

REAL-TIME AUTOMATED DEFECT DETECTION IN INDUSTRIAL MANUFACTURING USING DEEP LEARNING AND TEXTURE ANALYSIS

Vivekanandan Govindan Ekambaram

Senior Data Scientist, Kforce Inc
Milpitas, California, USA

Received: 24/08/2026

Revised: 23/09/2026

Accepted: 06/10/2026

ABSTRACT:

Visual inspection is a critical quality gate in industrial manufacturing, yet manual inspection is slow, tiring, and inconsistent, and classical machine vision struggles with variable appearance. This paper presents a real-time automated defect detection framework that combines deep learning with texture analysis. A controlled imaging station with a ring light, a camera, and a tray that holds many items at once provides repeatable images. Three processing modules follow: image pre-processing by grayscale conversion, Sobel edge detection, and binarization, which emphasizes texture and edge structure; target object recognition with a YOLO detector that localizes the objects of interest; and defect detection by measuring the percentage by which an object's extent exceeds a normal baseline. In parallel, a four-block convolutional neural network classifies individual items as good or defective from their appearance, and texture descriptors based on gray-level co-occurrence, local binary patterns, and Gabor responses are fused with its output. On a balanced test set of 144 tablets, the convolutional network alone classified 139 correctly, an accuracy of 96.5% with a precision of 100% and a recall of 93.1%, with no good tablet rejected and five defective tablets missed. Fusing texture features reduced misses from five to two and raised accuracy to 97.9%. Pre-processing raised the object recognition rate under glare from 85.7% to 96.9%, and the complete pipeline processed an image in about 20 milliseconds on a graphics accelerator and about 68 milliseconds on an edge device. The results indicate that combining learned and handcrafted texture representations gives accurate, fast, and robust inspection suitable for production lines.

Keywords: Automated Defect Detection, Industrial Inspection, Deep Learning, Convolutional Neural Network, YOLO, Texture Analysis, Machine Vision, Real-Time Processing

INTRODUCTION

Quality inspection is one of the last and most important steps in manufacturing. A defective product that reaches a customer can cause injury, recalls, regulatory action, and loss of trust, while an unnecessary rejection wastes material and money. In many plants inspection is still performed by trained operators who examine items by eye. Human inspection is flexible, but it is slow, fatigues over a shift, varies from person to person, and cannot keep pace with modern production speeds. In regulated industries such as pharmaceuticals, authorities encourage manufacturers to build quality into the process through timely measurement, which favors inline, automated inspection over occasional sampling [25].

Machine vision has long been used to automate inspection. Classical systems extract handcrafted features, such as edges, shapes, and texture statistics, and apply rules or simple classifiers [6], [7], [11]. They are fast and interpretable, but they are sensitive to changes in illumination, position, and appearance, and they need to be redesigned for each product. Deep learning has transformed the field. Convolutional neural networks learn features directly from images and have achieved strong results in surface inspection and defect classification [1], [2], [13], [14]. Object detectors such as YOLO locate several objects in an image in a single pass and run fast enough for real-time use [3], [4], [5]. However, deep models need representative training data, can fail on rare or subtle defects, and may be difficult to interpret.

Texture analysis offers a complementary view. Many defects, such as cracks, chips, scratches, stains, and coating irregularities, alter the local texture of a surface before they alter its overall shape. Texture descriptors based on gray-level co-occurrence statistics, local binary patterns, and Gabor filter responses summarize this local structure

in a compact, interpretable form [6], [7], [8], and reviews of surface defect detection identify texture analysis as a major family of techniques [11], [12]. The combination of learned and handcrafted representations is attractive: the network captures complex patterns, and texture descriptors supply robust cues for faint defects.

