape, color and texture. In this section, the various appearance-based methods can be applied to segment spatial part features using pixel characteristics, user feedback and object classification.

2.3.1 Pixel-based Segmentation

The GLCM is a second-order statistical texture method, which characterizes the texture of an image by calculating how often a pixel with intensity (gray-level) value i occurs in a spatial relationship to a pixel with value j[63]. The spatial relationship is defined as the pixel of interest and the adjacent pixel located either horizontally (0), vertically (90), or diagonally (-45, -135), as illustrated in Figure 2.10. Therefore, the spatial relationship between pixels is characterized by constant pixel intensity at a given orientation and a distance d. By calculation of the pixel spatial relationship using the GLCM function, the GLCM matrix can be obtained. This is significant as the statistical measures such as variance, correlation, homogeneity, and energy values can be extracted from the GLCM matrix[72][49]. The variation in Equation 2.8 measures the local variation in the GLCM matrix, energy in Equation 2.6 measures the joint probability occurrence of a particular

pixel pair, homogeneity in Equation 2.7 measures the spatial distribution of elements, and the correlation in Equation 2.5 derives the sum of squared values in GLCM.These statistical values, obtained from the GLCM matrix, are significant for performing texture classification in image objects[72][62][15][36].

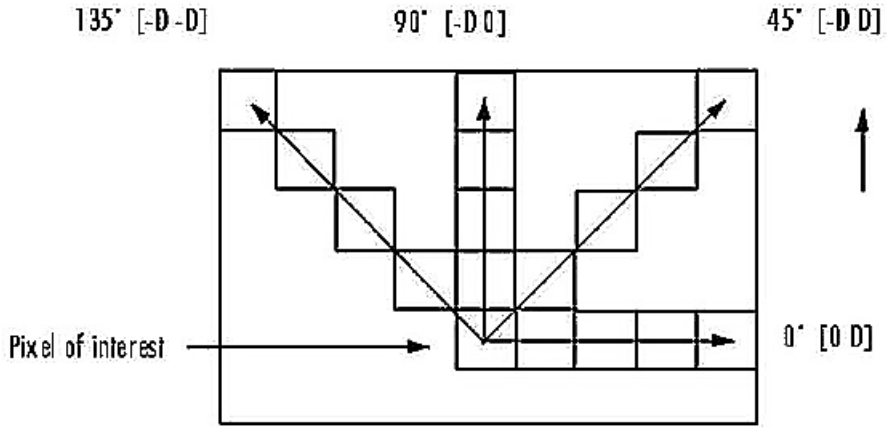


Figure 2.11: Spatial relationships of pixels, in which D is defined as the distance from the pixel of interest[36]

$$-\sum_{i,j}\frac{(i-\mu)(j-\mu)}{\sigma^2}P(i,j) \tag{2.5}$$

$$\sum_{i,j}P(i,j)^2 \tag{2.6}$$

$$\sum_{i,j}\frac{1}{1+(i-j)^2}P(i,j) \tag{2.7}$$

$$\sum_{i,j}(i-\mu)^2P(i,j) \tag{2.8}$$

To analyze texture based on pixel spatial locality, orientation selectivity, and frequency characteristics, the 2D Gabor filters are applied. The 2D Gabor filters are represented

as a sinusoidal signal, in which the signal frequency and orientation are modulated by a gaussian wave. To detect the presence of specific frequency bandwidth within a localized image region, the orthogonal directions of the 2D Gabor filter can be fine-tuned. The orthogonal directions consist of an imaginary and a real component which can be formed into a complex number as defined in Equation 2.9[80][77]. Within the complex number, the signal wavelength, represented by lambda, controls the width of texture strips, the direction, represented by theta, controls the orientation of strips, the phase offset, the psi, controls the phase difference, the standard deviation, sigma, controls the bandwidth size with the number of texture strips included within, and the aspect ratio, the gamma, controls the height of the Gabor function. The texture parameters within the complex equation are significant in thresholding the number of objects detected. Within the texture analysis, the texture can be segmented based on the texture repetition or tone characteristics defined by the spatial relationship of pixels, and the texture complexity or structural arrangement of the texture primitives[96][80].

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x'}{\lambda} + \psi\right) \tag{2.9}$$

In order to segment objects based on texture complexity, entropy-based segmentation is applied to analyze local entropy or complexity within a local pixel neighborhood. Through the application of an entropy filter, subtle variations within the local gray level distributions can be detected. The local entropy is measured by applying a structuring element, consisting of a specific radius value, to capture the local grey level distribution. This is significant as increasing or decreasing the disk radius of the entropy filter causes the image to blur or sharpen, respectively. The sharpened image increases the detection of texture complexity and leads to an increased binary segmentation of objects within an image[103][57][27].

As the illumination changes within different regions of the bowl-feeder cause sharp discontinuities in pixel variation, edge-based segmentation is evaluated by applying the canny edge detector method. As sharp discontinuity in pixel gray levels lead to edge formation within the image regions, the canny detector detects edges by measurement of gradient magnitude and direction. To detect the edges with the canny detector, the steps consist of noise reduction, gradient calculation, non-maximum suppression, double threshold, and Edge tracking by hysteresis[76].

The noise reduction is performed by applying a gaussian smoothing filter of a specific kernel size to blur regions within the image. The gaussian convolution masks, as defined in Equations 2.10 and 2.11, are applied in the X and Y directions. Afterward, gradient magni-

tude and direction are detected in Equations 2.12 and 2.13, which can represent the blurred images' horizontal, vertical and diagonal edges. The gradient direction is perpendicular to the edges[104]. Non-maximum suppression is significant in removing pixels, which are not considered part of the edges. The hysteresis thresholding is further performed to threshold pixels as part of the edges or the background image regions. Specifically, a pixel gradient higher than the upper threshold is classified as part of the edge. The meanshift is an unsupervised clustering algorithm, which detects blobs in a smooth density of samples. The centroid-based method updates the centroid value as the mean of points within a given image region. The mean shift steps consist of forming a sliding cluster for each data point, each sliding window is shifted towards higher density regions by shifting the regions, specific sliding windows are selected by deleting overlapping windows, and data points are updated iteratively to each of the sliding window[104][86].

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{2.10}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \tag{2.11}$$

$$\mathrm{G} = \sqrt{\mathrm{G_x^2} + \mathrm{G_y^2}} \tag{2.12}$$

$$\theta = \arctan\left(\frac{\mathrm{G_y}}{\mathrm{G_x}}\right) \tag{2.13}$$

2.3.2 Interactive-based Segmentation

To compare the performance of the GrabCut annotation tool with deep learning-based interactive segmentation, the feature-backpropagating refinement scheme (F-BRS) model is applied. The F-BRS model applies the DeepLabV3+ network architecture, which is trained on the Semantic Boundaries Dataset (SBD) 8,498 images and annotations of object classes such as vehicles, households, and animals. The DeepLabV3+ consists of a ResNet backbone with atrous convolutions for feature extraction, atrous spatial pyramid pooling (ASPP) module for resampling feature map at different rates and segmenting the object at multiple scales using semantic segmentation annotations, and a 1x1 convolution to

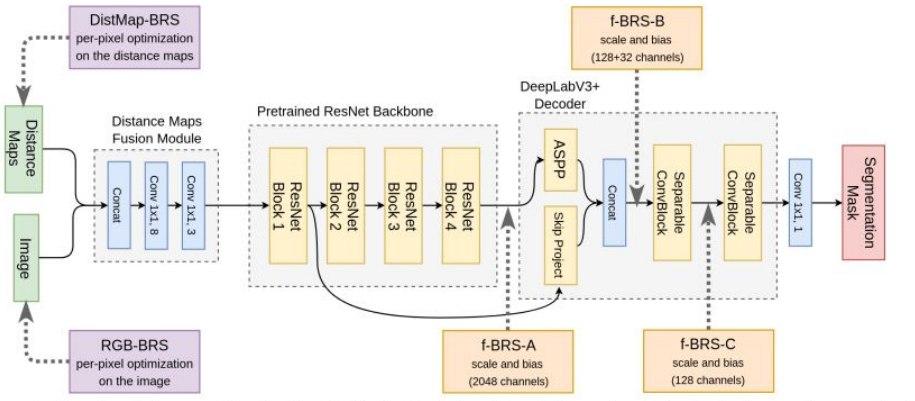


Figure 2.12: Model architecture: F-BRS Interactive Segmentation[42]

output final segmentation from concatenated masks, as illustrated in Figure 2.12[83]. The pre-trained FBRS segmentation model segments objects using positive and negative clicks from user-based feedback. The positive and negative clicks guide the generation of semantic segmentation masks on the various classes of objects.

2.3.3 Supervised Segmentation

Specifically, the feature pyramid network for feature extraction, the region proposal network for detection of regions with objects using multi-scale features, and the box head for classifying objects within the bounding box, as illustrated in Figure 2.13. The feature pyramid network applies a batch of images, of constant height and width, as input to extract features and output feature maps at different scales. For multi-scale feature maps in detection, the backbone of Base-RCNN-FPN consists of the ResNet50 block structure with stem block and multiple bottleneck blocks at various stages. The stem block performs down-sampling of the input using strided convolutions at a specified kernel size and outputs the feature map tensor. The bottleneck residual block reduces the number of parameters and matrix multiplication for dimensionality reduction. From the res2-res5 stages, the four tensors are generated: P2 at $\frac{1}{4}$ scale, P3 at $\frac{1}{8}$ scale, P4 at $\frac{1}{16}$ scale, and P5 at $\frac{1}{32}$ scale. The lastlevelmaxpool, at a specified kernel size, is applied to down-sample P5 to $\frac{1}{64}$ scale features and generates P6 output.

The feature extraction at different scales is significant to identify and extract pixel

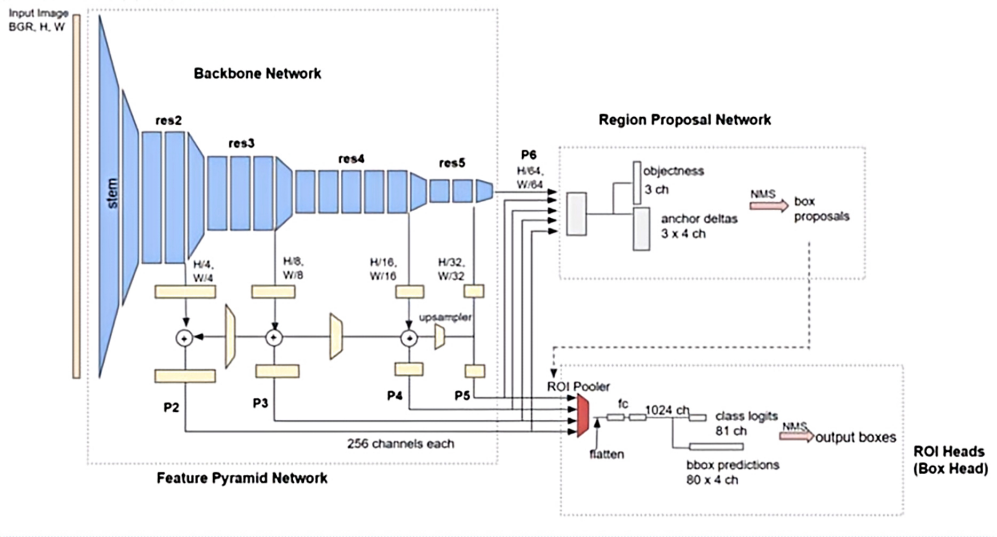


Figure 2.13: Model Architecture of Detectron2[32]

features in the presence of limited visual characteristics such as illumination variance and occlusion[98][81]. The multi-scale feature maps are then applied as inputs to the region proposal network, which detects regions of smaller objects, from P2 to P5, and larger objects, from P4 to P6. The process overview consists of generating an objectness map and a prediction of a relative box shape to anchors using the RPN head, generating anchors to align the objectness map to the ground truth boxes, associating ground truth boxes with generated anchors using Intersection-over-Unit (IoU) Matrix, optimizing location prediction of boxes using loss calculation in training, and selecting region proposal boxes based on predicted objectness score[81]. The box head applies the output of proposal boxes to warp feature maps into fixed-size features and generates fine-tuned box locations. Afterward, the box head classifies the object within the region of interest and applies further fine-tuning of box position and shape based on localization and classification loss[81].

2.4 Preprocessing Methods

This section outlines the preprocessing methods, which can be applied to address limited visual characteristics. The preprocessing methods consist of morphological operations and image registration techniques.The externally placed visual surveillance captures regions

of interest (ROIs) from multiple camera viewpoints. In order to compare spatiotemporal characteristics of the ROIs, an affine-based transform is applied for image registration. By application of such preprocessing methods, the visual characteristics can be addressed to increased accuracy in spatial anomaly detection.

2.4.1 Image Preprocessing

The opening applied on X&Y is the union of translations of Y, which fit within X, as shown in Equation 2.14[79]. This operation is equivalent to applying erosion followed by dilation, in which the structuring element of a specified kernel size removes thin protrusions from the foreground object and then adds pixels to the object boundaries. Therefore, removing the internal noise of the foreground objects. To break the fusion of joined segmentation masks, the closing operation is applied on X&Y, which is the complement of the union of Y translations that do not fit within X, as shown in Equation 2.15. This operation aims to dilate the foreground object and remove pixels at the object boundaries. Therefore, small holes within the foreground objects and mask fusion are eliminated by using this operation[79][19].

$$A \circ B = (A \ominus B) \oplus B \tag{2.14}$$

$$A \bullet B = (A \oplus B) \ominus B \tag{2.15}$$

2.4.2 Image Registration

The affine transformation matrix applies shear, scale, rotation, and translation as defined in transformation matrix in Equation 2.16. The affine transformation consists of 6 DoF with translation applied. The geometric transformation is performed to warp an image of interest to the reference image. Through image registration, images, captured from multiple viewpoints, can be transformed into the same coordinate system.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2.16}$$

2.5 Chapter Summary

This section presented the various approaches and working principle of motion-based and appearance-based methods. The outlined methods can be applied for foreground part segmentation within various assembly machines. By outlining the working principle of these methods, the main objective is to understand their technical advantages and disadvantages. For instance, the application of deep learning-based optical flow can increase detection accuracy in presence of limited visual characteristics. In comparison to the RAFT method, the assumptions of local and global methods limit accuracy in detection performance. By understanding the dataset characteristics and technical advantages of various methods, the research objectives can be met with increased accuracy and efficiency. Through the assessment of different methods, the appropriate methods to perform spatiotemporal segmentation are selected.

Chapter 3

Related Work

Within current research studies, automated visual inspection systems are integrated inline within the manufacturing process to perform anomaly detection within the various image datasets. Similar to the videos applied in this study, the pre-existing manufacturing datasets consist of varying anomaly specification, limited visual characteristics, and assembly parts of similar spatial characteristics. Therefore, various anomaly detection and image processing methods are explored to accurately detect spatial and temporal anomalies. This section aims to outline the related works on image processing methods, motion-based and appearance-based segmentation. In addition to handcrafted methods and deep learning-based models evaluated to perform anomaly detection in real-time manufacturing applications.

3.1 Motion-based Features

This section outlines motion-based segmentation methods applied to detect moving parts in real-time within the automated assembly system. According to the varying anomaly specification and dataset characteristics, the related works focus on motion-based segmentation such as background subtraction, optical flow and thresholding techniques.

3.1.1 Moving Object Detection

As assembly machines consist of different configurations, moving assembly parts regulate in synchronous, asynchronous, or continuous motion during the execution of main assembly

operations. To detect the spatiotemporal part irregularities during the machine operation, spatiotemporal part characteristics of the current frame are compared in a time-series analysis with normal behavior spatiotemporal part characteristics. In order to segment spatiotemporal part characteristics, motion-based segmentation methods are applied. As assembly parts consist of shared motion characteristics, real-time object detection is performed to identify and localize moving parts within the assembly machine. Specifically, for the application of visual inspection within manufacturing systems, real-time object detection methods such as background subtraction, optical flow, frame-differencing, and sensor-based detection are applied.

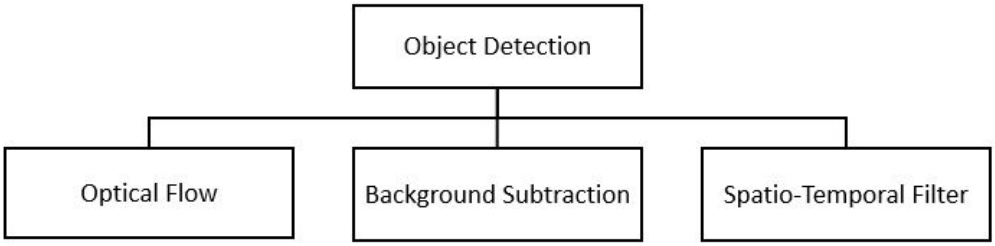


Figure 3.1: Taxonomy of moving object detection techniques

As the datasets consist of varying anomaly specifications and limited visual characteristics, different motion-based techniques are performed to increase the detection accuracy of moving objects. In Arnal et al, the visual inspection within an automobile production line is performed using the local optical flow method of Lucas Kanade to inspect defects within car body surfaces[6]. Specifically, approaches such as image fusion techniques, deflectometry principle, and optical flow are evaluated to detect small deformations within car body surfaces. As small defects cause the surface to become non-specular or less specular, small defects such as dings and dents cannot be detected by fusion image techniques and deflectometry principles. Specifically, these methods rely on the triangulation method and deviation of light change in the shape of the pattern to detect the presence of defects. As small defects can appear less specular, the research study applies a combination of optical flow and deflectometry principal approach to detect anomalies of different dimensions[6]. For the computation of global optical flow, Dosovitskiy et al, estimates the motion through the deep learning-based optical flow to perform motion detection[34].

As optical flow assumes brightness constancy, additional motion-based segmentation techniques can be applied within manufacturing to distinguish between foreground and background objects. In their work, Chen et al, detect motion-based objects in a surveillance system using low-rankness with a regularization method[44][17]. Background subtraction techniques can be applied within manufacturing systems, which can distinguish the foreground from the background in video frames. Additionally, in Rembold et al, the detection and handling of rectangular-shaped objects within the conveyor belt are evaluated using the methods: background monitoring, template refreshing, and template library. Within the first method, the background of the video is set as the template and the search window measures distortion within the template to detect the presence of the moving object. Although the background monitoring method detects the presence of moving objects, factors such as illumination variation, and a lower quality of grid granularity in the template can result in decreased detection accuracy of the position and rotation of the object. Therefore, to accurately detect object position and orientation, the second method detects the foreground object using a binary image, in which the change in pixel intensity indicates the presence of the moving object. With the detection of the foreground object, the second method refreshes the template to detect the presence of the next incoming object. As the template refreshing method is unable to detect concave objects, a template library is created using the third method, in which objects, at different orientations and positions, are saved into memory. With template matching, moving objects at different positions and orientations are detected in the video frames[74].

Additionally, sensor-based techniques are also applied to detect motion in manufacturing systems. In Chavez-Garcia et al, the color sensor and metal detector are applied to detect the spatial features of the object. Additionally, an ultrasonic distance detection sensor is placed on the conveyor line to monitor the movement of the moving object within different regions of the conveyor belt. With the application of vision-based and sensor-based detection, assembly parts with defects can be detected and identified for separation in the assembly line[16].