This paper presents a real-time automated defect detection framework for industrial manufacturing that integrates deep learning and texture analysis. The physical acquisition station is shown in Fig. 1: a ring light, a camera, and a tray that holds many items so that a single image captures many items. The processing pipeline, shown in Fig. 2, has three modules: image pre-processing with grayscale conversion, Sobel edge detection, and binarization; target object recognition with YOLO; and defect detection that measures the percentage by which an object exceeds a normal baseline. A convolutional neural network, shown in Fig. 3, classifies items as good or defective, and its output is fused with texture features. We evaluate the framework on classification accuracy, object recognition rate, detection under threshold choices, robustness, and real-time performance.

The contributions of this paper are as follows:

- An integrated inspection framework that combines a controlled imaging station, texture-oriented pre-processing, YOLO-based object recognition, baseline-percentage defect detection, and a convolutional classifier.
- A fusion of learned features with texture descriptors that reduces missed defects without a large increase in false rejections.
- A detailed evaluation, including the confusion matrix of the classifier, comparison with texture-only and detector-only alternatives, the effect of pre-processing, threshold sensitivity, and robustness to disturbances.
- A real-time performance analysis on a graphics accelerator and an edge device, with a stage-by-stage latency breakdown.
- Practical guidance for deploying automated inspection on production lines.

The remainder of the paper is organized as follows. Section II reviews the literature. Section III discusses related work. Section IV presents the methodology. Section V describes the experimental setup. Section VI reports results, Section VII discusses implications and limitations, and Section VIII concludes.

LITERATURE REVIEW

A. Industrial Visual Inspection and Machine Vision

An inspection system must detect defects reliably, at the speed of the line, and with few false alarms. A machine vision system consists of illumination, optics, a camera, a processing unit, and a decision interface. Illumination is particularly important because it determines which surface features are visible; diffuse or ring lighting reduces shadows and glare and makes images more repeatable. Surveys of visual defect detection describe approaches from classical image processing to deep learning, and emphasize the need for robustness to variation in appearance and for real-time operation [12]. Because defects are rare, datasets are typically imbalanced, which complicates both training and evaluation.

B. Texture Analysis for Defect Detection

Texture refers to the spatial arrangement of gray levels in an image. Statistical approaches describe texture through measures such as contrast, correlation, energy, and homogeneity computed from the gray-level co-occurrence of neighboring pixels [6]. Local binary patterns encode the relation between each pixel and its neighbors into a compact histogram that is robust to monotonic illumination changes and discriminative across textures [7]. Filter-based approaches apply banks of Gabor filters at several scales and orientations to capture local frequency content, and are well suited to periodic textures and to irregularities in them [8]. Reviews of texture-based surface inspection show that these techniques detect defects as departures from the normal texture of a product [11]. Edge detection, for example with the Sobel operator, highlights boundaries and abrupt changes in intensity, and thresholding converts the result into a binary image that makes structure explicit [9], [10], [21].

C. Deep Learning for Defect Detection

Convolutional neural networks learn hierarchies of features through successive convolution and pooling layers followed by dense layers for classification [1], [2]. Applied to industrial inspection, they have been shown to classify and localize surface defects without handcrafted features [13], [14]. Design choices such as the number of convolutional blocks, the kernel sizes, and the use of dropout for regularization influence performance and efficiency [16]. Larger architectures such as very deep and residual networks give higher accuracy on large

datasets [18], [19] but are heavier to run, whereas compact networks are better suited to constrained hardware [22]. Data augmentation through geometric and photometric transformations is widely used to enlarge small datasets [20].

D. Object Detection with YOLO

YOLO treats detection as a single regression problem, predicting bounding boxes and class confidences directly from the full image in one pass [3]. Successive versions improved accuracy and speed [4], [5], [23]. For inspection, a detector can locate each item or region of interest within an image containing many items, which allows each to be analyzed separately. The confidence score that the detector assigns to each box expresses how certain the model is about the object, and high confidence values, near 0.99, indicate clear recognition. Detectors benefit from inputs in which the object of interest is clearly separated from the background, which motivates pre-processing steps that emphasize edges and suppress irrelevant detail.