3.1.2 Moving Object Tracking

As moving assembly parts can be detected and identified within the assembly machines using real-time detection techniques, motion tracking methods are applied to track the movement of the detected object in consecutive frames. Through object tracking, the spatiotemporal part characteristics in each frame can be obtained for comparison with normal behavior part characteristics and to perform anomaly detection. Therefore, various object tracking methods are applied within manufacturing systems to continuously

track objects in presence of limited visual characteristics and with varying anomaly specifications for the real-time surveillance system. In Burt et al, the method focuses on applying high-speed feature detection and hierarchical scaling of images to perform real-time surveillance[44][45]. Wiklund et al apply image differencing methods to perform motion tracking of objects[5][13]. In Goldberg et al, motion tracking is performed using temporal filtering with vision hardware[44]. In Wessel et al, real-time application of motion tracking is performed using Horn and Schunk's method[5]. In his research study, Safadi applies a tracking filter and a pyramid-based vision system to perform motion tracking[5].Durrant-White and Rao apply a Kalman filter-based tracking to track object motion using multiple cameras[5]. Miller integrates a camera and an arm to perform a tracking track, in which the kinematic and control parameters of the system serve as the learning objective[5]. In Koller et al, the motion tracker consisted of two Kalman filters, in which the first filter was estimated for the position and the second filter was used for the shape estimation of moving vehicle traffic[93]. Similarly, Meyer applied motion filtering toward position estimation through the application of a motion filter to measure the affine parameters of an object[93].

3.1.3 Temporal Segmentation

Through the application of object detection and motion tracking, the moving assembly parts are detected in real-time, and the parts are tracked in consecutive frames using motion. As spatiotemporal characteristics of parts can be segmented using the following computer vision techniques, the limitations in visual characteristics result in decreased accuracy in the performance of real-time object detection and motion tracking methods. In order to increase accuracy in spatiotemporal segmentation, supervised and unsupervised temporal segmentation methods can be performed within manufacturing applications. In Nakamura et al, semi-supervised temporal segmentation is applied to segment a specialized vehicle for manufacturing applications[64]. To perform supervised segmentation, Zhao et al apply a two-stream network architecture for action localization, in which one stream applies feature extraction from input video frames and the other stream, computes the optical flow of features extracted[56]. Temporal segmentation can be applied in manufacturing to perform action recognition and detect external disturbances caused due machine operator activity. In a similar case, Zhou et al performed the unsupervised task of action segmentation through the maximization problem of the clustering score. Specifically, the method divides the input frames into segments in order for the clustering score to be maximized[104].

3.2 Appearance-based Features

This section outlines appearance-based segmentation methods applied to detect parts in real-time within the automated assembly system. Based on the part spatial characteristics, the related works focus on various appearance-based segmentation such as pixel-based characteristics and supervised object detection methods to group similar foreground objects within the manufacturing image datasets.

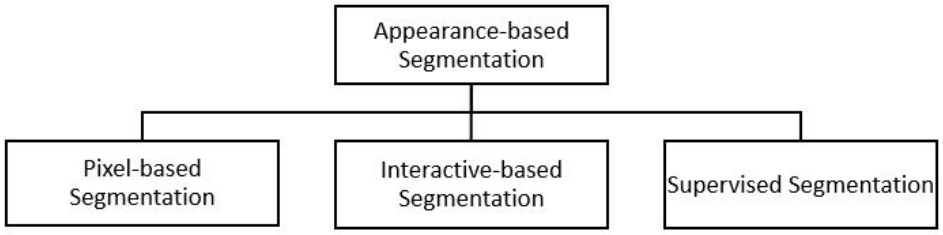


Figure 3.2: Taxonomy of appearance-based techniques to segment foreground objects based on spatial characteristics

3.2.1 Pixel-based Segmentation

As motion-based methods allow characterization and segmentation of parts based on shared motion characteristics, appearance-based methods can be applied to segment parts based on spatial features such as texture, color, and shape. In order to detect part irregularities within assembly machines, detection and segmentation of spatial part characteristics in each frame can be compared with normal spatial characteristics of parts. In texture segmentation, the features can be segmented using methods such as wavelength transform, morphological filter, Gabor filter, etc. Farrokhnia et al applied texture features, calculated from Gabor filters, to classify the uniformity of the painted metallic surface[90]. Serra et al applied morphological operations to detect defects in wood. Similarly, Distante et al applied oriental texture analysis with a morphological approach for leather inspection[90]. Additionally, color-based segmentation can be applied to perform defect detection in various industrial applications. Specifically, in detecting defects within the food production

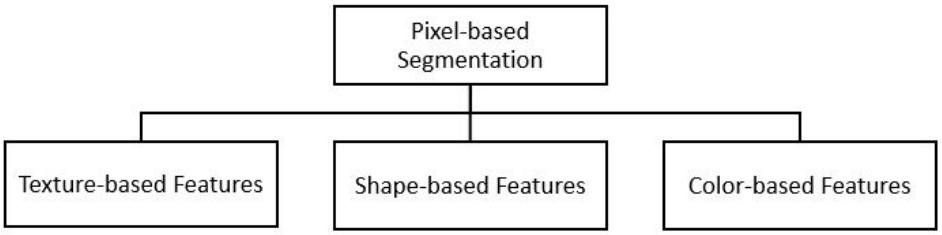


Figure 3.3: Taxonomy of pixel-based techniques to segment spatial characteristics: texture, shape and color of foreground objects

line. As the food production line requires vision-based tasks related to enhancement, recognition, and visualization, the color-based methods allow segmentation of assembly products in presence of limited visual characteristics. In Loresco et al, color space analysis is performed using KNN for lettuce crop stages identification in a smart farm setup. Specifically, K-nearest neighboring is used to perform image segmentation for the RGB, HSV, and YCbCr color spaces[52]. Different color-based segmentation methods are applied to obtain the best color space for the identification of the different growth stages within the agricultural process. Similarly, pixel-based methods can be applied in the segmentation of object shapes. As assembly machines assume the same shape of part type, shape-based segmentation can be used to detect spatial anomalies such as missing objects and the insertion of new part-type of varying shapes. In Lou et al, watershed segmentation is applied to extract topological features in additively manufactured surfaces and perform anomaly detection analysis.

3.2.2 Interactive Segmentation

As limitations in visual characteristics such as occlusion, and illumination changes can result in pixel intensity variation, evaluation of pixel-based segmentation methods can show decreased detection accuracy of assembly parts. In order to increase detection accuracy in the presence of limited visual characteristics, user feedback can be integrated within the vision-based system to group similar objects. In Oh et al, the method consists of applying interactive segmentation to perform object decomposition, in which user feedback is used for part-type segmentation and surface-type segmentation[67]. Additionally, deep learning-based methods can be applied to perform interactive segmentation within applications in

various industries. For example, Xue et al, apply deep learning-based interactive segmentation, which applies click points and boundary boxes as inputs to train the CNN-based model for annotation tasks[103]. Similarly, Maninis et al apply points such as left-most, top, right-most, and bottom pixels as input to train the proposed DEXTR model in the segmentation of objects[61].

3.2.3 Supervised Object Detection and Segmentation

Although interactive-based segmentation increases accuracy in object segmentation during the presence of limited visual characteristics, the repeated user feedback required decreases efficiency and causes a slower model deployment in real-time applications. Therefore, supervised detection methods can be applied to segment parts accurately and efficiently. For instance, a Region-based convolutional neural network illustrates accuracy in the detection and localization of objects within images. Mask RCNN model, an extension of Faster RCNN, can be applied to perform region segmentation at the pixel level. To perform object detection in images at multiple scales, Farhadi et al generate predictions at different layers of the feature extraction network. Mangat et al apply a supervised learning approach to train object detection model and detection of low-texture objects within a manufacturing setting. Specifically, by applying YOLOv3 towards object detection in synthetic manufacturing dataset[60].

3.3 Assembly Machines Videos: Anomaly Detection

This section outlines the application of handcrafted methods for spatial feature detection. The extracted spatial part characteristics in current frame can be compared with normal behavior part characteristics to perform spatial anomaly detection.

3.3.1 Hand-Crafted Features

To perform anomaly detection within manufacturing applications, handcrafted methods can be used to perform feature extraction and for the detection of anomalies in various datasets. For instance, grey level co-occurrence matrices and local binary patterns can be applied to perform spatial feature extraction and perform surface defect detection. Spectral feature extraction based on Gabor transforms and wavelet transforms can be applied for defect detection in textured surfaces. The application of handcrafted methods can also be

used in real-time crowd anomaly detection. As manufacturing video datasets consist of temporal patterns evident in traffic and crowd activity datasets, handcrafted methods performed in crowd activity and traffic datasets can be applied within manufacturing datasets. For instance, Wang et al apply a spatiotemporal texture model to perform feature extraction, in which the texture feature space is formed using wavelet transform[101].

3.3.2 Chapter Summary

This section outlines the existing related works based on the motion-based and appearance-based segmentation methods. These methods are applied on manufacturing datasets, which consist of similar spatial and temporal characteristics to the datasets applied in this study. By review of these related works, the appropriate motion-based and appearance-based segmentation methods are selected to evaluate on the conveyor system and the bowl-feeder datasets. Specifically, in applying motion-based segmentation methods such as optical flow and background subtraction techniques to segment moving foreground parts within the bowl-feeder machine. The appearance-based segmentation methods based on texture and color space analysis can be applied to segment spatial part characteristics.

Chapter 4

Methodology

In this section, the following methods are evaluated to conduct spatio-temporal segmentation of assembly parts and to perform anomaly detection within various automated manufacturing machines.The outlined bottom-up architecture is applied for motion-based and appearance-based segmentation using computer vision techniques and deep learning-based models. Additionally, image processing methods are evaluated to address limitations within the visual characteristics. By addressing limited visual characteristics and segmentation of the part characteristics, anomaly detection methods are applied on spatial features using handcrafted methods and deep learning-based models.

4.0.1 Illumination Invariance

Figure 4.1: Overview of performing and evaluating color-space segmentation through moving parts detection in assembly machines

As the regions within the bowl-feeder outer shelves consist of illumination variation, the pixel intensity variation due to lighting changes results in a lower number of assembly parts detected. To address illumination variance within the bowl-feeder regions, the image datasets are transformed into grayscale and the hue, saturation, and value (HSV) color space. The converted images are applied to perform background subtraction using Local SVD Binary Pattern (LSBP) and Mixture of Gaussian (MoG) methods. As the color-space conversion and illumination robust background subtraction address illumination variance, increased number of foreground parts can be detected.

4.0.2 Synthetic Dataset Creation

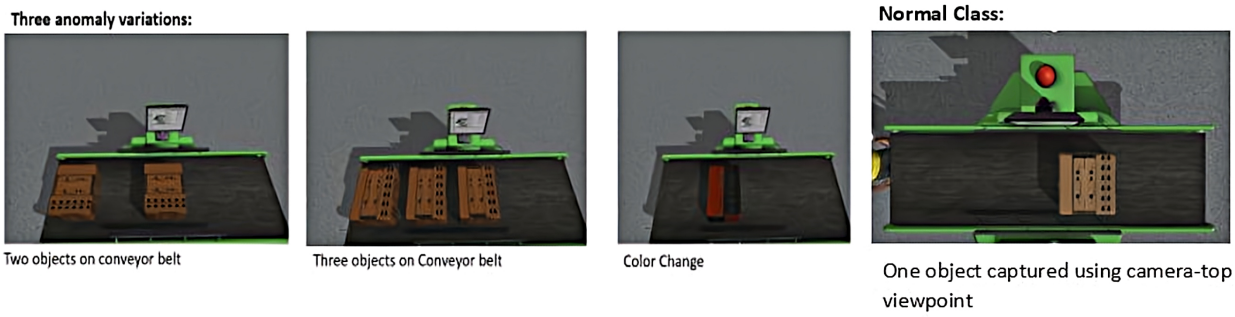


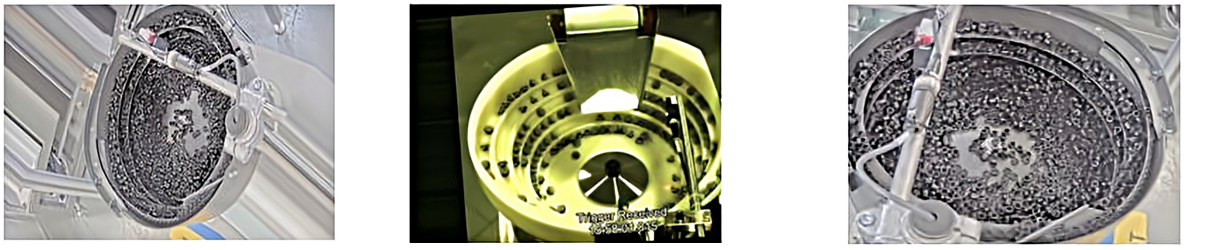
Figure 4.2: Synthetic dataset creation of linear conveyance system with instances of normal and anomalous behavior

As the linear conveyance dataset consists of low image resolution resulting in texture inconsistency, the limited visual characteristics cause image registration errors due to the failure of corresponding texture features between two extracted ROIs. To illustrate image registration between extracted ROIs, a synthetic conveyor dataset is created using the FlexSim manufacturing software. The dataset consists of conveyor belt videos with one object transfer and multiple objects transferred at once to indicate normal and anomalous behavior, respectively.Additional anomalous behaviors consist of part texture and orientation changes within the conveyor belt system. Similar to the real-time linear conveyance system in manufacturing applications, the surveillance videos are captured from multiple and fixed camera viewpoints. This is significant for illustrating image registration by transforming images, of multiple viewpoints, into one coordinate system. To accurately

detect moving parts in real-time within surveillance videos, the conveyor belt is controlled at average speed to allow small object motion and enable accurate moving part detection using optical flow methods.

4.0.3 Data Augmentation

In order to perform supervised object detection and segmentation, deep learning-based methods require datasets for model training and evaluation. Due to the limited datasets in manufacturing and limited visual characteristics such as camera movement, lower image resolution, and occlusion, model training and evaluation results in decreased detection accuracy. To address the challenge of the limited dataset, data augmentation techniques are applied to the bowl-feeder datasets such as brightness intensity increase, zoom, width shift, rotation, and horizontal flip. The brightness intensity increase was applied to train a model for object detection with varying illumination distribution and in instances of pixel intensity variation. The zoom parameter was introduced within the dataset to identify and detect pixel characteristics corresponding to assembly part shape and texture. Additionally, the parameters such as width shift, rotation, and horizontal flip were applied to change the visual appearance of the bowl-feeder. The variational appearance of the bowl-feeder would enable the model to detect assembly parts within different real-time applications of the bowl-feeder machines.



Bowl-feeder visualization with image rotation

Minimal fill of bowl-feeder in increased brightness intensity

Zoomed-in dense part clusters

Figure 4.3: Data augmentation techniques applied on the bowl-feeder dataset, which consists of variational parts-feed

4.1 Motion-based Features

In this section, the motion-based segmentation methods based on traditional computer vision techniques and deep learning models are evaluated. Specifically, in applying techniques to perform moving object detection, pixel-based tracking, and deep learning-based segmentation to visualize current and previously moved parts. Within this study, the RAFT deep learning-based method is applied to compute the moving object flow field. The iteratively updated flow field of RAFT is applied to create an optical flow-based tracker to visualize current and previously moved parts in different colored segments.

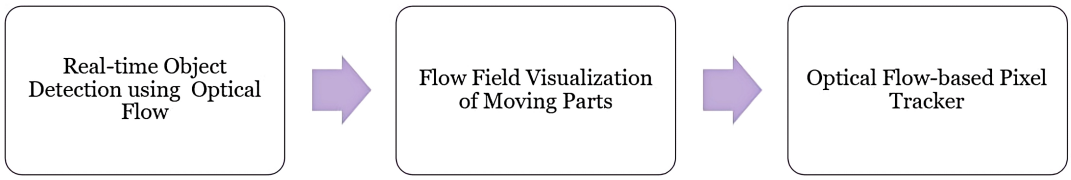


Figure 4.4: Overview of motion-based segmentation method applied

4.1.1 Moving Object Detection

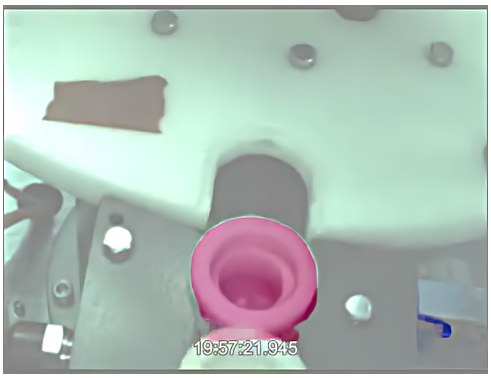


Figure 4.5: Initial evaluation of RAFT optical flow method for moving object detection in digital assembly dataset

The analysis of spatiotemporal irregularities within assembly machines is performed by segmenting moving parts for comparison of their spatiotemporal feature characteristics with the normal behavior feature characteristics. To conduct spatiotemporal object segmentation, the moving parts are identified and localized within the videos using real-time object detection. As all parts within a specific assembly machine consist of synchronous or asynchronous motion, real-time part identification and localization is based on motion characterization of base assembly parts.

Specifically, local and global methods based on traditional computer vision techniques and deep learning-based models are evaluated to perform optical flow on various datasets of assembly system. Additionally, RAFT is applied to detect and visualize the object motion between consecutive frames at different thresholds of the flow magnitude, as illustrated in 4.5. The thresholding of flow magnitude is applied to reduce noise in predicted flow masks and reduce the application of morphological operations in the generation of flow-based binary segmentation masks. Image processing steps are applied to extract regions of moving objects within the bowl-feeder.

Flow-based segmentation of moving objects can consist of varying shapes and sizes. Therefore, variational parameters are required within object contour detection and ROI extraction steps in order to perform further spatial data analysis. To form a pipeline with generalizable image processing steps, inverse binary masks are generated, in which the stationary parts are visualized as regions of interest. As the dense regions of stationary parts are mainly located within the base-shelf of the bowl-feeder machine, the inverse binary segmentation masks would generate ROI masks of uniform shape and size. To visualize the quiver plot in consecutive frames of bowl-feeder videos, the iteratively updated flow field is applied, in which the arrow length and direction illustrate flow magnitude and direction, respectively.

4.1.2 Moving Object Pixel Tracking

To perform pixel-based tracking of moving parts within the assembly machines, the iteratively updated flow field of the RAFT optical flow are applied to visualize previous and current moving parts in different colored segments. The 2D flow field is also simulatenously visualized to illustrate flow magnitude and direction of assembly parts, as illustrated in Figure 4.6. The current and previous colored segments are visualized in white-colored and blue-colored segments, respectively. By visualization of current and previous moved parts, the motion-based segmentation can segment accumulated parts within the bowl-feeder base shelf and moving parts regulating within the outer-shelves of the bowl-feeder.