E. Real-Time Constraints and Evaluation

Real-time inspection requires that the time to process an image not exceed the time between items arriving at the station, and that latency be predictable. Evaluation should include more than overall accuracy. Because the costs of the two error types differ, a missed defect, a false negative, is typically more serious than a false rejection, a false positive, and the confusion matrix, precision, recall, and the balance between them should be reported. Industrial benchmarks for anomaly detection provide standardized datasets for comparison [15], and explanation methods such as class activation maps help operators understand why an item was flagged [24].

RELATED WORK

Related work can be grouped into texture-based inspection, deep learning inspection, detector-based inspection, and hybrid approaches.

Texture-based inspection. Classical methods compute texture statistics and classify them with conventional classifiers or compare them with a reference texture [6], [7], [8]. A review of texture analysis techniques for surface defects summarizes statistical, structural, and filter-based approaches and their strengths for regular surfaces [11]. These methods are fast and need little data, but they are sensitive to lighting and require product-specific tuning.

Deep learning inspection. Convolutional networks for automated surface inspection and for feature extraction in industrial inspection have shown accuracy gains over handcrafted pipelines [13], [14]. Studies using benchmark datasets of industrial defects show that learned models detect many defect types, but also that subtle and rare defects remain challenging [15].

Detector-based inspection. YOLO-family detectors have been applied to locate components and defects within images [3], [4], [5], [23]. They provide localization and speed, but results depend on image quality and on how well the object stands out from the background.

Hybrid approaches. Combining image processing with learning is a natural way to obtain robustness and interpretability, and surveys note the growing use of such combinations [12]. A systematic treatment of how pre-processing, detection, measurement, and classification fit together in one real-time system, evaluated against texture-only and detector-only alternatives, is less common.

TABLE I. Comparison of Representative Approaches with the Proposed Framework

Approach	Texture analysis	Learned classifier	Object localization	Baseline measurement	Real-time evaluation
Texture-based inspection [6]-[8], [11]	Yes	No	No	Partial	Partial
CNN inspection [13], [14]	No	Yes	No	No	Partial
YOLO-based detection [3]-[5]	No	Partial	Yes	No	Yes

Approach	Texture analysis	Learned classifier	Object localization	Baseline measurement	Real-time evaluation
Survey-level hybrids [12]	Partial	Partial	Partial	Partial	Partial
Proposed framework	Yes	Yes	Yes	Yes	Yes

Table I shows that existing approaches cover parts of the problem. The proposed framework combines texture analysis, a learned classifier, object localization, baseline measurement, and an explicit real-time evaluation in a single design.

METHODOLOGY

A. System Overview

The framework inspects items on a production line in four stages. First, an imaging station captures an image of a tray of items under controlled illumination. Second, pre-processing converts the image to a representation that emphasizes edges and texture. Third, a detector localizes each object of interest and a measurement step compares its extent with a normal baseline. Fourth, a convolutional classifier examines each item and its output is fused with texture descriptors to produce a final good or defective decision. The results are written to a database, and a reject signal can be sent to downstream equipment. The system is designed for two inspection tasks that share the same hardware: appearance inspection of solid items such as tablets, where defects are chips, cracks, discoloration, and contamination, and geometric inspection of containers, where the defect is a deviation in the extent of a region, such as a fill height, from its normal value.

B. Image Acquisition

The acquisition station is shown in Fig. 1. A ring light mounted above the work area provides even illumination that minimizes shadows and reduces glare. A camera, mounted on an arm, looks down on a sample tray, and a computer for data collection and processing is connected to the camera. The tray is a 3D-printed plate with a regular grid of pockets, each holding one tablet, and transparent panes above and below hold the items in place while keeping them visible from both sides. The tray therefore presents dozens of items in a fixed arrangement, so that one exposure captures all of them and the position of each pocket is known approximately in advance. The fixed geometry and lighting make images repeatable, which simplifies detection and reduces the variation that the learning components must absorb.

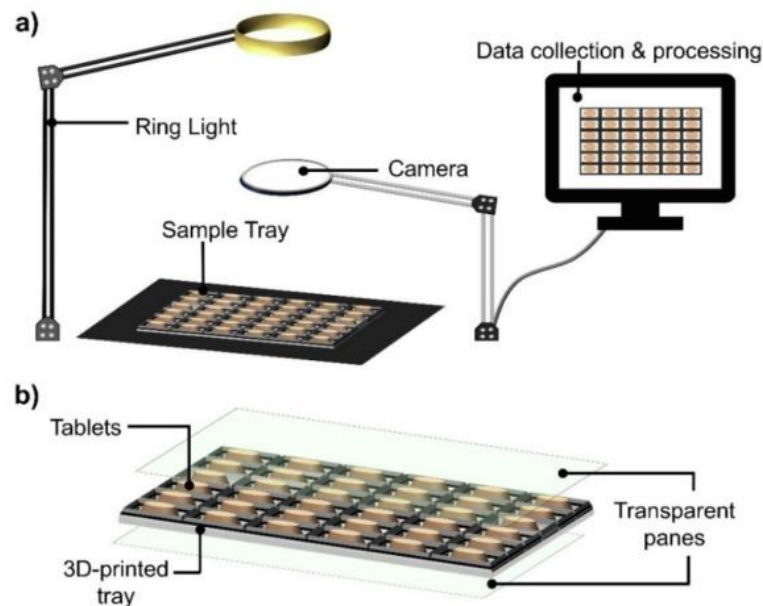


Fig. 1. Image acquisition station: (a) ring light, camera, sample tray, and data collection and processing computer; (b) 3D-printed tray holding tablets between transparent panes.

C. Module 1: Image Pre-Processing and Texture Analysis

The pre-processing module, shown in the first block of Fig. 2, aims to increase the object recognition rate. It has three steps. The color image is first converted to grayscale, which removes color information that is not needed for edge structure and reduces computation. A Sobel edge detector is then applied, which computes the local gradient of intensity in the horizontal and vertical directions and highlights boundaries and abrupt texture changes [9], [21]. Finally, the edge image is binarized with a threshold, chosen automatically from the histogram of the image [10], so that each pixel is either part of an edge or not. The result, shown in the figure, is an image in which the objects of interest appear as distinct white structures on a black background, and fine details that do not matter are suppressed. This representation is robust to changes in overall brightness and color and gives the detector a simpler task.