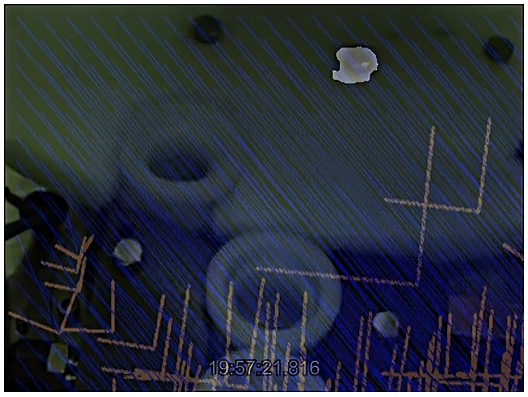


Figure 4.6: Preliminary evaluation of pixel-based tracker on digital assembly dataset for tracking current and previously moved objects

4.1.3 Spatio-Temporal Segmentation

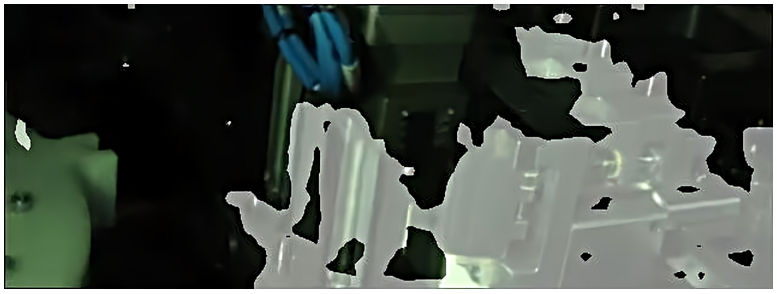


Figure 4.7: Initial evaluation of Space-time Graph Neural Network(ST-GNN) model on digital assembly dataset for spatio-temporal segmentation of moving objects

In order to compare the segmentation performance of pixel-based tracker, the ST-GNN model is applied to conduct temporal segmentation on manufacturing datasets. To perform the temporal segmentation, the model visualizes video frame as a graph representation, as illustrated in Figure 2.9, and learns visual correspondence in palindrome sequences. As the query and target node are defined as the same image patch across time, the model applies cycle-consistency to locally correspond image patches in palindrome sequences.

The model is evaluated on original videos of the bowl-feeder and digital assembly

datasets, as illustrated in Figure 4.7. To initialize the ST-GNN model, a first-frame annotation mask is generated using the RAFT optical flow method.

4.2 Appearance-based Features

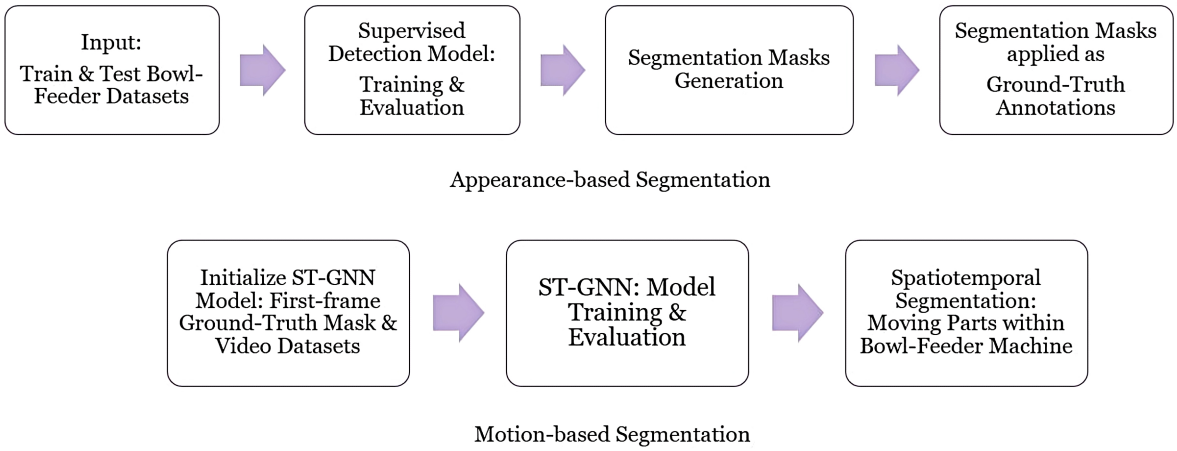


Figure 4.8: Overview of the appearance-based method applied to enable automatic initialization of the ST-GNN model

This section aims to highlight the appearance-based segmentation methods applied to segment foreground assembly parts based on spatial characteristics. As spatial features can be segmented based on pixel characteristics, user-based feedback and object classes, pixel-based segmentation, interactive-based segmentation and supervised segmentation methods are evaluated, respectively. The generated segmentation mask, with increased boundary mask IoU, will be applied as first-frame annotation to enable automatic initialization of the ST-GNN model segmentation model, as shown in methods overview in Figure 4.8.

4.2.1 Pixel-based Segmentation

As the objects can be grouped based on similar texture, shape, and color given that pixel brightness intensity remains constant, pixel-based methods are evaluated to perform appearance-based segmentation. As the moving parts within the vibratory bowl-feeder

consist of a similar spatial arrangement of intensity levels within the pixel local neighborhood, texture-based segmentation methods are evaluated to segment the foreground moving parts and the background bowl-feeder surface.

The texture analysis is performed to group objects based on texture repetition and complexity. To perform texture repetition analysis, the Gray-Level Co-Matrix (GLCM) and Gabor filters are evaluated to characterize texture in similar objects based on pixel orientation and intensity levels. The entropy-based method is evaluated to segment foreground objects based on texture complexity. As the datasets consist of limited visual characteristics, limitations such as illumination variance can cause pixel intensity variation and errors in texture-based segmentation. To increase accuracy in pixel-based segmentation, shape-based and color-based segmentation methods are evaluated. Specifically, the clustering-based meanshift segmentation and region-based techniques such as watershed segmentation and graph-based segmentation methods are evaluated on the bowl-feeder datasets. As illustrated in Figure 4.9, the watershed segmentation consists of performing threshold and edge-based segmentation to reduce background noise and obtain the boundaries of foreground parts. The boundary-based edge segmentation serves as location guidance for the superpixel generation. To reduce superpixel oversegmentation and background segmentation, gaussian blur and K-Means clustering are applied. To segment foreground parts based on color characteristics, the bowl-feeder datasets are converted to the HSV color space.

Figure 4.9: Method Overview: Watershed Segmentation

4.2.2 Interactive Segmentation

As objects with similar spatial characteristics can be grouped with user feedback, manual interactive and deep learning-based segmentation methods are evaluated. In this study, the GrabCut annotation tool is applied to segment foreground objects from the image background. Specifically, in applying the zoom-in feature to focus on segmenting assembly

parts with specific shape and texture characteristics[87]. Afterward, user feedback is applied to manually segment the foreground assembly parts within the bowl-feeder machine. The foreground objects are assigned to different classes than the background surface and visualized using different colored segments.

To compare the performance of the manual segmentation with deep learning-based interactive segmentation, the F-BRS segmentation model is applied. The model is evaluated on the bowl-feeder datasets. By applying the interactive segmentation model, the user guides the generation of segmentation mask at specified user-click locations.

4.2.3 Supervised Object Detection and Segmentation

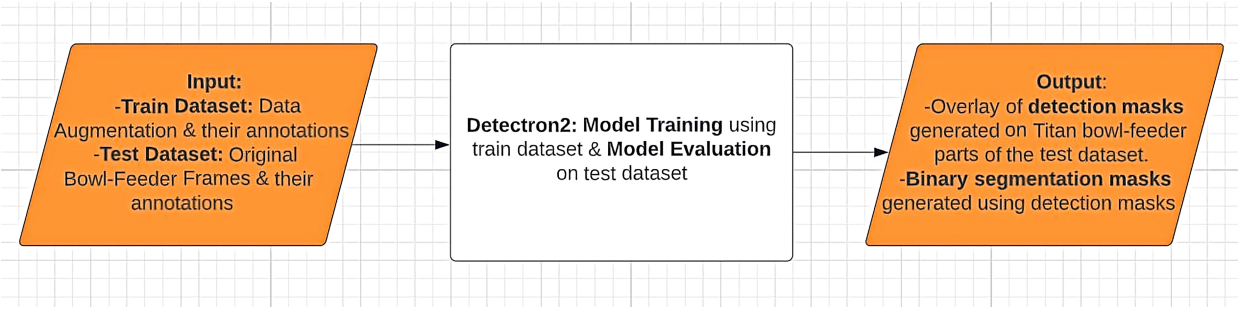


Figure 4.10: Overview of the supervised segmentation method applied on the bowl-feeder dataset

As interactive-based methods require repeated user feedback toward the generation of object segmentation masks, the manual feedback can result in decreased system efficiency and slower model deployment in real-time applications. Therefore, supervised segmentation is evaluated to generate a first-frame annotation mask and enable an automatic initialization of the ST-GNN model. To detect and segment foreground parts using the supervised approach, the Detectron2 object detection model is trained and evaluated on bowl-feeder videos consisting of variational parts-load. As the pixel intensity in foreground image objects contributes to various color and texture characteristics, the extracted object-based features can be applied for model training and evaluation in object detection, using the architecture in Figure 2.13[98].

In this study, the model is trained and evaluated using an 80:20 train & test split with 200 training images and 50 test images. The training images were split into two batches

for the application of different data augmentation techniques and for the detection of parts boundaries with varying levels of illumination, image rotation, and distorted bowl-feeder orientation. Specifically, Batch 1 consisted of an increase in brightness scale change zoom, and Batch 2 consisted of applied horizontal flip with an increase in rotation width shift. The test dataset consisted of original frames of variational parts-load such as minimally filled and dense part clusters.

4.3 Anomaly Detection

In this section, the outlined methods are evaluated to detect spatial anomalies within the synthetic conveyor belt dataset. Specifically, in analyzing features obtained from hand-crafted techniques and deep learning-based methods to detect spatial irregularities within part characteristics. The preprocessed and registered images allow accurate comparison of spatial part characteristics with normal behavior.

4.3.1 Image Preprocessing

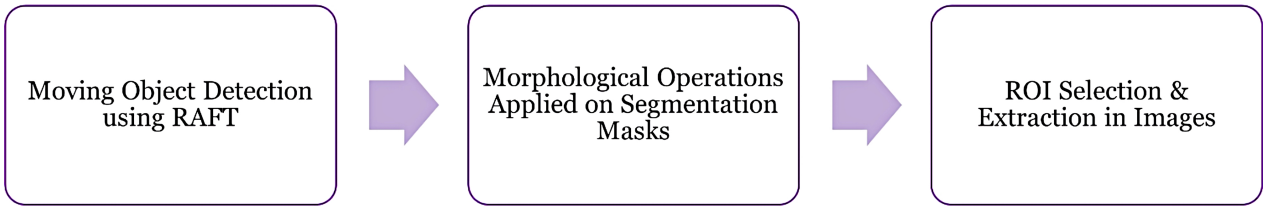


Figure 4.11: Overview of preprocessing methods applied to the flow-based segmentation masks

In order to perform anomaly detection using spatial features of moving assembly parts, image preprocessing methods must be applied to the optical flow segmentation masks. The preprocessing methods consist of performing morphological operations to reduce the noise in optical flow masks, applying region of interest (ROI) selection and extraction for background subtraction, and evaluating image registration techniques to transform extracted ROIs to the same coordinate system. As the segmentation masks generated from RAFT contain noise such as joined segmentation masks and detected motion on

background features, morphological techniques such as opening and closing operations are applied. After applying preprocessing methods to the segmentation masks generated from RAFT, the ROI selection and extraction are performed for background subtraction. Specifically, the fine-tuned mask is used to find object contours, and the coordinates of the largest contour area are set to the defined coordinates of the inner rectangle. By doing so, the adjusted coordinates of the contour are used to draw a rectangle on the ROI without including the additional background of the image. Each ROI is then extracted and saved externally to perform image registration techniques.

4.3.2 Image Registration

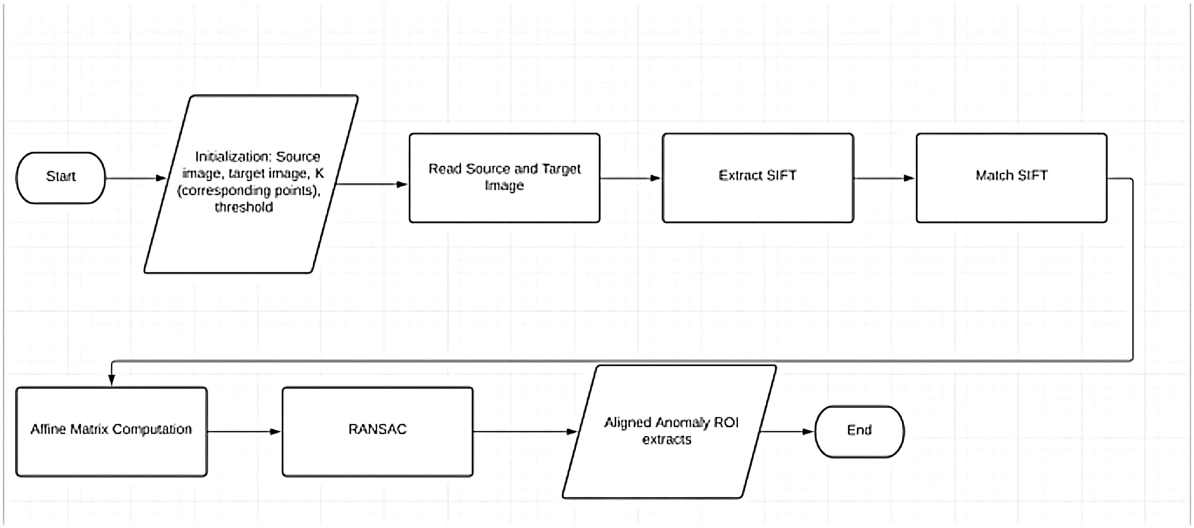


Figure 4.12: Process overview of affine-based transform applied with RANSAC to ensure robust image registration

To detect and compare spatial features of extracted ROIs to normal behavior characteristics, the captured ROIs from multiple viewpoints must be transformed into the same coordinate system. Specifically, the points are mapped from one extracted ROI to corresponding points of a reference ROI within a coordinate system. By registering viewpoints into the same coordinate system and aligning two images, handcrafted features such as texture and color can be compared to indicate spatial anomalies such as a missing object

or changes in part-type orientation. As the linear conveyance dataset consists of lower image resolution and small dimensions of extracted ROIs, keypoint detection results in errors and an empty feature descriptor. Therefore, the synthetic dataset is applied to demonstrate image registration and handcrafted methods on registered images for the detection of spatial anomalies.

In this study, the objects on the conveyor belt are captured from the side top view and top view, in which image registration is applied to transform the two images into the same coordinate system. As illustrated in Figure 4.12, the image registration process is initialized with the inputs: source image, target image, the minimum number of corresponding points to estimate the affine matrix, and threshold value to define outliers in RANSAC. The source image serves as a reference to map the target image coordinates to its corresponding points within the reference coordinate system. Afterward, the Scale Invariant Feature Transform (SIFT) features are extracted and matched between the two images. Specifically, the SIFT descriptors of the input source and target images are matched. To choose the best matches of descriptors, the L2 distance metric is applied to filter out descriptors closest to each other.A match ratio is applied to filter out the best matches of descriptors. To ensure robust feature matching, the thresholding value is applied to remove outliers. The corresponding points are applied as an input in the RANSAC algorithm to estimate the affine transformation matrix between the two images. The image registration will be applied on the synthetic conveyor belt dataset. Additional image registration techniques were applied with varying object texture; however, variation in pixel intensity and lower image resolution can result in image registration errors[71].

4.3.3 Hand-Crafted Features

In order to compare the spatial features between aligned images, handcrafted methods such as color and texture descriptors are applied to detect spatial anomalies such as changes in part orientation and part type, as shown in Figure 4.13. In this study, the Histogram of Oriented Gradients (HOG), and the color descriptor are applied to compare shape and color features between two images, respectively. The HOG descriptor measures the frequency of gradient orientation within a localized region of an image. To compute the shape features, the HOG descriptor applies the magnitude and angle to generate a histogram of oriented gradients. The HOG descriptors generated from two images can be compared using Euclidean distance. This is significant as an increased Euclidean distance between descriptors indicates the presence of spatial anomalies such as missing objects and changes in part orientation. Additionally, color descriptors were also applied to detect spatial anomalies such as the absence of parts. Specifically, as the color descriptor calculates the histogram of

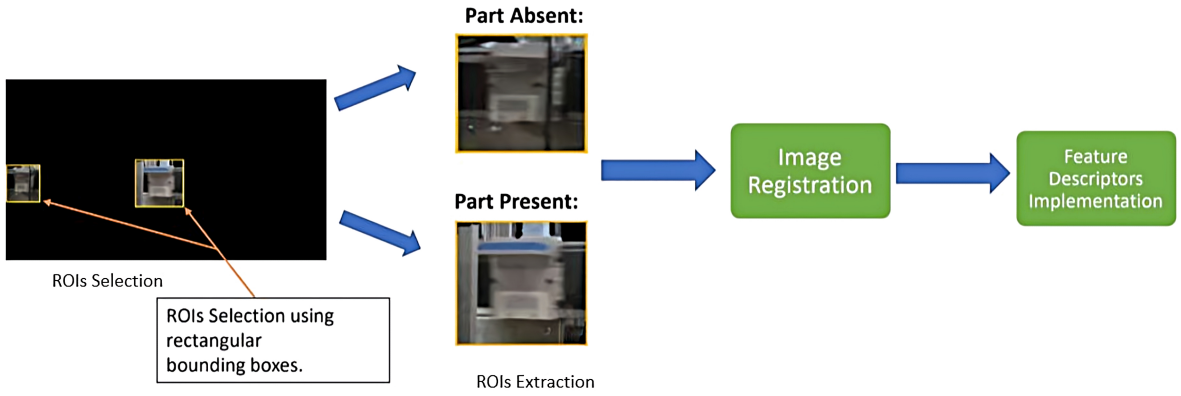


Figure 4.13: Process overview of the hand-crafted methods applied on the post-processed Region of Interests (ROI) extracts

the red, blue, and green channels, the absence of a blue assembly part within the assembly cell would be indicated by a decreased count of the blue channel.

4.3.4 PatchCore Anomaly Detection

To compare the performance of handcrafted methods with deep learning-based anomaly detection methods, the PatchCore anomaly detection model is applied to the synthetic linear conveyance dataset. The PatchCore anomaly detection model consists of a pre-trained network on ImageNet classes[75]. In order to detect anomalies using the deep learning-based method, the model is trained and evaluated on normal and anomalous classes of the conveyor dataset, respectively. The normal class consisted of one object on the conveyor belt, and the anomalous class consisted of variations such as the object color and multiple objects moving on the conveyor belt. To perform model training and evaluation, the model was trained on 300 nominal images and evaluated on 114 images and ground-truth annotations of anomaly variations with 20 images of normal class. The moving objects for the training and testing images with ground-truth annotation masks were detected using the RAFT optical flow. Morphological operations and ROI extraction are applied as preprocessing steps to segment flow-based segmentation masks.