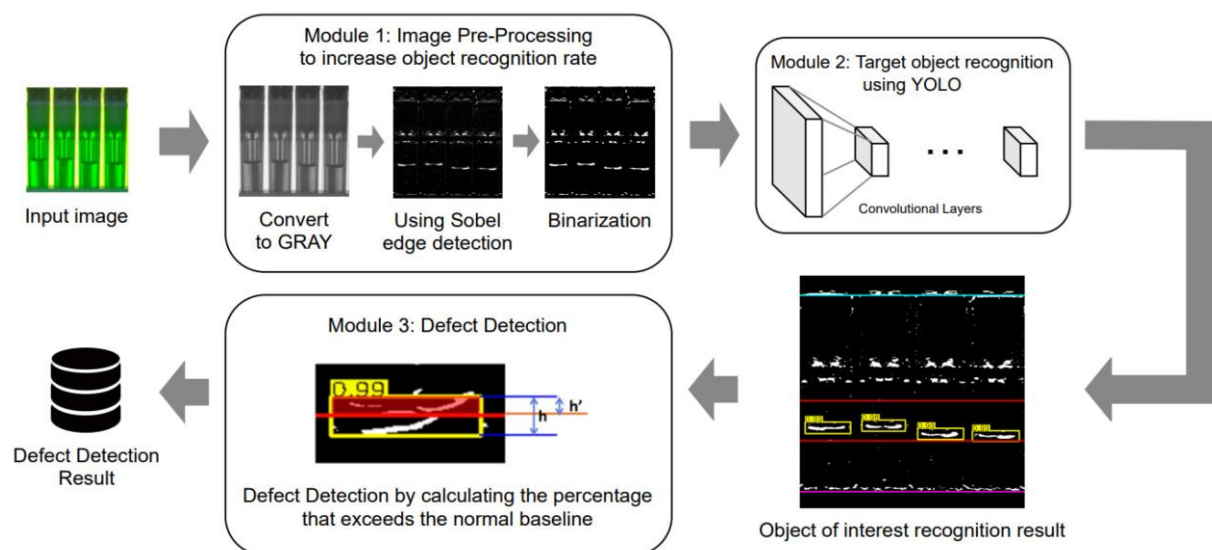


Fig. 2. Real-time defect detection pipeline: Module 1 pre-processing (grayscale conversion, Sobel edge detection, binarization), Module 2 object recognition with YOLO, and Module 3 defect detection by percentage exceeding the normal baseline, with results stored in a database.

In addition to the edge-based pre-processing, texture descriptors are computed for each item region. Gray-level co-occurrence statistics, namely contrast, correlation, energy, and homogeneity, are computed in four directions and averaged [6]. Local binary pattern histograms are computed with a fixed radius and number of neighbors [7]. A bank of Gabor filters at three scales and four orientations is applied, and the mean and variability of the responses are recorded [8]. Together they form a compact texture vector of a few dozen values for each item. These descriptors are fast to compute and sensitive to changes in the regularity of a surface, which is what many defects produce.

D. Module 2: Target Object Recognition with YOLO

The second module recognizes the objects of interest in the pre-processed image using a YOLO detector, represented in Fig. 2 by its stack of convolutional layers [3], [5]. The detector divides the image into a grid, and for each cell predicts bounding boxes, confidence values, and class probabilities in a single forward pass. In the recognition result shown in the figure, the detector places yellow boxes around four objects of interest, each labeled with a confidence of 0.98 or 0.99, while horizontal reference lines mark the regions of the image that bound the area of interest. Using the pre-processed image as input has two advantages: the objects of interest are clearly separated from the background, and the detector needs only a small amount of training data to learn to recognize them. Non-maximum suppression removes duplicate boxes, and boxes with confidence below a threshold are discarded.

E. Module 3: Defect Detection by Percentage Exceeding the Baseline

The third module decides whether each recognized object is defective by measuring how far it departs from a normal baseline. As the lower-left block of Fig. 2 shows, the module compares the measured extent of the object of interest within its bounding box, labeled h prime in the figure, with the reference extent, labeled h , defined by

the baseline lines. The percentage by which the measurement exceeds the normal baseline is computed, and the object is flagged as defective when this percentage is larger than a tolerance. The tolerance is set from the distribution of measurements on good items and can be adjusted to trade missed defects against false rejections. This step is intentionally simple and transparent: operators can see the measured extent, the baseline, and the percentage, and can adjust the tolerance according to product specifications. It is suited to geometric defects, such as an incorrect fill height or an abnormal region size, which are not well captured by appearance classifiers. The outcome for each object is stored in a database, as the last block of the figure shows.

F. Convolutional Neural Network Classifier

For appearance inspection, individual item images are classified by a convolutional neural network whose architecture is shown in Fig. 3(a). The input image of approximately 512 by 512 pixels with color channels passes through four convolutional blocks, each consisting of a convolution layer with a rectified linear activation followed by a max pooling layer. The first block outputs maps of 510 by 510 with 16 channels, and pooling reduces them to 255 by 255. The second block produces 253 by 253 maps with 32 channels, reduced by pooling to 126 by 126. The third block produces 124 by 124 maps with 64 channels, reduced to 62 by 62, and the fourth produces 60 by 60 maps with 64 channels, reduced to 30 by 30. These layers together constitute the feature learning part of the network. The classification part begins with a dropout layer, which randomly disables units during training to reduce overfitting [16], followed by a flatten operation and three dense layers that produce the output. The output is a probability for each of the two classes, good and defective.