During model training, the nominal samples are broken into neighborhood-aware patch

level features as input into the memory bank, as illustrated in Figure 4.14. To reduce inference time, the neighborhood patches within the memory bank are down-sampled using greedy coreset sampling. In model evaluation, the anomalies within the conveyor dataset are classified if a minimum of one patch is anomalous. After detection of anomalies during the model evaluation, a pixel-level anomaly segmentation map for anomaly localization is

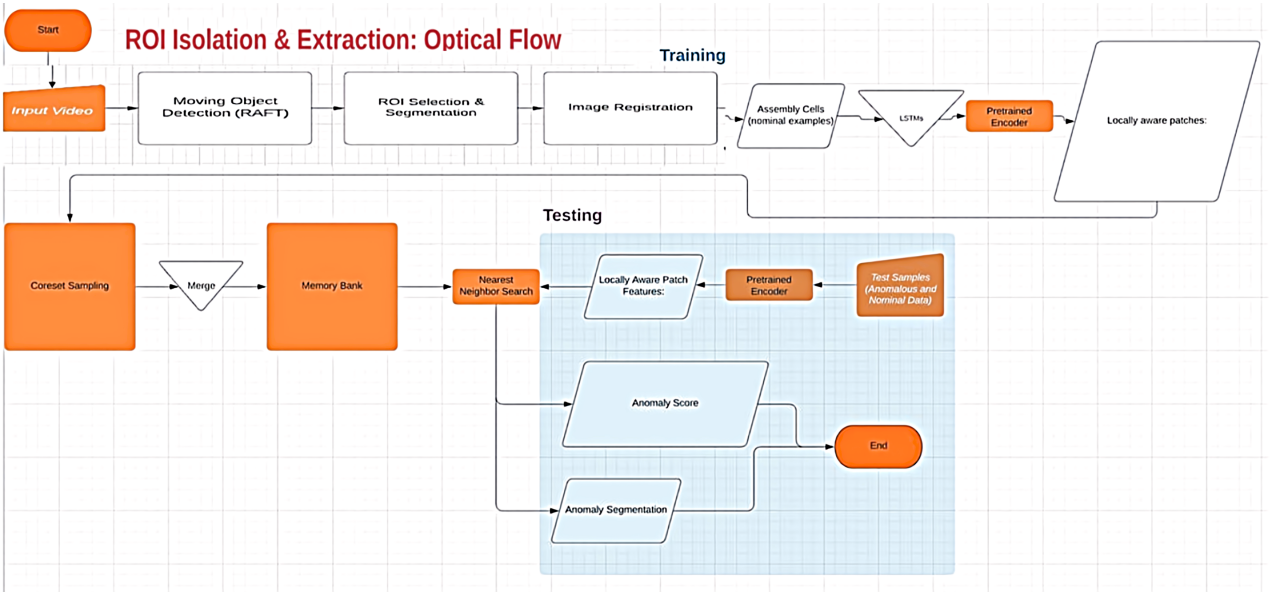


Figure 4.14: PatchCore deep learning-based anomaly detection method applied on synthetic conveyance dataset

4.4 Chapter Summary

This section outlined the methods evaluated to perform motion-based and appearance-based segmentation of foreground assembly parts within the various automated manufacturing machines. Specifically, in exploring traditional and deep learning-based methods to increase accuracy of moving object detection. The pixel-based tracker and the ST-GNN models are compared to evaluate segmentation accuracy of foreground objects. To enable automatic initialization of ST-GNN model, appearance-based segmentation methods are explored. Specifically, pixel-based, interactive-based and supervised segmentation methods to accurately detect foreground parts based on spatial characteristics. Additionally, the preprocessing methods are outlined, which reduce limitations in visual characteristics. The

handcrafted and deep learning-based methods are outlined to perform anomaly detection on processed datasets.

Chapter 5

Experiments & Results

This section outlines the qualitative and quantitative results generated from the evaluation of motion-based and appearance-based segmentation methods. To segment foreground assembly parts based on shared temporal characteristics, motion-based segmentation methods such as moving object detection, object tracking and temporal segmentation were performed. To segment foreground assembly parts based on their shared spatial characteristics, the appearance-based methods of pixel-based, interactive-based and supervised-segmentation were performed. The spatial features consist of parts texture, color and shape characteristics.

By using pixel characteristics, user-based feedback and object classes, the appearance-based methods accurately segmented foreground parts. Additionally, this section illustrates and compares spatial anomaly detection results generated using handcrafted methods and deep learning model. In this study, motion-based and appearance-based segmentation were performed and compared on various instances of bowl-feeder machine. Different applications of bowl-feeder consisting of variational parts-load and illumination variance were applied due to data availability at the time of experiment with respective methods. Methods evaluation on such dataset characteristics improved method robustness and accuracy of results.

5.1 Illumination Invariance

The bowl-feeder machine consists of foreground moving parts, as illustrated using manual annotation in Figure 5.1. As the datasets consisted of limited visual characteristics such as

illumination variance, color-space conversion and illumination robust background subtraction techniques were applied to the bowl-feeder image dataset.The objective was to reduce limited visual characteristics by applying color-space conversion and background subtraction techniques. The experimental setup consisted of initially applying color-space conversion and then performing background subtraction techniques on the converted images to assess illumination robust moving object detection. Based on the results in Figure 5.2, the illumination robust Local SVD Binary Pattern (LSBP) background subtraction method showed increased moving parts detection than Mixture of Gaussian (MOG) method. This is significant as the LSBP method compares the local pixel values to calculate structure modelling of local image regions. In comparison to background subtraction MOG, LSBP consists of increased robustness to cast shadows and illumination variance.

5.2 Motion-based Features

In this section, the motion-based segmentation was performed by evaluation of moving object detection, pixel-based tracking and spatial-temporal segmentation methods. By performing motion-based segmentation, the foreground moving objects would be segmented based on shared motion characteristics.

5.2.1 Moving Object Detection

To perform motion-based segmentation of foreground assembly parts, moving object detection was applied to detect parts based on shared temporal characteristics. Motion characteristics such as synchronous or asynchronous parts regulation within assembly machines. Therefore, techniques such as local Lucas-Kanade and global Gunnar-Farneback optical flow methods were evaluated to detect and segment moving parts. In presence of limited visual characteristics, the traditional local method would reduce noise in object detection using sparse feature detection and extraction. The experimental setup consisted of performing independent evaluation of local and global method on the bowl-feeder dataset. Based on the results in Figure 5.3, the local method detected increased number of objects within outer-shelves of the bowl-feeder and simultaneously visualized the flow direction of detected objects. In comparison to the local optical flow, the global gunnar-farneback method showed increased number of object detection within the bowl-feeder machine. This is significant as the global method computed optical flow using dense feature detection and extraction. Therefore, the results in Figure 5.3 and 5.4 illustrated increased accuracy in object detection using the global method in comparison to the local method.

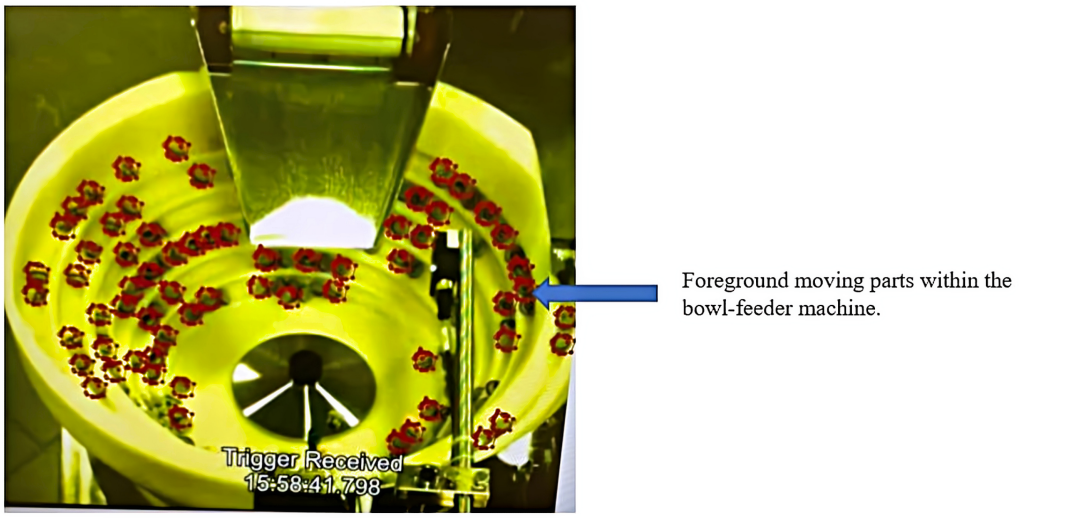


Figure 5.1: Manual annotation of all foreground parts within the bowl-feeder machine

In order to increase detection accuracy in presence of limited visual characteristics such as illumination variance and occlusion, the deep learning-based RAFT method was applied to compute optical flow. The RAFT optical flow performs multi-scale feature extraction to detect moving objects. The method was applied on various manufacturing machines datasets such as the digital assembly line, linear conveyance system and vibratory bowl-feeder machine. In comparison to the traditional methods, RAFT illustrated increased object detection in presence of occlusion and illumination variation. In Figure 5.5, the segmentation masks generated using RAFT showed decreased mask IoU and increased alignment with parts boundary. This is significant as the RAFT feature network attenuates noise in detection by performing multi-scale feature detection and extraction. The 2D flow field in section c of Figure 5.5 illustrates the flow magnitude using variation in vector length and parts movement in counter-clockwise direction.

The RAFT method was applied to conduct separate detection analysis studies, in which the moving and stationary foreground objects were detected, respectively. Inverse binary segmentation masks were generated to detect and segment stationary objects. By detection of stationary objects, the generated segmentation masks would consist of uniform shape and size as most accumulated parts were located within the base shelf of the bowl-feeder. Therefore, decreased preprocessing steps such as morphological operations and ROI ex-

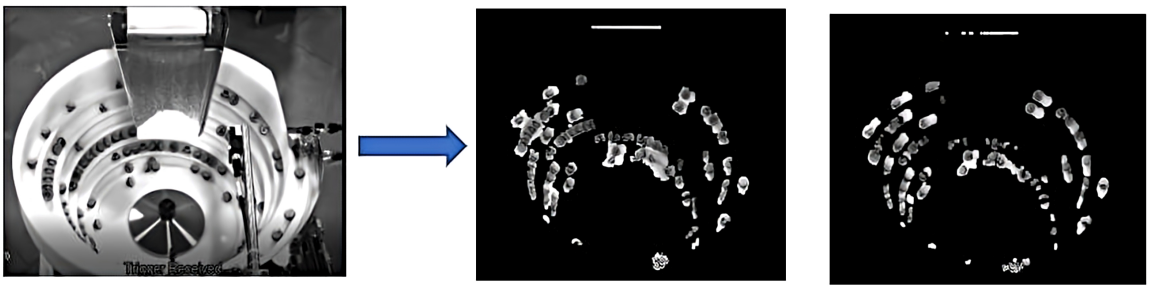


Figure 5.2: Application of color-space conversion and background subtraction techniques for robustness to illumination variation, noise and shadows

traction would be required to perform on the segmentation masks. Based on the results in Figure 5.6, RAFT accurately detected and segmented moving objects within bowl-feeder in comparison to detection of stationary objects.

5.2.2 Moving Object Pixel Tracking

As RAFT accurately detected the foreground moving objects within the automated machines, its iteratively updated flow field is applied to create a pixel-based tracker. The objective of the pixel-based tracker was to differentiate previous and current moving parts in different colored segments and simulatenously visualizes the flow field to illustrate flow magnitude and direction. Based on the results in Figure 5.7, the tracker accurately segmented the current and previously moving parts in the bowl-feeder and the digital assembly datasets. For instance, the tracker accurately visualized the previous moved parts within the base shelf in blue-colored segments. As the parts within the outer-shelf moved more frequently than accumulated parts within base-shelf, the tracker accurately visualized them in white-colored segments. Additionally, the evaluation of pixel-based tracker on the digital assembly dataset illustrated segmentation of current and previously moved parts using similar visual characteristics.

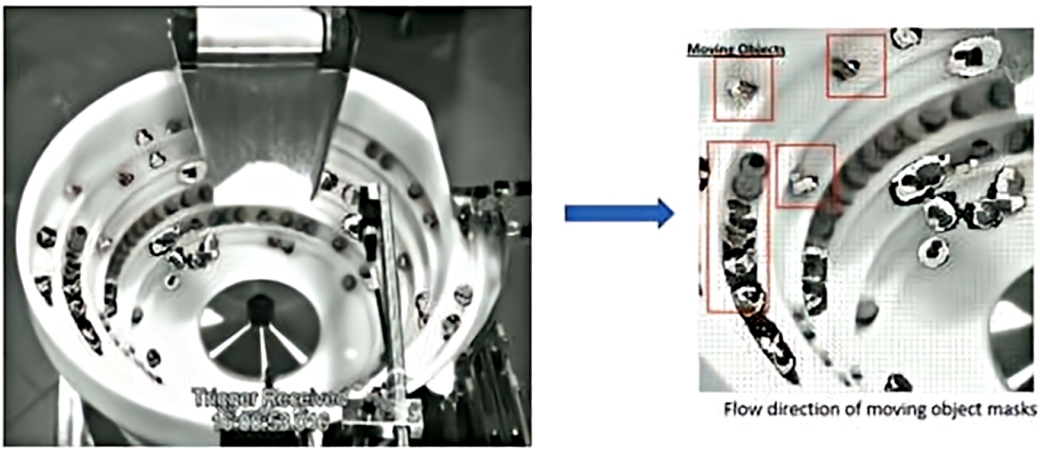


Figure 5.3: Lucas Kanade optical flow applied to detect foreground moving parts in real-time. Detected parts segmented in grayscale masks

5.2.3 Spatio-Temporal Segmentation

Although the pixel-based tracker accurately segments the moving objects, the generated segmentation masks consisted of increased boundary IoU, which resulted in the segmentation of foreground moving parts and background surface of machines. The different colored segments would increase the preprocessing steps to select and extract the region of interests. To increase accuracy in segmentation boundary mask IoU with parts boundary, the space-time graph neural network (ST-GNN) was evaluated on the bowl-feeder and digital assembly datasets. The experimental setup consisted of evaluating the ST-GNN model on the vibratory bowl-feeder, linear conveyance and digital assembly dataset. Based on the results in Figure 5.9, the ST-GNN model accurately generated a segmentation mask in alignment with the parts boundary of rotational indexer and the robotic manipulator.

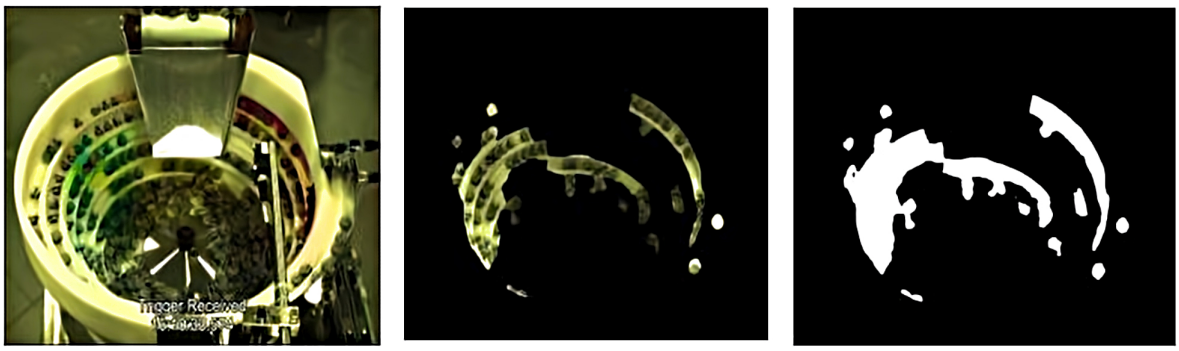


Figure 5.4: Gunnar Farneback optical flow method applied to detect moving foregrounds parts in real-time. Flow direction of detected parts visualized using color variation of segmentation masks

5.3 Appearance-based Features

As the ST-GNN model was applied to proceed with motion-based segmentation, the appearance-based segmentation techniques were evaluated to generate the first-frame annotation mask. The generated first-frame annotation mask would enable automatic initialization of the ST-GNN model.To perform appearance-based segmentation, pixel-based, interactive-based and supervised segmentation methods were evaluated on the bowl-feeder image dataset.

5.3.1 Interactive Segmentation

To group and segment similar foreground objects based on the user-based feedback, the F-BRS deep learning-based interactive segmentation and GrabCut manual segmentation methods were evaluated on the bowl-feeder datasets. Figure 5.10 illustrates the application of appearance-based segmentation methods, which applied user-based feedback to segment foreground assembly objects. In specific, the manual segmentation using GrabCut annotation tool and the F-BRS interactive deep learning-based model. Based on the methods evaluation in Figure 5.10, the results illustrated increased segmentation accuracy

using manual segmentation in comparison to the segmentation performance of F-BRS deep learning-based model.

5.3.2 Pixel-based Segmentation

Although the manual interactive-based method illustrated increased accuracy in segmentation of the foreground objects, the method required repeated user-based feedback thus decreasing method efficiency. To group object based on spatial characteristics such as texture, shape and color, pixel-based segmentation methods were evaluated. The experimental setup consisted of independent evalution of various pixel-based segmentation techniques on the bowl-feeder image dataset.Based on the results in Figure 5.12, the region-merging segmentation techniques of Watershed segmentation accurately segmented foreground parts than the texture-based segmentation methods, as illustrated in Figure 5.13. With Gaussian blur and K-means clustering applied, the Watershed segmentation accurately separated the foreground objects without superpixels oversegmentation. Although the texture segmentation using GLCM and Gabor filters detected foreground parts, additional preprocessing steps would be required to extract ROIs of irregular shapes and size. The entropy-based segmentation illustrated decreased accuracy in foreground segmentation as background parts with similar texture characteristics were also segmented. In Figure 5.14, the clustering-based Meanshift method accurately segmented dense cluster parts in comparison to the Graph-based segmentation method applied. To conclude, the region-merging Watershed segmentation and clustering-based segmentation accurately segmented the foreground object shapes than the texture-based segmentation methods.

The pixel-based segmentation accurately segmented the foreground parts within the bowl-feeder than interactive-based segmentation method. Due to the pixel intensity variation and limited visual characteristics, the pixel-based segmentation resulted in decreased accuracy.

5.3.3 Supervised Object Detection and Segmentation

To increase accuracy in segmentation of foreground parts, the supervised segmentation was applied using the Detectron2 detection model. The deep learning model was trained on augmented datasets of the bowl-feeder machine. The model evaluation was performed on the original bowl-feeder datasets, which consisted of variational parts load. Based on the results in Figures 5.17 and 5.18, the Detectron2 model accurately segmented all of the foreground parts in presence of limited visual characteristics. Model generalization

was performed to evaluate detection accuracy on various other instances of the bowl-feeder machine. Figure 5.17 shows missed detection of assembly parts within the various bowl-feeder machines. To conclude, the Detectron2 model evaluation illustrated increased detection accuracy in original bowl-feeder dataset than in evaluation on other bowl-feeder instances.

5.4 Anomaly Detection

In order to perform anomaly detection in spatial part characteristics, handcrafted methods and deep learning-based models were evaluated to detect spatial irregularities within the part characteristics. The preprocessed and registered images were applied to increase accuracy in feature extraction and comparison with normal behavior part characteristics.

5.4.1 Image Preprocessing

To detect spatial anomalies within moving assembly parts, the RAFT optical flow method was primarily applied to detect moving parts and generate the parts segmentation masks. Therefore, the experimental setup for image preprocessing consisted of moving object detection using RAFT, morphological operations applied on segmentation masks and ROI extraction. In Figure 5.19, the preprocessing methods such as morphological operations and ROI selection were performed to attenuate noise and extract the ROIs from the background surface. Based on the evaluation in Figure 5.19, ROIs of uniform shape and size were extracted for detection of spatial anomalies.

5.4.2 Image Registration

As the assembly parts were captured from multiple camera viewpoints, affine-based image registration was applied to transform images into one coordinate system. The image registration is significant for comparison of spatial characteristics between aligned images. The experimental setup consisted of extracting and matching SIFT features between the read source and target images. The closest matches of descriptors between the two images and correspondence selection were applied to estimate the affine transformation matrix for image registration. Based on the results in section c of Figure 5.19, the input target image with top view was completely aligned to the side top viewpoint of the input reference image. The higher image resolution in synthetic manufacturing dataset was significant for robust image registration.