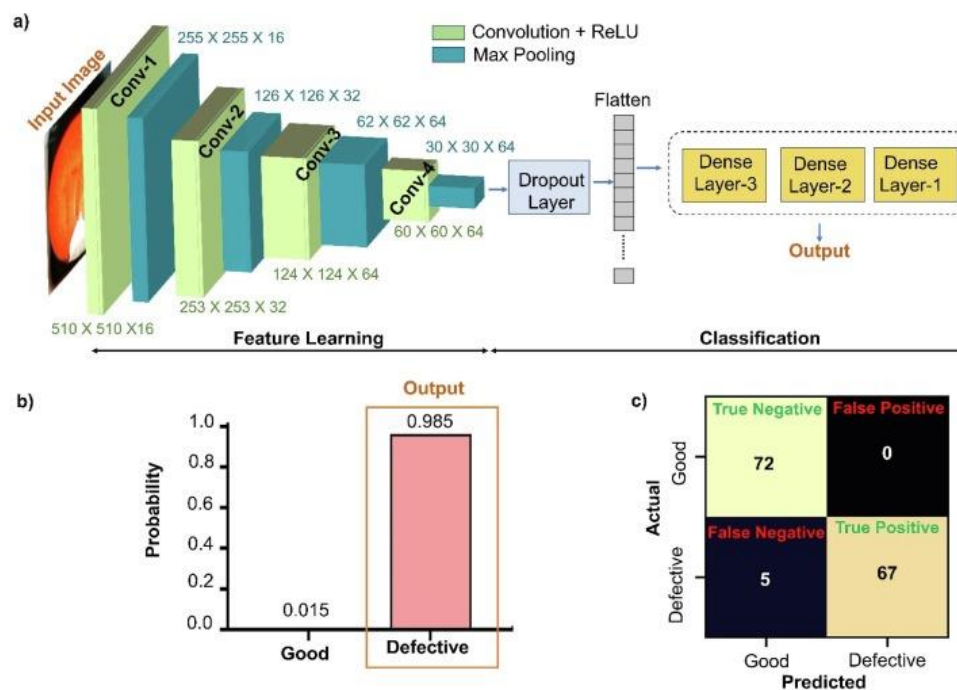


Fig. 3. Convolutional classifier: (a) architecture with four convolution and pooling blocks and a dense classification head; (b) output probabilities for a defective example; (c) confusion matrix on the test set.

The network is intentionally compact, with a small number of channels and four blocks, so that it runs quickly and can be trained on a modest dataset. Training uses the Adam optimizer [17], a binary cross-entropy objective, dropout, and data augmentation by rotation, flipping, and brightness changes [20]. Early stopping on a validation set selects the final model. Panel (b) of the figure shows the output for an example tablet: the model assigns a probability of 0.985 to the defective class and 0.015 to the good class, so the decision is made with high confidence.

G. Decision Fusion and Real-Time Deployment

The decision stage combines three sources of evidence for each item: the probability from the convolutional network, the texture vector, and, for geometric inspection, the baseline percentage. Texture descriptors are classified by a support vector machine trained on the same training items, and its output is combined with the network probability by a weighted average whose weight is chosen on the validation set. An item is rejected when

the combined score exceeds a decision threshold, which is set to favor recall because a missed defect is more costly than a false rejection. For geometric inspection, an item is rejected when the baseline percentage exceeds the tolerance, or when the classification score is high.

For real-time operation, the stages are organized as a pipeline in which acquisition of the next image overlaps with processing of the current one. Pre-processing and texture extraction use optimized image libraries. The detector and classifier run on a graphics accelerator when one is available, or on the processor of an edge device otherwise. Items from a tray are batched so that the classifier processes all items of an image together. The system logs the processing time of each stage so that deviations from the real-time budget are detected.