5.4.3 Hand-Crafted Features

To compare the spatial features of two extracted ROIs for anomaly detection, the experimental setup consisted of extracting spatial handcrafted features such as color and texture on the various manufacturing datasets. Specifically, in Figure 5.19 and 5.20, the handcrafted method and deep learning-based methods were applied to detect spatial anomalies such as missing parts, changes in part-type orientation and color. Based on the results in Figure 5.19, the handcrafted methods such as texture and color descriptors illustrated detection of missing object. In specific, the indication of missing objects through the changes in texture gradient orientations of the HOG texture descriptor. The variation within the RGB values of the color histogram also indicated the absence of blue-rectangular object within the assembly cell. Specifically, the increase in the value of the red color channel of the color histogram, as illustrated in section b of Figure 5.19.

5.4.4 PatchCore Anomaly Detection

To compare the anomaly detection performance of the handcrafted methods, the deep learning-based PatchCore anomaly detection method was evaluated on the synthetic dataset of the conveyance system. As the original surveillance videos of the conveyance system consisted of lower image resolution, a synthetic dataset was generated to address limited visual characteristics and reduce image registration errors. The synthetic dataset consisted of spatial anomalies such as the presence of multiple parts, changes in part-type orientation and color. Based on the results in Figure 5.20, the PatchCore anomaly detection accurately detected and localized spatial anomalies within the moving parts of conveyor system. The method generated segmentation map to localize and illustrate spatial anomalies. The segementation map accurately localized the spatial anomalies consisting of multiple parts present and color change of part.

5.5 Chapter Summary

This section presented the visualizations and quantitative results generated using the motion-based and appearance-based segmentation methods. The evaluation of traditional and deep learning-based motion segmentation techniques increased accuracy within moving object detection, object tracking and temporal segmentation. Based on evaluation of moving object detection techniques, the deep learning-based optical flow outperformed the traditional local and global methods in presence of limited visual characteristics. In

comparison to the pixel-based tracker, the results illustrated increased accuracy in parts segmentation using the ST-GNN model. To enable automatic initialization of the ST-GNN model, the supervised appearance segmentation method outperformed the interactive-based and pixel-based segmentation methods. The evaluation of handcrafted methods and deep learning-based model accurately detected the spatial anomalies within the manufacturing datasets. To conclude, various motion-based and appearance-based methods were evaluated to increase accuracy in temporal and spatial parts segmentation.

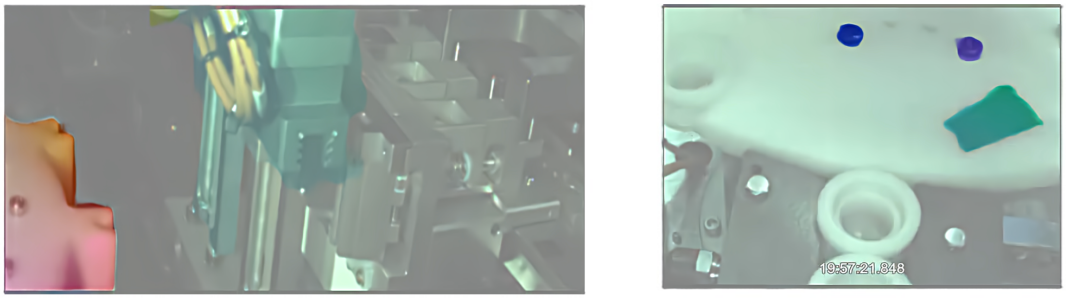


Figure Section (a)

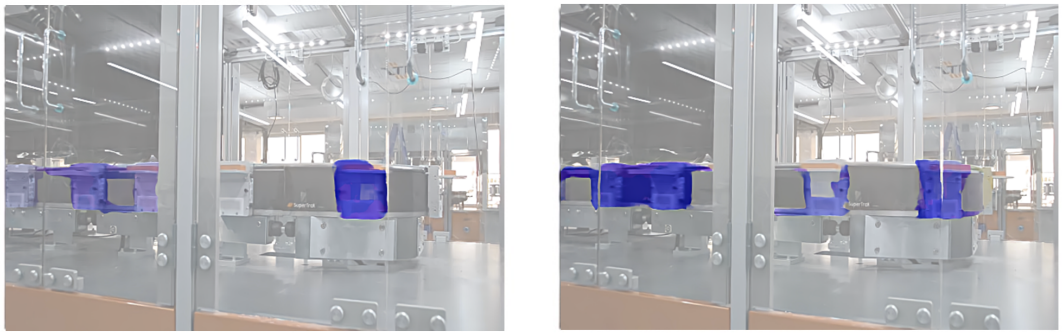


Figure Section (b)

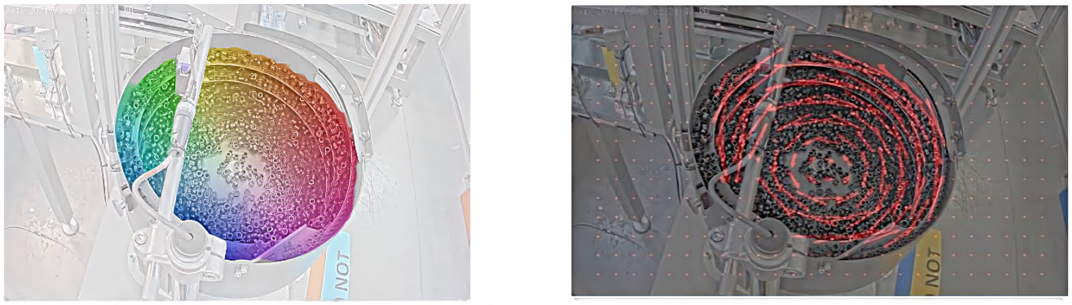


Figure Section (c)

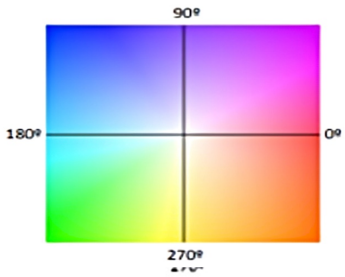


Figure Section (d)

Figure 5.5: Deep learning-based RAFT optical flow applied: digital assembly (a),conveyance system (b), and vibratory bowl-feeder(c). Flow direction and magnitude of parts illustrated using varied colors and color intensity, as shown in color wheel (d)

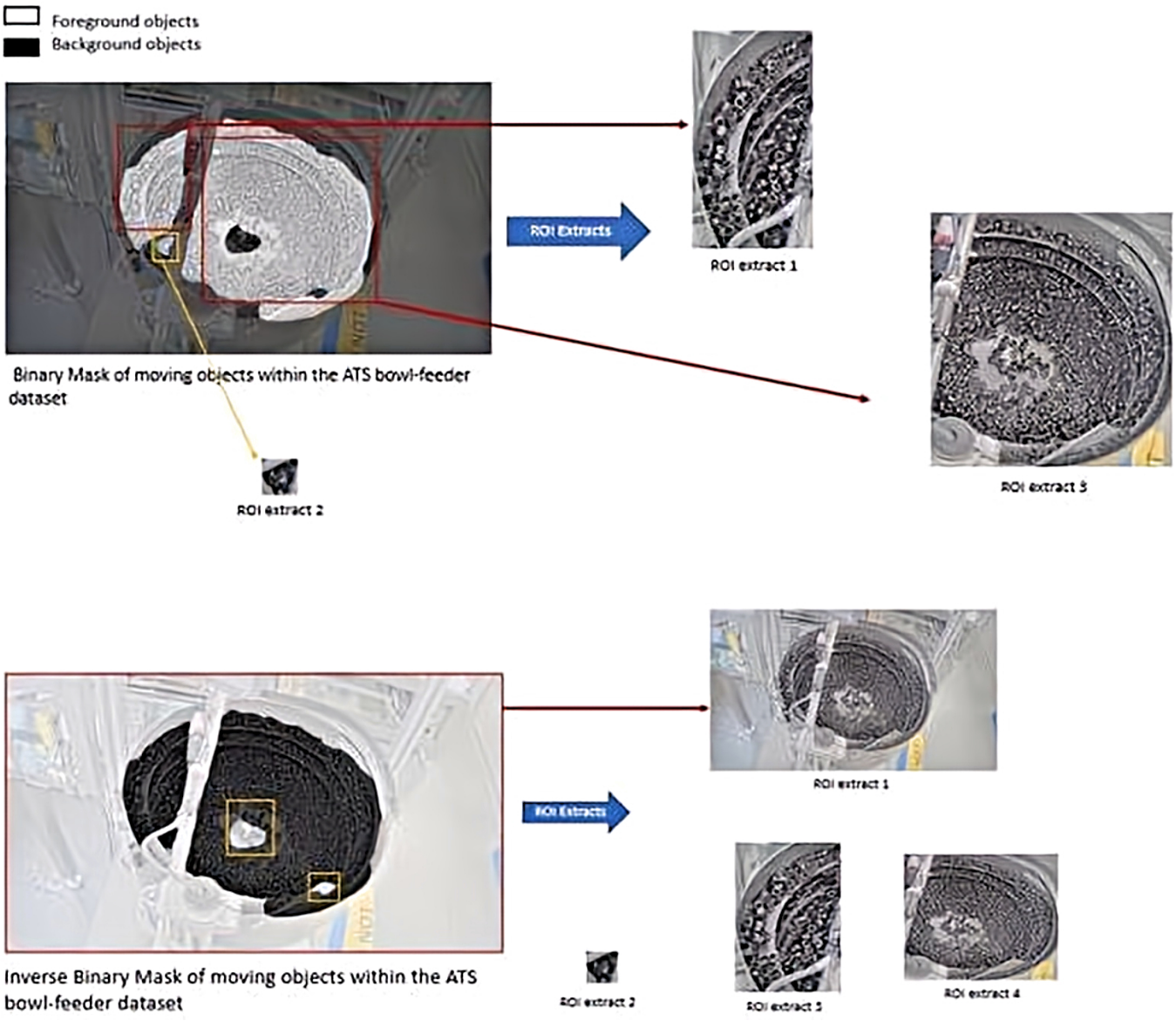


Figure 5.6: Comparison of stationary and moving parts detection using RAFT optical flow method. Illustration of ROIs generated from flow-based detection

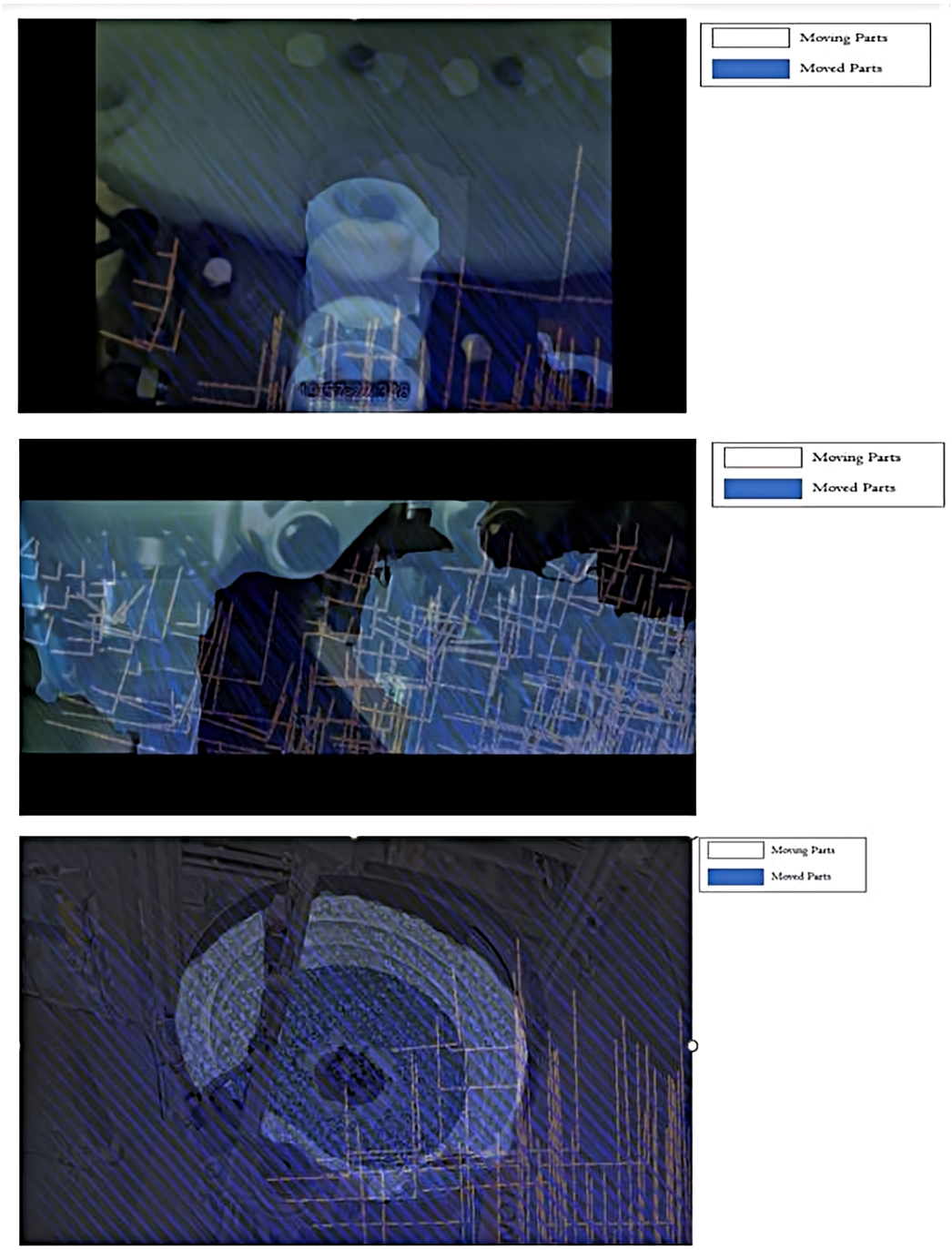


Figure 5.7: Temporal segmentation of current and previously moved objects using optical flow-based tracker

	Method:	Parts Present	Parts Detected
Moving Object Detection	Lucas-Kanade	77	38
	Gunnar-Farneback	216	61
	RAFT	476	398
Object Tracker	Pixel-based Tracker	476	445

Figure 5.8: Evaluation of motion-based segmentation methods for moving part detection and segmentation

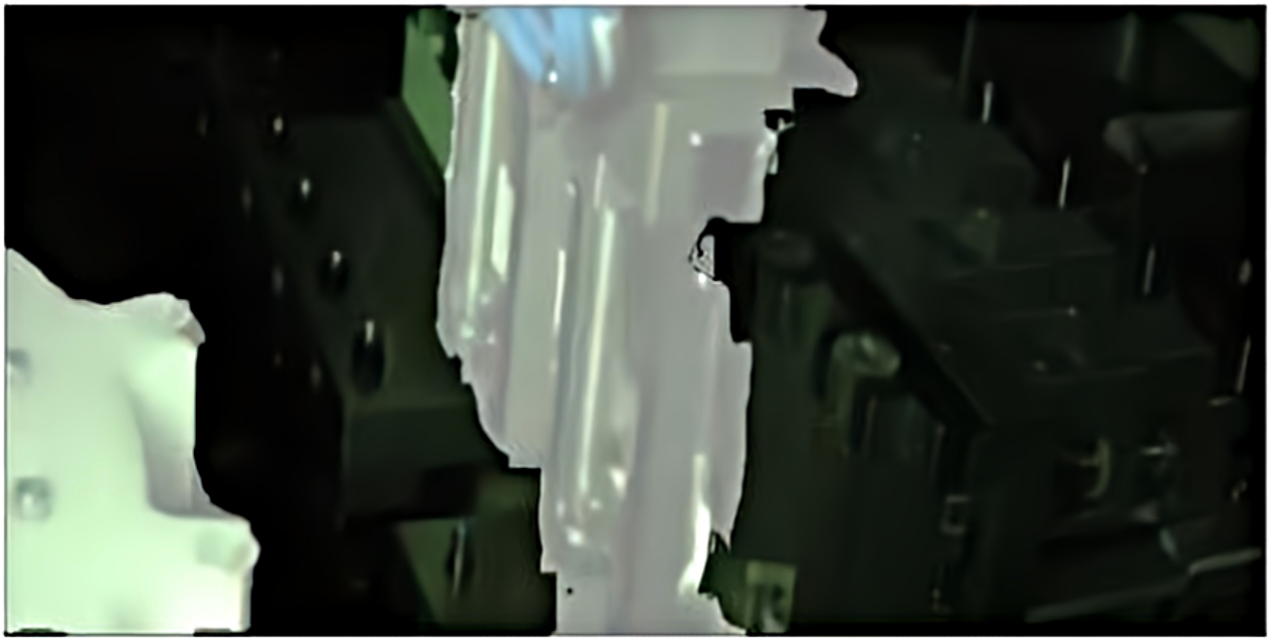
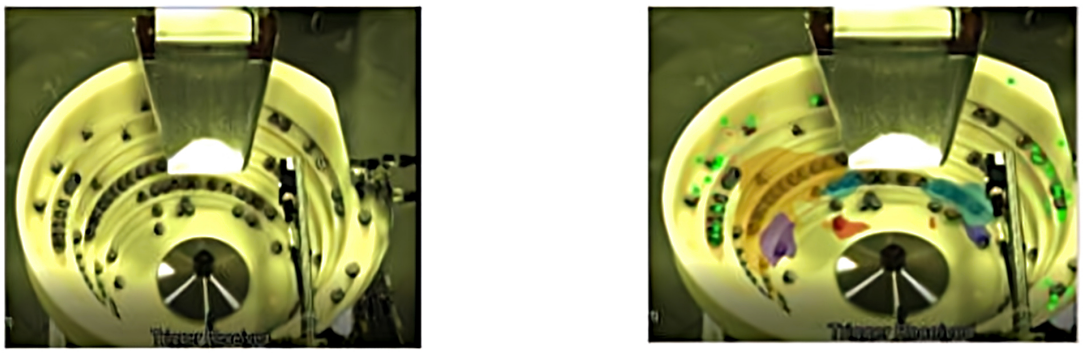


Figure 5.9: Temporal segmentation of foreground moving assembly parts with Space-Time Graph Neural Network (ST-GNN) model



Input Image: Bowl-feeder with dense part clusters

Segmentation mask for moving objects

Figure Section (a)

Input Image: Bowl-feeder with dense part clusters

Segmentation mask for moving objects

Figure Section (b)

Figure 5.10: Comparison of manual and deep learning-based interactive segmentation methods: GrabCut Manual Annotation Tool(b) and f-BRS segmentation model(a)

	Method:	Parts Present	Parts Detected
Interactive-based Segmentation	Manual Annotation: GrabCut Segmentation	216	216
	F-BRS Segmentation	70	23

Figure 5.11: Evaluation of interactive-based segmentation methods for part detection and segmentation based on spatial part characteristics