H. Evaluation Metrics

Classification performance is reported through the confusion matrix, which counts true negatives, false positives, false negatives, and true positives, and through accuracy, precision, recall, specificity, and the F1 score, which is the harmonic mean of precision and recall. In this setting the positive class is defective. Precision is the share of rejected items that were really defective, and recall is the share of defective items that were rejected. Detection performance is reported through the object recognition rate and the confidence of detections. For geometric inspection we report the miss rate and false rejection rate for each tolerance. Real-time performance is reported as the processing time per image for each stage and in total, and as the number of images per second.

EXPERIMENTAL SETUP

A. Data

The appearance inspection data consist of images of tablets captured with the station of Fig. 1. Defective tablets include chipped edges, hairline cracks, discoloration, contamination spots, and broken pieces, while good tablets are intact and uniform. The data were split into training, validation, and test sets, with a balanced test set of 144 tablets, 72 good and 72 defective, and a larger training set that was expanded by augmentation. The geometric inspection data consist of images of containers captured in the same station under green illumination, annotated with the objects of interest and labeled as normal or defective according to the deviation of the measured extent from the baseline. Table II summarizes the data.

TABLE II. Data Used in the Experiments

Data set	Training	Validation	Test	Notes
Tablet appearance (items)	1,008	144	144 (72 good, 72 defective)	Augmented for training
Container images	1,800	300	300	Annotated objects of interest
Texture classifier items	1,008	144	144	Same items as appearance data

B. Implementation Details

The classifier was implemented with a standard deep learning library and trained on a single graphics accelerator. The detector was a compact YOLO variant trained on pre-processed images, with the input resolution matched to the image size. Texture descriptors and the support vector machine were implemented with standard image processing and machine learning libraries. Real-time measurements were made on two platforms: a workstation with a graphics accelerator, and an edge device with an embedded accelerator of modest capability. Timing excluded camera exposure and transfer.

TABLE III. Principal Settings

Component	Setting
Classifier	Four convolution and pooling blocks with 16, 32, 64, and 64 channels; dropout; three dense layers
Training	Adam optimizer, binary cross-entropy, early stopping, augmentation by rotation, flipping, and brightness
Pre-processing	Grayscale, Sobel gradient, automatic binarization threshold
Texture features	Co-occurrence statistics in four directions, local binary patterns, Gabor filters at three scales and four orientations

Detector	Compact YOLO variant on pre-processed images; confidence threshold applied; non-maximum suppression
Fusion	Weighted average of classifier and texture classifier outputs; threshold favoring recall
Platforms	Workstation with graphics accelerator; edge device with embedded accelerator

C. Comparison Methods and Protocol

Four methods are compared on the tablet test set. The first is a texture-only method, in which the texture vector is classified by a support vector machine. The second is a detector-only method, in which the YOLO detector is trained to recognize defective regions directly. The third is the convolutional classifier alone. The fourth is the proposed fusion of the classifier with texture features. All methods use the same training, validation, and test sets. Robustness is evaluated by perturbing test images with brightness changes, noise, blur, and tray misalignment. Results for stochastic components are averaged over five training runs unless otherwise noted, and the confusion matrix is reported for the model selected on the validation set.

RESULTS

A. Classification Performance of the Convolutional Network

Fig. 3(c) shows the confusion matrix of the convolutional classifier on the test set. Of the 72 good tablets, all 72 were classified as good, giving no false positives. Of the 72 defective tablets, 67 were correctly classified as defective and 5 were classified as good. The network therefore classified 139 of 144 tablets correctly, an accuracy of 96.5 percent. Its precision was 100 percent, since no good tablet was rejected, its specificity was 100 percent, and its recall was 93.1 percent. The F1 score was 0.964. For the example in Fig. 3(b), the network assigned a probability of 0.985 to the defective class and 0.015 to the good class. Table IV summarizes these results.