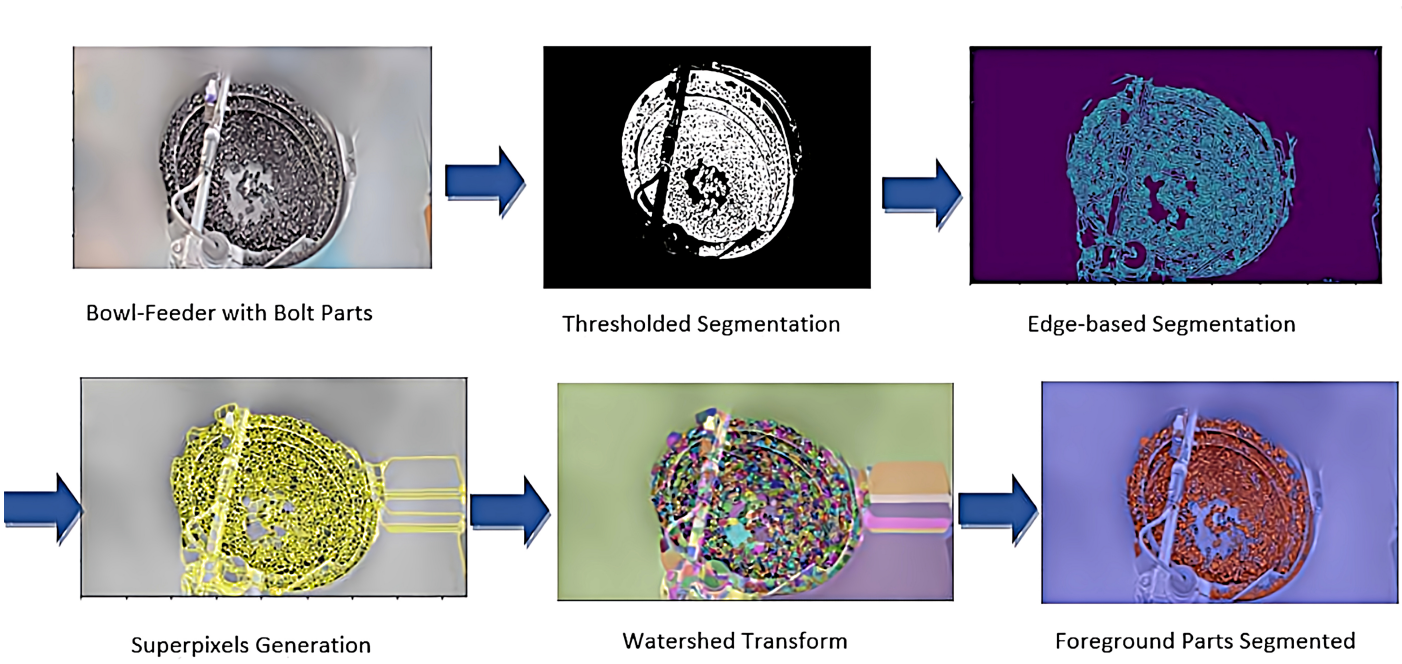


Figure 5.12: Region-merging Watershed segmentation method applied to detect and segment spatial part characteristics

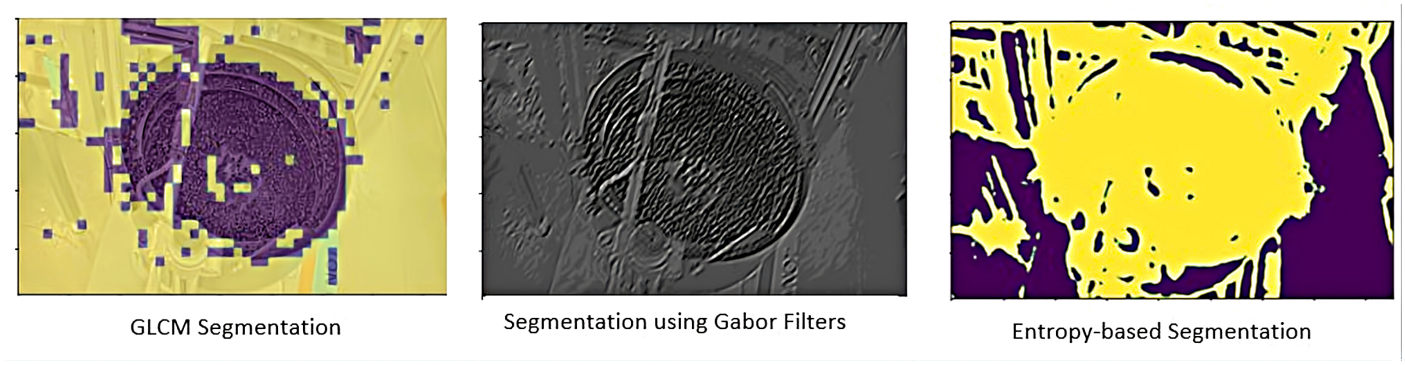


Figure 5.13: Texture-based methods: GLCM, Gabor filter and Entropy-based segmentation applied to perform foreground parts segmentation based on texture-analysis

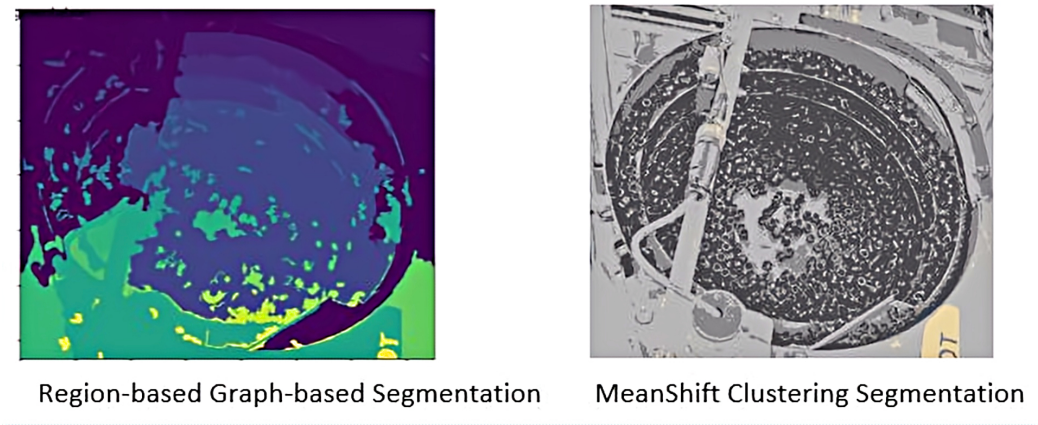


Figure 5.14: Comparison of region-based Graph segmentation (left) and clustering-based Meanshift segmentation (right) to perform detection and segmentation of bowl-feeder parts.

Figure 5.15: Color-based segmentation of foreground parts using the Value (luminance) channel of HSV color-space

	Method:	Parts Present	Parts Detected
Texture-based Segmentation	GLCM	476	467
	Gabor Filter	476	476
	Entropy-based	476	0
Shape-based Segmentation	MeanShift	476	476
	Graph-based	476	0
Color-based Segmentation	HSV Color-Space Conversion	77	71

Figure 5.16: Evaluation of pixel-based segmentation methods for part detection and segmentation based on spatial characteristics

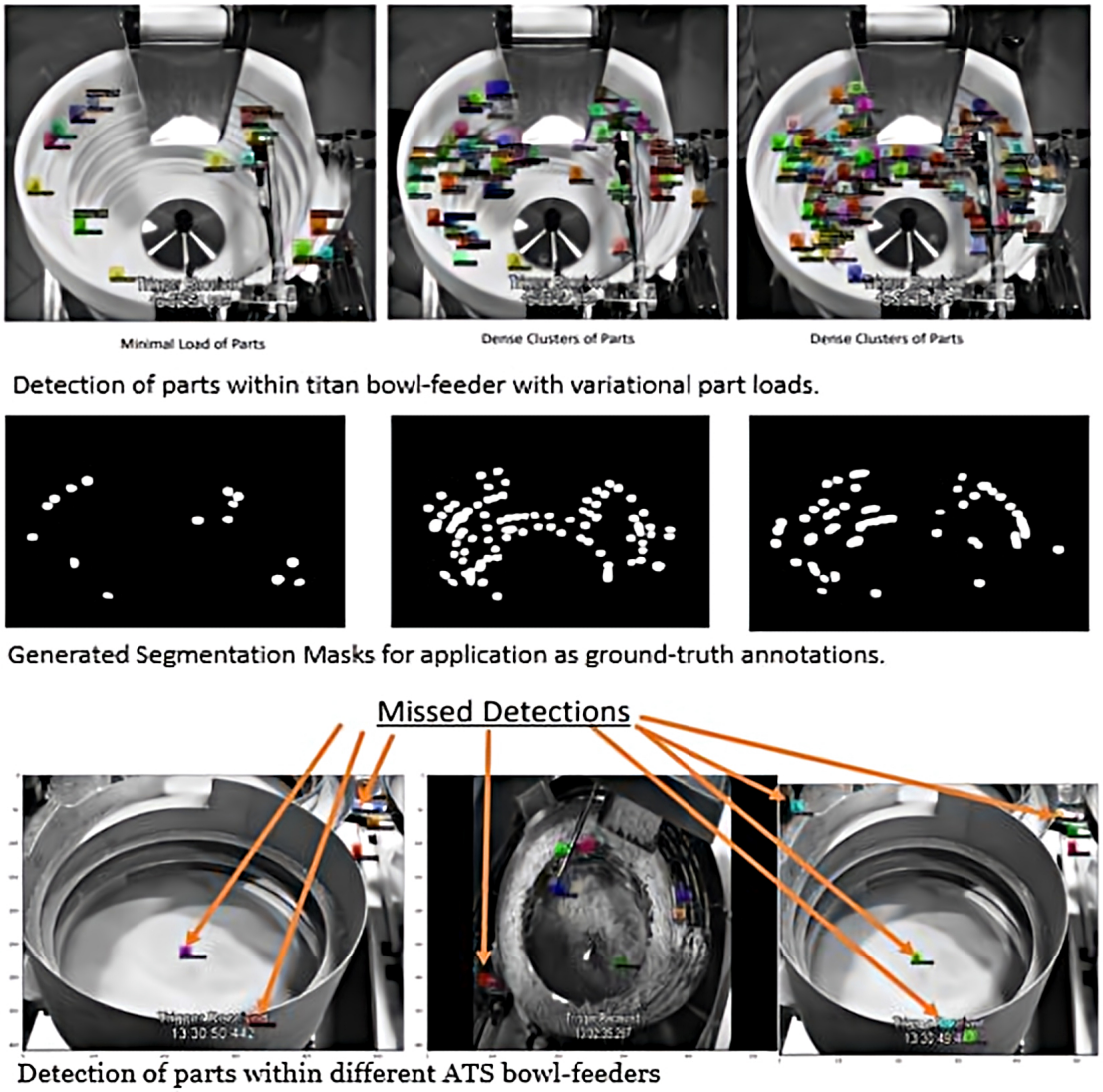


Figure 5.17: Application of supervised segmentation method using Detectron2 to detect and segment spatial part characteristics

	Method:	Parts Present (Minimal-filled, Dense Part Clusters)	Parts Detected
Supervised-based Segmentation	Detectron2	15	15
		54	54
		89	89

Figure 5.18: Evaluation of supervised-based segmentation methods for part detection and segmentation based on object classification

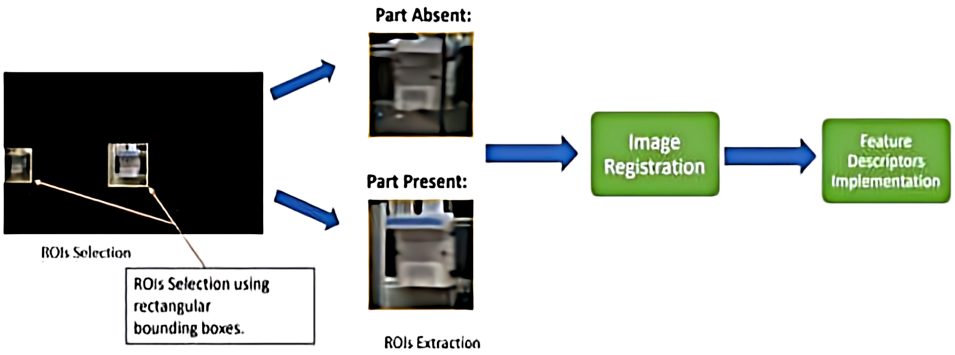


Figure Section (a)

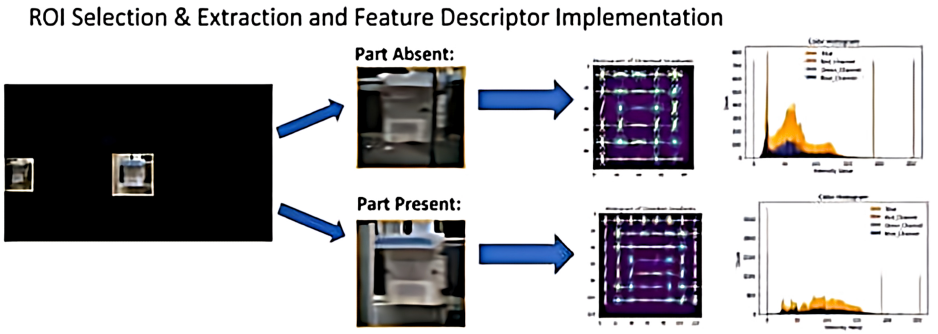


Figure Section (b)

Figure Section (c)

Figure 5.19: Application of preprocessing methods applied to optical flow-based segmentation masks and image registration to align images, captured from multiple view points, to one coordinate system(a) and (c). Handcrafted methods applied to detect spatial anomalies in registered images(b).

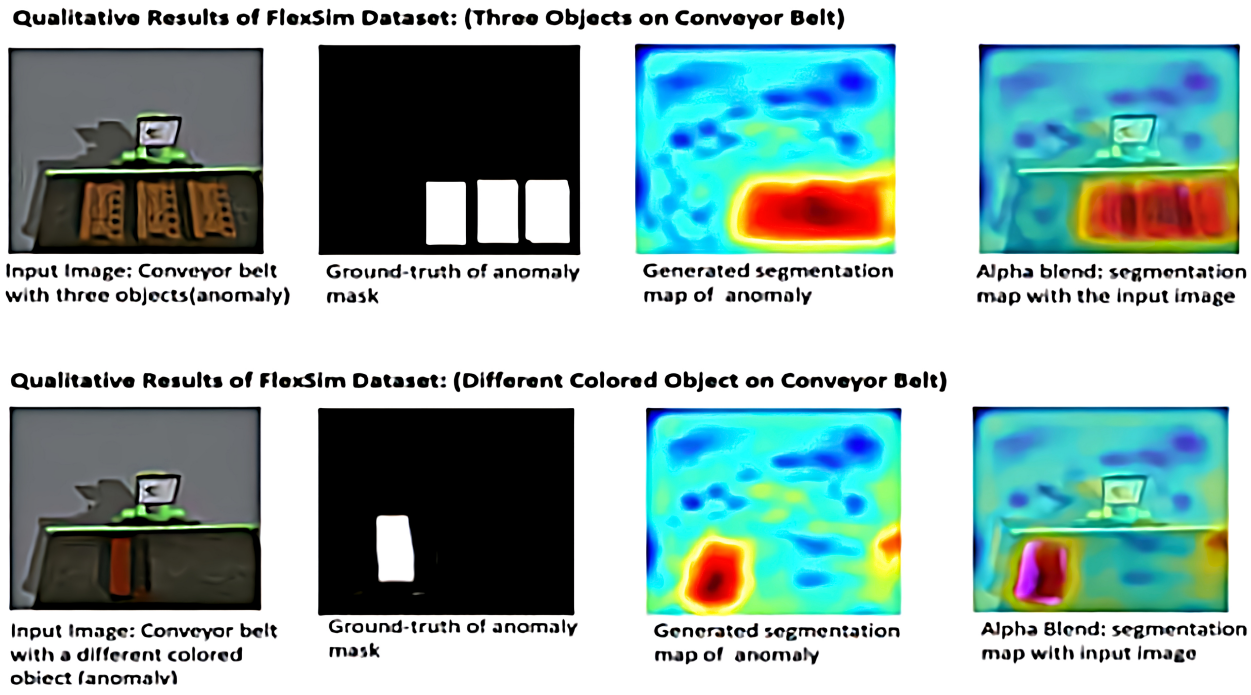


Figure 5.20: Application of PatchCore deep learning-based anomaly detection method to detect the spatial anomalies. Results of spatial anomaly detection compared with hand-crafted features.

Chapter 6

Discussion & Conclusion

In this section, an analysis is performed to assess the results generated from the evaluation of appearance-based and motion-based segmentation methods. Additionally, the evaluation of preprocessing methods is discussed to improve spatial anomaly detection within manufacturing datasets.

The surveillance videos of various automated assembly systems consist of limited visual characteristics such as illumination variance, occlusion, and lower image resolution. As the limitations in visual characteristics can result in pixel intensity variation, the intensity changes cause errors in moving object detection and image registration. To address these limited visual characteristics, methods such as color-space segmentation and synthetic dataset creation are applied to illustrate illumination invariance and higher image resolution. The color-space conversion is applied to transform the dataset into the Hue, Saturation, and Value (HSV) color space, as illustrated in Figure 5.15 of Experiments Results section. Color-space conversion is applied to bowl-feeder image datasets such as grayscale and HSV color space. The converted images are applied to perform the background subtraction methods and assess the accuracy of moving object detection. The illumination invariance is significant towards an increase in parts detection accuracy in bowl-feeder regions with varying illumination.

Based on the background subtraction evaluation, the illumination robust LSBP background subtraction method resulted in a higher number of detections. As illustrated in Figure 5.2, the LSBP background subtraction shows increased moving parts detected than MOG method. To address the lower image resolution, the synthetic dataset of the linear conveyance system is created with a higher image resolution. By the creation of a synthetic dataset with various anomaly instances consisting of multiple camera viewpoints,

image registration is successfully applied to transform various images into one coordinate system. To address the limited dataset challenge within manufacturing applications, data augmentation methods are applied to train and evaluate models with supervised learning approaches. The data augmentation techniques such as brightness intensity increase, zoom, and image rotation increase spatial detection of parts in varying visual instances. The data augmentation techniques such as zoom and brightness intensity change also increase accuracy in the detection of parts shape and texture.

To detect moving objects in real-time, the local optical flow method of Lucas Kanade is evaluated, which estimates the sparse motion between two consecutive frames using the corner features extracted. To solve for the pixel displacement between consecutive frames, the Lucas Kanade method assumes brightness consistency, as illustrated in Equation 6.1, in which the pixel brightness intensity with respect to changes in pixel position I_x,I_y in consecutive frames over time, I_t, remains the same. The constant pixel intensity values in spatial and temporal derivatives allow solving for the unknown parameters u and v, which represent the pixel displacement in x and y directions, respectively. To resolve the aperature problem of single gradient pixel direction and indicate true object motion, the local method assumes constant motion among neighboring pixels in a defined region, as outlined in Equation 6.2. This is significant as the optical flow velocity vector, v in Equation 6.2, is valid in regions, defined within matrix A, over the spatiotemporal derivative. The optical flow velocity vector is obtained by solving the least square approximation in Equation 6.3. Specifically, the matrix, $A^\top A$, represents the structure tensor of an image at point p, in which the larger matrix eigenvalues λ meet the assumption of constant motion within the local neighborhood of pixels. This is significant as the structure tensor matrix,$A^\top A$, used to valid image regions for the optical flow application resembles the parameters outlined in the Harris Corner Detector. Therefore, to ensure accuracy in real-time moving object detection with optical flow, the features extracted from the Harris Corner Detector are applied as corners consist of higher eigenvalues, illustrated in Figure 6.1. Based on the local method evaluation, the results illustrated fewer parts detected in outer shelves, which vary through different regions within the shelves. This is significant as the parts in the base shelf rotate at a higher rate than in parts within the higher shelves. Therefore, the pixels of a moving object within the base shelf consist of small motion thus resulting in a higher number of detections in the base shelf.

$$I_x u + I_y v + I_t = 0 \tag{6.1}$$

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \tag{6.2}$$

$$A^\top A = \begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix} \hat{x} = \begin{bmatrix} u \\ v \end{bmatrix} A^\top b = -\begin{bmatrix} \sum_{p\in P} I_x I_t \\ \sum_{p\in P} I_y I_t \end{bmatrix} \tag{6.3}$$

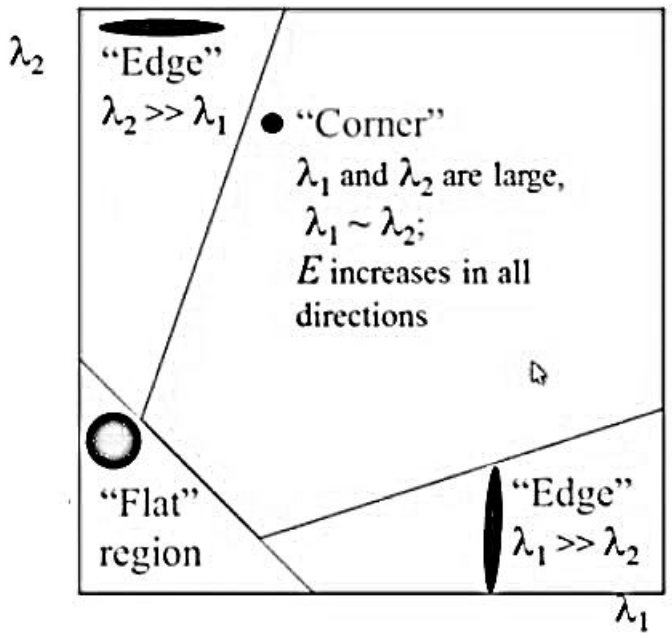


Figure 6.1: Classification of Image Points as Edges and Corners[8]

To increase real-time detections of moving objects, the global Gunnar-Farneback optical flow method was applied, which estimates the dense motion between two consecutive frames based on polynomial expansion. Based on the Gunnar-Farneback evaluation, the results illustrated an increased number of part detections on the outer shelves of the bowl-feeder than on the base shelf accumulated with parts consisting of small motion. Within the regions of outer shelves, the real-time detection of parts is intermittent. This is significant as the outer shelves consist of background regions with illumination variation and

continuous rotation of circular parts, which have varying surface colors. This illustrates the failure to meet the brightness consistency assumption and the constant pixel intensity values. The changes in pixel intensity of parts due to variation in illumination and surface color reflectance resulted in intermittent real-time parts detection. Therefore, the coefficients, in Equations 6.4, cannot be equated and solved for pixel displacement as increased unknown parameters exist due to changes in pixel intensity values. In comparison to the local method of Lucas Kanade optical flow, the Gunnar-Farneback method illustrated an increased number of detections. This is significant as the local method applies motion estimation for features detected using Harris corner detectors. As the outer shelves of the bowl-feeder consist of illumination variation, the failure in brightness consistency assumption results in fewer detections. Specifically, fewer detections resulted as part regions with most discontinuities in illumination, surface reflectance, depth, and surface normal are exposed to the lighting distribution changes within the bowl-feeder surface. Whereas, the global Gunnar-Farneback method approximated motion estimation for all pixels within the image thus resulting in an increased number of parts detected even in presence of the illumination variation.

$$\begin{aligned} \mathbf{A}_2 &= \mathbf{A}_1 \\ \mathbf{b}_2 &= \mathbf{b}_1 - 2\mathbf{A}_1\mathbf{d} \\ c_2 &= \mathbf{d}^T\mathbf{A}_1\mathbf{d} - \mathbf{b}_1^T\mathbf{d} + c_1. \end{aligned} \tag{6.4}$$

In comparison to the moving parts detection using traditional global and local methods, the application of the deep learning-based model RAFT resulted in an increased number of moving parts detections. As the traditional local Lucas-Kanade and global Gunnar-Farneback methods assume brightness constancy and smooth flow, the pixel intensity variation due to illumination variation and occlusion in varying bowl-feeder regions result in missed part detections. Although the global RAFT deep learning-based method assumes smooth flow and brightness constancy, the integration of patch-based feature extraction using different filter sizes attenuates noise in image regions with varying illumination. In order to address the large displacement of assembly parts, the RAFT method also applies a correlation volume, which is constructed using the multi-scale filters of the 4-layer correlation pyramid. The different filter sizes allow the detection of smaller and larger objects with large displacement using multi-scale features. With the integration of a multi-scale feature network and the 4-layer correlation pyramid, the deep learning-based method allows for increased moving object detection in presence of limited visual characteristics. In comparison to the RAFT method, this is significant as the Gunnar-Farneback method illustrated a lower number of moving parts detections. Specifically, due to the failure in brightness con-

stancy assumption, which prevented tracking pixel displacement in consecutive frames due to pixel intensity changes. The pixel intensity changes caused an increase in the unknown parameters to solve for pixel coefficient displacement using the polynomial expansion. As the RAFT optical flow method outperformed traditional local and global optical flow, the iteratively updated flow field of RAFT was applied to compute optical flow and create a pixel-based tracker. Based on the RAFT optical flow evaluation to detect moving objects and stationary objects as separate studies, the results showed increased accuracy in the detection of moving objects in comparison to the detection of stationary objects. The application of RAFT to detect stationary objects resulted in failed detections as the flow magnitude threshold value of object motion was declared to be 0. This is significant as the inverse binary masks generated not only detected foreground stationary objects but also background objects, which remained stationary. Therefore, the iteratively updated flow field of moving objects in consecutive frames was applied to create a pixel-based tracker. Based on the moving object detection using RAFT, the 2D flow field visualization illustrated accuracy in assembly parts motion. Specifically, the accumulated parts within the base-shelf consist of small motion, whereas the outer shelves with less comparative parts consist of larger motion displacement in a counter-clockwise direction.

As the RAFT method illustrated only current real-time moving objects, the pixel-based tracker was created to illustrate current and previous moving parts within the various automated assembly machines. The tracker differentiated the previous and current moving parts in different colored segments and simultaneously visualized the flow field to illustrate the movement direction and magnitude. Based on the pixel-based tracker evaluation, the results visualization accurately illustrated the counter-clockwise movement of assembly parts within the various automated assembly machines. Specifically, the counter-clockwise movement of bowl-feeder assembly parts, the rotary indexer motion in a counter-clockwise direction, the movement direction of the robotic manipulator, and the assembly plate within the digital assembly. The pixel-based tracker illustrated the current moving objects and previously moved objects in white and blue colored segments, respectively. Based on the results visualization, the tracker correctly visualized the assembly parts within the base shelf in the blue-colored segment as the parts move less frequently in comparison to the moving parts in the outer shelves of the bowl feeder. As the moving parts in the outer shelves of the bowl-feeder move more frequently and with larger motion displacement, the tracker correctly visualized the current moving parts in a white-colored segment.

In order to compare the segmentation performance of the pixel-based tracker with the deep learning-based segmentation model, the space-time graph neural network (ST-GNN) model was applied. Based on the ST-GNN model evaluation, the generated segmentation masks consisted of decreased boundary IoU, which resulted in increased alignment accuracy

of segmentation masks with parts boundary. In comparison to the ST-GNN segmentation performance, the pixel-based tracker consisted of increased boundary IoU, which resulted in overlapping segmentations of parts and bowl-feeder background surface. Therefore, the ST-GNN model illustrated increased accuracy in segmentation performance than the pixel-based tracker. Additionally, as the pixel-based tracker visualized moving parts in different colored segments, the region of interest extraction can be challenging in presence of masks with varying shapes and colors. As the ST-GNN model addressed the limited dataset challenge by learning visual correspondence as a contrastive random walk in palindrome sequences, the ST-GNN was applied further to perform motion segmentation in surveillance videos of various assembly machines.

To reduce repeated manual annotations and accelerate model deployment in real-time production, the model initialization was automated to generate a first-frame annotation mask. Appearance-based segmentation methods were evaluated to group objects based on pixel similarity, user-based feedback, and classes defined within a pre-trained object detection model. As the image regions located within the bowl-feeder machine consisted of discontinuities in illumination, surface-reflectance, orientation, and depth, local variances existed within pixel intensity. An example of such discontinuities is illustrated in Figure 6.2.The variations in pixel intensity allowed the detection of low-level features such as object texture, color, and shape. To distinguish between foreground parts and the background surface of the bowl-feeder, pixel-based segmentation methods were evaluated to group pixels based on variations in low-level features.

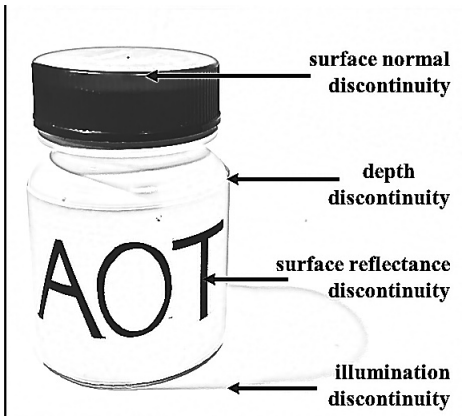


Figure 6.2: Illustration of discontinuities in surface normal, depth, surface reflectance and illumination[70]

The difference in the spatial arrangement of regions, with local variances in pixel intensity, leads to the formation of different texels within an image. The variation in pixel intensity properties and spatial relationship between texels forms textures such as fine, grained, coarse, and smooth. As image objects with varying surface reflectance and illumination result in different textured surfaces, characteristics in the spatial arrangement of regions such as texture repetition, directionality, and complexity can be applied to segment foreground and background objects in an image. In order to segment images based on texture repetition, grey level co-occurrence matrix (GLCM) and Gabor filter were applied to differentiate based on orientation on the spatial frequencies of the texture pattern. Based on the segmentation results, the GLCM and Gabor filter performed better in the segmentation of dense parts than the Entropy-based segmentation. Specifically,467 and 476 parts were segmented, using GLCM and Gabor filter, of the total 476 parts present within the bowl-feeder.The GLCM texture analysis calculated statistical measures such as contrast, correlation, homogeneity and energy values to segment foreground parts.

Although the texture-based analysis segmented foreground objects of similar texture characteristics, multiple post-processing steps would be required to select and extract region of interests. In comparison to the texture-based segmentation, the region-merging watershed method segmented 377 parts within the total 476 parts present within the bowl-feeder machine. Although the watershed segmentation consisted of decreased number of segmentation, the generated segmentation mask consisted of decreased mask IoU and accurate mask alignment with parts boundary. This is significant as the K-means and gaussian blur applied to the image reduced superpixel oversegmentation and detection of background surface. In comparison to the region-based watershed segmentation, the graph-based merging method segmented 0 parts of the 476 parts present within the bowl-feeder.This is significant as gaussian blur was not applied with graph-based segmentation to reduce noise present within the background surface. Therefore, the segmentation results illustrated that the watershed segmentation, meanshift, GLCM and gabor filter accurately segmented an increased number of foreground parts within the bowl-feeder.

As limitations in visual characteristics such as occlusion and illumination change cause pixel intensity variation, pixel-based segmentation methods oversegment image regions in presence of increased noise. To address the decreased segmentation accuracy posed by oversegmented regions, interactive-based segmentation methods were evaluated to segment parts within the ATS vibratory bowl-feeder. The user feedback within deep learning-based and manual annotation segmentation served as location guidance to interactively mark the mislabeled regions and provide precise boundary segmentation of the foreground object. To reduce computational costs in applying deep learning-based interactive segmentation on bowl-feeder datasets, the feature backpropagating refinement scheme (f-BRS) was eval-

uated, which performed backpropagation on only intermediate layers within the network rather than the whole network, as illustrated in Figure 2.10.

After f-BRS model evaluation with ground truth mask generated with RAFT, the segmentation results illustrated an increase in mislabeled regions, misaligned mask intersection over union (IoU) with parts boundary with an applied zoom-in feature, and an increase of 30-50 interactive clicks required to generate an object segmentation mask. Additionally, as illustrated by the green dots located in Figure 5.10 over bowl-feeder parts, minimal segmentation resulted even with a significant increase in user clicks provided. This is significant as the f-BRS model applies the DeepLabV3+ network architecture trained on the Semantic Boundaries Dataset (SBD) 8,498 images and annotations of object classes such as vehicles, households, and animals. The DeepLabV3+ consists of a ResNet backbone with atrous convolutions for feature extraction, atrous spatial pyramid pooling (ASPP) module for resampling feature map at different rates and segmenting the object at multiple scales using semantic segmentation annotations, and a 1x1 convolution to output final segmentation from concatenated masks.

As the model was trained on image features and respective semantic labels of the SBD dataset, the current segmentation applied to the bowl-feeder dataset resulted in mislabeled regions and misaligned masks as pixel-wise classification corresponds to pre-trained SBD semantic labels. Due to the challenge of a limited dataset of ATS machines, and lower image resolution present within existing public bowl-feeder videos, additional changes were not applied to the model training parameters. Additionally, the f-BRS optimization task, outlined in Equation 6.5, minimized mean squared area loss in the provided locations of user clicks, x_i, y_i, by regularizing channel-wise scaling and bias in the intermediate layers of the network. This is significant as the initially predicted mask, M^{pred}, with mislabeled regions resulted due to pixel-wise correspondence on pre-trained pixel labels and spatial information; therefore, an increase in negative user clicks was required to classify objects as background and reduce mislabeled regions. Additionally, the evaluation of the zoom-in feature within the f-BRS model illustrated an increase in mislabeled regions. As the zoom-in feature applied the location of the first-click prediction mask for bounding box localization and image cropping, the decrease in initial mask IoU caused the feature to direct towards incorrect background objects thus resulted in mislabeled regions.

$$\lambda \left(\|S - 1\|_2 + \|B\|_2 \right) + \sum_{i=1}^{n} \left(M_{x_i, y_i}^{pred} - l_i \right)^2 \tag{6.5}$$

As the f-BRS model evaluation showed a decrease in mask IoU and an increase in mislabeled regions and the user clicks required, the GrabCut annotation tool was evaluated.

The manual annotation tool allows the user to select an initial region of interest (ROI) and designate the background and foreground object using classID labels of different colors and bounding boxes. After setting the classID to foreground objects, the user can begin to manually scribble a mask on top of the bowl-feeder parts. Although the GrabCut annotation tool required increased user feedback and manual scribbling on foreground objects, the results illustrated increased accuracy in segmentation mask generation than the segmentation results generated using an f-BRS model. This is significant as the bounding box selection allowed accurate binary segmentation between the foreground area as a region within the bowl-feeder and the background area as a region outside of the bowl-feeder. Specifically, to perform binary segmentation through an iterative update of the region space, which was modeled as a mixture of Gaussians in color space. The accuracy in localization between background and foreground region allowed producing accurate classID labels for specific objects through manual scribble-based annotation. Therefore, segmentation results illustrated that the Grabcut annotation tool performed better in comparison to the pre-trained f-BRS model on the SBD dataset.

Although Grabcut annotation tool allowed accurate segmentation of all parts within the bowl feeder, the process required manual annotation thus decreasing efficiency in the ST-GNN model initialization. To accelerate model deployment in real-time production, supervised segmentation methods were evaluated to enable automatic initialization and accurately segment all parts within the bowl-feeder. For real-time detection of moving parts within the ATS machines, the Detectron2 deep learning-based detection model was trained with applied data augmentation and evaluated on the original frames of the bowl-feeder dataset. The segmentation masks were then created from the detection results and evaluated for mask IoU with parts boundary.

The Detectron2 model was evaluated to perform part detection and segmentation within the bowl-feeder. The model consisted of a backbone network for the generation of feature maps at different scales, a region proposal network to detect regions with objects, and an ROI box head for the classification of a detected object. To ensure robust detection performance, the dataset consisted of shuffled frames with instances of variational parts filled in bowl-feeder such as minimal-filled and dense part clusters. To apply model generalization, the datasets consisted of bowl-feeders frames with variational part types consisting of metal bolts, springs, and pipettes. After applying model training and evaluation, the segmentation results illustrated the detection of all parts within the titan bowl-feeder datasets with complete mask IoU alignment with the part boundary. Due to the lower image resolution of part shape, the results consisted of missed detections in other instances of the bowl-feeder videos. The results showed detection of each part with varying orientation and with a varying accumulation of parts within different shelves of the bowl-feeder.

To detect the spatial anomalies within the various manufacturing datasets, the handcrafted methods and deep learning-based models were applied. As the moving objects were detected using the RAFT optical flow method, the generated segmentation masks detected motion on the background conveyor belt surface. Therefore, the application of preprocessing methods such as morphological operations reduced background noise and the steps required to extract the region of interests. Based on the results of spatial anomaly detection, the handcrafted methods such as color and HOG texture descriptor indicated the missing part within the assembly cell. The variation in the RGB values of the color histogram and the gradient in texture orientation illustrated the absence of blue rectangular part within the assembly cell. As the application of handcrafted features required ROI extract of the same shape and size, additional post-processing steps were required. To increase efficiency in anomaly detection, the deep learning-based PatchCore anomaly detection method was evaluated. Based on the evaluation of PatchCore model, the results illustrated increased accuracy in identification and localization of various spatial anomalies within the assembly parts. The generated segmentation maps consisted of decreased mask IoU and increased accuracy in mask alignment with parts boundary. This is significant as the generation of synthetic manufacturing dataset with higher image resolution increased detection accuracy of spatial pixel characteristics. Therefore, in comparison to the handcrafted methods, the Patchcore method increased efficiency in anomaly detection and provided an accurate segmentation map to localize the spatial anomalies.

6.1 Conclusion

In this study, we addressed the increased production cycle-time and reduced machine operator safety as repeated human intervention is required to manually clear part jams within varying locations of the subsystems. To ensure machine operator safety and reduce production cycle-time, we performed spatiotemporal part segmentation within the various manufacturing lines. To address limitations within the dataset characteristics such as illumination variance and lower image resolution, we performed preprocessing methods and color conversion techniques for real-time moving object detection.

Due to the lower image resolution, we created the synthetic manufacturing dataset with different camera viewpoints and verified complete image alignment using affine-based transformation and RANSAC. To ensure robust image registration, we compared the color and texture handcrafted features of the registered images. To compare the performance between handcrafted features and deep learning-based anomaly detection method, we evaluated Patchcore Anomaly Detection method, pre-trained on manufacturing dataset, with

Flex-Sim dataset. Based on evaluation of anomaly detection methods, the PatchCore method generated a segmentation map for anomaly localization and increased system efficiency.

To perform motion-based and appearance-based segmentation, we followed a bottom-up architecture to evaluate computer vision techniques and deep learning-based models. In presence of occlusion and illumination variance, the deep learning-based optical flow RAFT showed increased real-time moving object detection than classical local and global methods in optical flow. Therefore, RAFT was applied to compute optical flow and its iteratively updated flow field were applied to create a pixel-based object tracker. We compared the segmentation performance of an optical flow-based tracker with a space-time graph neural network (ST-GNN), and it showed increased accuracy in boundary mask IoU alignment than the pixel-based tracker.

To enable automatic initialization of the ST-GNN model, we explored appearance-based segmentation methods such as pixel-based, interactive-based, and deep learning-based segmentation methods. After evaluation of these methods on the bowl-feeder dataset, the supervised segmentation method outperformed the pixel-based and interactive-based segmentation methods.

6.1.1 Future Work

This study focused on spatiotemporal segmentation of parts within the automated manufacturing machines. To improve performance of the spatiotemporal segmentation methods in future, additional manufacturing datasets are required for application in model training and evaluation. The datasets should consist of reduced limited visual characteristics and similar spatiotemporal characteristics. To detect spatiotemporal irregularities within the machines, the future work consists of performing feature extraction and a time-series spatio-temporal analysis. During the spatiotemporal time-series analysis, extracted features within current frame will be compared with normal behavior part characteristics for anomaly detection.

References

[1] Camera vision inspection system. https://www.platzhirschclub-davos.com/kdyo?cname=camera+vision+inspection+system&cid=63, August 2022.

[2] U Ahmad, MAP Adji, et al. Accuracy in estimating visual quality parameters of mango fruits as moving object using image processing. In *IOP Conference Series: Earth and Environmental Science*, volume 542, page 012008. IOP Publishing, 2020.

[3] Zubair Akhter, Abhishek Kumar Jha, and Mohammad Jaleel Akhtar. Generalized rf time-domain imaging technique for moving objects on conveyor belts in real time. *IEEE Transactions on Microwave Theory and Techniques*, 65(7):2536–2546, 2017.

[4] UYAR Ali. Cost and management accounting practices: a survey of manufacturing companies. *Eurasian Journal of Business and Economics*, 3(6):113–125, 2010.

[5] Peter K Allen, Aleksandar Timcenko, Billibon Yoshimi, and Paul Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation*, 9(2):152–165, 1993.

[6] Laura Arnal, J Ernesto Solanes, Jaime Molina, and Josep Tornero. Detecting dings and dents on specular car body surfaces based on optical flow. *Journal of Manufacturing Systems*, 45:306–321, 2017.

[7] Ahmadreza Baghaie, Roshan M D'Souza, and Zeyun Yu. Dense descriptors for optical flow estimation: a comparative study. *Journal of Imaging*, 3(1):12, 2017.

[8] U Baid. Classification of points using eigen values and r. *Shri Guru Gobind Singhji Inst. Eng. Technol., Nanded, India, Tech. Rep*, 2022.

[9] Christian Barat and Maria-Joao Rendas. A robust visual attention system for detecting manufactured objects in underwater video. In *OCEANS 2006*, pages 1–6. IEEE, 2006.

[10] Adel Benamara, Serge Miguet, and Mihaela Scuturici. Real-time multi-object tracking with occlusion and stationary objects handling for conveying systems. In *International Symposium on Visual Computing*, pages 136–145. Springer, 2016.

[11] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[12] D Bhargava, S Vyas, and Ayushi Bansal. Comparative analysis of classification techniques for brain magnetic resonance imaging images. In *Advances in Computational Techniques for Biomedical Image Analysis*, pages 133–144. Elsevier, 2020.

[13] Josef Bigün, Goesta H. Granlund, and Johan Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(08):775–790, 1991.

[14] R.R. Chand, R. Subbiah, G. Manohar, P.S.H. Jose, L. Girisha, and D.K. Veeranna. *Computer Integrated Manufacturing: FOR INDUSTRIAL AUTOMATION*. Forschung Publications, 2020.

[15] Chaves. Glcms-a great tool for your ml arsenal. https://towardsdatascience.com/glcms-a-great-tool-for-your-ml-arsenal-7a59f1e45b65, Jan 2022.

[16] Ricardo Omar Chavez-Garcia and Olivier Aycard. Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):525–534, 2015.

[17] Bo-Hao Chen, Ling-Feng Shi, and Xiao Ke. A robust moving object detection in multi-scenario big data for video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(4):982–995, 2018.

[18] Vladimir Chernov, Jarmo Alander, and Vladimir Bochko. Integer-based accurate conversion between rgb and hsv color spaces. *Computers & Electrical Engineering*, 46:328–337, 2015.

[19] Chhikara. Understanding-morphological-image-processing-and-its-operations. https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756, 2022.

[20] Jaemin Cho, Sangseung Kang, and Kyekyung Kim. Real-time precise object segmentation using a pixel-wise coarse-fine method with deep learning for automated manufacturing. *Journal of Manufacturing Systems*, 62:114–123, 2022.

[21] Amira Chriki, Haifa Touati, Hichem Snoussi, and Farouk Kamoun. Deep learning and handcrafted features for one-class anomaly detection in uav video. *Multimedia Tools and Applications*, 80(2):2599–2620, 2021.

[22] Yuval Cohen, Maurizio Faccio, Francesco Pilati, and Xifan Yao. Design and management of digital manufacturing and assembly systems in the industry 4.0 era, 2019.

[23] Elias Ribeiro da Silva, Ana Carolina Shinohara, Christian Petersson Nielsen, Edson Pinheiro de Lima, and Jannis Angelis. Operating digital manufacturing in industry 4.0: the role of advanced manufacturing technologies. *Procedia CIRP*, 93:174–179, 2020.

[24] Norwalt Design. Custom machinery. https://www.norwalt.com/designing-automated-assembly-machines-pt1/, Jun 2020.

[25] Samer Dofash. Core transformation process for a manufacturing company. https://www.slideshare.net/samerdofash/manufacturing-and-service-technologies, August 2022.

[26] Shuchen Du. Understanding optical flow amp; raft. https://towardsdatascience.com/understanding-optical-flow-raft-accb38132fba, Sep 2020.

[27] Tonichi Edeza. Image processing with python- working with entropy. https://towardsdatascience.com/image-processing-with-python-working-with-entropy-b05e9c84fc36, Jan 2021.

[28] Gunnar Farneback. Orientation estimation based on weighted projection onto quadratic polynomials. In *5th International Fall Workshop. Vision, Modeling, and Visualization 2000, 22-24 November 2000, Saarbrücken, Germany*, pages 89–96, 2000.

[29] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[30] Kostas Haris, Serafim N Efstratiadis, Nikolaos Maglaveras, and Aggelos K Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on image processing*, 7(12):1684–1699, 1998.

[31] Dirk Holz, Angeliki Topalidou-Kyniazopoulou, Jörg Stückler, and Sven Behnke. Real-time object detection, localization and verification for fast robotic depalletizing. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1459–1466. IEEE, 2015.

[32] Hiroto Honda. Detailed architecture of base-rcnn-fpn. https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd, August 2022.

[33] Bin-Juine Huang, Cheng-Kang Guan, Shih-Han Huang, and Wei-Fang Su. Development of once-through manufacturing machine for large-area perovskite solar cell production. *Solar Energy*, 205:192–201, 2020.

[34] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[35] Yuri Ivanov, Aaron Bobick, and John Liu. Fast lighting independent background subtraction. *International Journal of Computer Vision*, 37(2):199–207, 2000.

[36] IzMiran. Using a gray-level co-occurrence matrix analyzing and enhancing images. http://matlab.izmiran.ru/help/toolbox/images/enhanc15.html, 2005.

[37] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020.

[38] Vijay Kakani, Van Huan Nguyen, Basivi Praveen Kumar, Hakil Kim, and Visweswara Rao Pasupuleti. A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*, 2:100033, 2020.

[39] Bahadir Karasulu and Serdar Korukoglu. Moving object detection and tracking by using annealed background subtraction method in videos: Performance optimization. *Expert Systems with Applications*, 39(1):33–43, 2012.

[40] Muhammad Monjurul Karim, David Doell, Ravon Lingard, Zhaozheng Yin, Ming C Leu, and Ruwen Qin. A region-based deep learning algorithm for detecting and tracking objects in manufacturing plants. *Procedia Manufacturing*, 39:168–177, 2019.

[41] Kyekyung Kim, Joongbae Kim, Sangseung Kang, Jaehong Kim, and Jaeyeon Lee. Object recognition for cell manufacturing system. In *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 512–514. IEEE, 2012.

[42] Sergey Alekseevich Korchagin, Sergey Timurovich Gataullin, Aleksey Viktorovich Osipov, Mikhail Viktorovich Smirnov, Stanislav Vadimovich Suvorov, Denis Vladimirovich Serdechnyi, and Konstantin Vladimirovich Bublikov. Development of an optimal algorithm for detecting damaged and diseased potato tubers moving along a conveyor belt using computer vision systems. *Agronomy*, 11(10):1980, 2021.

[43] Gerald Kuehne, Stephan Richter, and Markus Beier. Motion-based segmentation and contour-based classification of video objects. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 41–50, 2001.

[44] Manoj Kumar, Susmita Ray, and Dileep Kumar Yadav. A survey on moving object detection in video using a moving camera for smart surveillance system. In *Proceedings of International Conference on Machine Intelligence and Data Science Applications*, pages 241–253. Springer, 2021.

[45] Rakesh Kumar, Harpreet Sawhney, Supun Samarasekera, Steve Hsu, Hai Tao, Yanlin Guo, Keith Hanna, Art Pope, Richard Wildes, David Hirvonen, et al. Aerial video surveillance and exploitation. *Proceedings of the IEEE*, 89(10):1518–1539, 2001.

[46] Jae-Sung Kwon, Jong-Min Lee, and Whoi-Yul Kim. Real-time detection of foreign objects using x-ray imaging for dry food manufacturing line. In *2008 IEEE International Symposium on Consumer Electronics*, pages 1–4. IEEE, 2008.

[47] D.M. Lane, M.J. Chantler, and Dongyong Dai. Robust tracking of multiple objects in sector-scan sonar image sequences using optical flow motion estimation. *IEEE Journal of Oceanic Engineering*, 23(1):31–46, 1998.

[48] Hongzu Li and Pierre Boulanger. Structural anomalies detection from electrocardiogram (ecg) with spectrogram and handcrafted features. *Sensors*, 22(7):2467, 2022.

[49] Min Li and Jian Jun Liao. Texture image segmentation based on glcm. In *Applied Mechanics and Materials*, volume 220, pages 1398–1401. Trans Tech Publ, 2012.

[50] Guoyuan Liang, Fan Chen, Yu Liang, Yachun Feng, Can Wang, and Xinyu Wu. A manufacturing-oriented intelligent vision system based on deep neural network for

object recognition and 6d pose estimation. *Frontiers in Neurorobotics*, 14:616775, 2021.

[51] Santosh Lohumi, Byoung-Kwan Cho, and Sangdeok Hong. Lctf-based multispectral fluorescence imaging: System development and potential for real-time foreign object detection in fresh-cut vegetable processing. *Computers and Electronics in Agriculture*, 180:105912, 2021.

[52] Pocholo James M Loresco, Ira C Valenzuela, and Elmer P Dadios. Color space analysis using knn for lettuce crop stages identification in smart farm setup. In *TENCON 2018-2018 IEEE Region 10 Conference*, pages 2040–2044. IEEE, 2018.

[53] Shan Lou, Luca Pagani, Wenhan Zeng, X Jiang, and PJ Scott. Watershed segmentation of topographical features on freeform surfaces and its application to additively manufactured surfaces. *Precision Engineering*, 63:177–186, 2020.

[54] Ren C. Luo and Chun-Hao Liao. Robotic conveyor tracking with dynamic object fetching for industrial automation. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pages 369–374, 2017.

[55] Ren C Luo and Chun-Hao Liao. Robotic conveyor tracking with dynamic object fetching for industrial automation. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pages 369–374. IEEE, 2017.

[56] Yongqing Lv, Bing Liu, Ning Liu, and Minghui Zhao. Design of automatic speed control system of belt conveyor based on image recognition. In *2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 227–230. IEEE, 2020.

[57] Bo Ma and Zheru Chi. Texture image segmentation based on entropy theory. In *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, volume 1, pages 103–108 Vol. 1, 2004.

[58] Vaïa Machairas, Etienne Decencière, and Thomas Walter. Waterpixels: Superpixels based on the watershed transformation. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4343–4347. IEEE, 2014.

[59] Mario Malave. Will 3-d printing take us to mars? relativity space thinks so.

[60] Amolkirat Singh Mangat, Juergen Mangler, and Stefanie Rinderle-Ma. Interactive process automation based on lightweight object detection in manufacturing processes. *Computers in Industry*, 130:103482, 2021.

[61] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625, 2018.

[62] Fardin Mirzapour and Hassan Ghassemian. Using glcm and gabor filters for classification of pan images. In *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pages 1–6. IEEE, 2013.

[63] Fardin Mirzapour and Hassan Ghassemian. Fast glcm and gabor filters for texture classification of very high resolution remote sensing images. 2015.

[64] Kazuaki Nakamura, Naoko Nitta, Noboru Babaguchi, Kensuke Fujii, Satoki Matsumura, and Eiji Nabata. Semi-supervised temporal segmentation of manufacturing work video by automatically building a hierarchical tree of category labels. *IEEE Access*, 9:68017–68027, 2021.

[65] S Nashat, A Abdullah, S Aramvith, and MZ Abdullah. Support vector machine approach to real-time inspection of biscuits on moving conveyor belt. *Computers and Electronics in Agriculture*, 75(1):147–158, 2011.

[66] Stefan Escaida Navarro, David Weiss, Denis Stogl, Dimitar Milev, and Bjoern Hein. Tracking and grasping of known and unknown objects from a conveyor belt. In *ISR/Robotik 2014; 41st International Symposium on Robotics*, pages 1–8. VDE, 2014.

[67] Yosep Oh. *Assembly design and production planning towards additive manufacturing-based mass customization*. PhD thesis, State University of New York at Buffalo, 2019.

[68] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. In *Science and information conference*, pages 128–144. Springer, 2019.

[69] Bernhard Preim and Charl Botha. *Image Analysis for Medical Visualization*. 2014.

[70] Mengyang Pu, Yaping Huang, Qingji Guan, and Haibin Ling. Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6879–6888, 2021.

[71] Quqixun. Quqixun/imageregistration: A demo that implement image registration by matching sift descriptors and appling ransac and affine transformation. https://github.com/quqixun/ImageRegistration.

[72] Andrik Rampun, Harry Strange, and Reyer Zwiggelaar. Texture segmentation using different orientations of glcm features. In *Proceedings of the 6th International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, pages 1–8, 2013.

[73] Ray. Computer vision-watershed algorithm. https://medium.com/analytics-vidhya/computer-vision-watershed-algorithm-ca16bd00485, Sep 2020.

[74] D. Rembold, U. Zimmermann, T. Langle, and H. Worn. Detection and handling of moving objects. In *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*, volume 3, pages 1332–1337 vol.3, 1998.

[75] K Roth, L Pemula, J Zepeda, B Schölkopf, T Brox, and P Gehler. Towards total recall in industrial anomaly detection. arxiv 2021. *arXiv preprint arXiv:2106.08265*.

[76] Eli S Saber and A Murat Tekalp. Integration of color, edge, shape, and texture features for automatic region-based image annotation and retrieval. *Journal of Electronic Imaging*, 7(3):684–700, 1998.

[77] Safa Sadaghiyanfam. Using gray-level-co-occurrence matrix and wavelet transform for textural fabric defect detection: A comparison study. In *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, pages 1–5. IEEE, 2018.

[78] Stefano Savian, Mehdi Elahi, and Tammam Tillo. Optical flow estimation with deep learning, a survey on recent advances. In *Deep biometrics*, pages 257–287. Springer, 2020.

[79] Silvia Sellan, Jacob Kesten, Ang Yan Sheng, and Alec Jacobson. Opening and closing surfaces. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020.

[80] Anuj Shah. Through the eyes of gabor filter. https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97, Jun 2018.

[81] Sai Shashank. Detectron2 vs. yolov5 (which one suits your use case better?). https://medium.com/ireadrx/detectron2-vs-yolov5-which-one-suits-your-use-case-better-d959a3d4bdf, 2022.

[82] S Shishira, Vidyadhar Rao, and Sithu D Sudarsan. Proximity contours: Vision based detection and tracking of objects in manufacturing plants using industrial control systems. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 1021–1026. IEEE, 2019.

[83] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8623–8632, 2020.

[84] Lei Su, Hua Huang, Lunming Qin, and Wenbin Zhao. Transformer vibration detection based on yolov4 and optical flow in background of high proportion of renewable energy access. *Frontiers in Energy Research*, page 71, 2022.

[85] Jiaze Sun, Huijuan Lee, and Jun Yang. The impact of the covid-19 pandemic on the global value chain of the manufacturing industry. *Sustainability*, 13(22):12370, 2021.

[86] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[87] Kazuhito Takahashi. Grabcut-annotation-tool. https://github.com/Kazuhito00/GrabCut-Annotation-Tool/blob/main/README_EN.md.

[88] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. pages 402–419. Springer, 2020.

[89] Sebastian Thiede, Poorya Ghafoorpoor, Brendan P Sullivan, Sebastian Bienia, Michael Demes, and Klaus Dröder. Potentials and technical implications of tag based and ai enabled optical real-time location systems (rtls) for manufacturing use cases. *CIRP Annals*, 2022.

[90] Mihran Tuceryan and Anil K Jain. Texture analysis. *Handbook of pattern recognition and computer vision*, pages 235–276, 1993.

[91] Furkan Ulger, Seniha Esen Yuksel, and Atila Yilmaz. Anomaly detection for solder joints using β-vae. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(12):2214–2221, 2021.

[92] Dainius Varna and Vytautas Abromavičius. A system for a real-time electronic component detection and classification on a conveyor belt. *Applied Sciences*, 12(11):5608, 2022.

[93] Harini Veeraraghavan, Osama Masoud, and Nikolaos P Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):78–89, 2003.

[94] Hong-Son Vu, Jia-Xian Guo, Kuan-Hung Chen, Shu-Jui Hsieh, and De-Sheng Chen. A real-time moving objects detection and classification approach for static cameras. In *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 1–2, 2016.

[95] Robert Ward, Payam Soulatiantork, Shaun Finneran, Ruby Hughes, and Ashutosh Tiwari. Real-time vision-based multiple object tracking of a production process: industrial digital twin case study. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 235(11):1861–1872, 2021.

[96] Alan Wong. Gabor filter in edge detection with opencv. https://www.freedomvc.com/index.php/2021/10/16/gabor-filter-in-edge-detection/, Oct 2021.

[97] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Gazedirector: Fully articulated eye gaze redirection in video. In *Computer Graphics Forum*, volume 37, pages 217–225. Wiley Online Library, 2018.

[98] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2022.

[99] Li Da Xu, Chengen Wang, Zhuming Bi, and Jiapeng Yu. Autoassem: An automated assembly planning system for complex products. *IEEE Transactions on Industrial Informatics*, 8(3):669–678, 2012.

[100] Dileep Kumar Yadav and Karan Singh. A combined approach of kullback–leibler divergence and background subtraction for moving object detection in thermal video. *Infrared Physics & Technology*, 76:21–31, 2016.

[101] Jing Yang, Shaobo Li, Zheng Wang, and Guanci Yang. Real-time tiny part defect detection system in manufacturing using deep learning. *IEEE Access*, 7:89278–89291, 2019.

[102] Ying-Hao Yu, QP Ha, and NM Kwok. Chip-based design for real-time moving object detection using a digital camera module. In *2009 2nd International Congress on Image and Signal Processing*, pages 1–5. IEEE, 2009.

[103] Haohao Zhao, Xuezhi Feng, and Yan Chen. Entropy-based texture analysis and feature extraction of urban street trees in the spatial frequency domain. In *MIPPR 2009: Automatic Target Recognition and Image Analysis*, volume 7495, pages 286–292. SPIE, 2009.

[104] Xiang Sean Zhou and Thomas S Huang. Edge-based structural features for content-based image retrieval. *Pattern recognition letters*, 2001.