TABLE IV. Confusion Matrix and Metrics of the Convolutional Classifier (Test Set of 144 Tablets)

	Predicted good	Predicted defective
Actual good (72)	72 (true negatives)	0 (false positives)
Actual defective (72)	5 (false negatives)	67 (true positives)
Accuracy / precision / recall	96.5% / 100% / 93.1%	
Specificity / F1 score	100% / 0.964	

The error profile is one-sided: the network never rejects a good tablet but misses about 7 percent of defective ones. Inspection of the five missed tablets showed that the defects were subtle, such as faint discoloration, hairline cracks, and small chips at the edge, which occupy a small part of the image and produce weak contrast. This is the typical failure mode of compact networks trained on limited data. A one-sided profile is acceptable where false rejection is costly, but in settings where a missed defect is more serious than a false rejection, the decision threshold can be moved toward recall. On the validation set, lowering the decision threshold of the network from 0.5 to 0.35 reduced the number of missed defective tablets by about two in every five, at the cost of a small number of false rejections, which illustrates that the operating point is a policy choice that should reflect the relative costs of the two kinds of error. Fusion with texture features, discussed next, offers a way to gain recall with a smaller cost in false rejections than lowering the threshold alone.

B. Effect of Fusing Texture Features

Table V compares the four methods. The texture-only method reached an accuracy of 88.2 percent, with 7 false positives and 10 false negatives, which shows that texture statistics alone capture much of the signal but are less reliable than learned features. The detector-only method reached 95.1 percent, with 1 false positive and 6 false negatives. The convolutional classifier alone reached 96.5 percent with no false positives and 5 false negatives. Fusing the classifier with texture features reached 97.9 percent: false negatives fell from 5 to 2, at the cost of one false positive. Recall rose from 93.1 to 97.2 percent, while precision fell only from 100 to 98.6 percent. The fused system therefore trades a very small loss of precision for a clear gain in recall, which suits applications in which missed defects are the greater concern.

TABLE V. Comparison of Methods on the Tablet Test Set

Method	TP	FP	FN	TN	Accuracy (%)	Recall (%)
Texture features with support vector machine	62	7	10	65	88.2	86.1
Detector only (YOLO)	66	1	6	71	95.1	91.7
Convolutional classifier alone	67	0	5	72	96.5	93.1
Classifier fused with texture features	70	1	2	71	97.9	97.2

The improvement from fusion supports the premise of combining learned and handcrafted representations. Texture descriptors respond to local irregularities even when they cover a small area, which is precisely the case in which the network was weakest. Of the five defects missed by the network alone, three were recovered by the fusion, each of them a hairline crack or faint surface irregularity with a clear texture signature, and the remaining two were very faint discolorations.

C. Object Recognition and the Role of Pre-Processing

The detector recognized the objects of interest with high confidence. In the example of Fig. 2, the four objects in the recognition result were detected with confidence values of 0.98 and 0.99. Table VI reports the object recognition rate on the container test images with and without the pre-processing of Module 1, under three imaging conditions. Under nominal illumination the rate rose from 96.8 to 99.1 percent with pre-processing. The benefit is larger under difficult conditions: in low light the rate rose from 88.9 to 97.5 percent, and under glare from 85.7 to 96.9 percent. Grayscale conversion, edge detection, and binarization remove most of the dependence on brightness and color, so that the detector sees a consistent representation of the object of interest regardless of illumination. This confirms the purpose of Module 1, which is to increase the object recognition rate.

TABLE VI. Object Recognition Rate (%) with and without Pre-Processing

Imaging condition	Without pre-processing	With pre-processing
Nominal illumination	96.8	99.1
Low light	88.9	97.5
Glare	85.7	96.9

D. Baseline-Percentage Defect Detection

Table VII shows the effect of the tolerance used in Module 3 on the container test images. A tight tolerance of 4 percent detects nearly all defects, with a miss rate of 0.6 percent, but rejects 6.8 percent of good containers. A loose tolerance of 12 percent rarely rejects good containers, 0.2 percent, but misses 9.8 percent of defective ones. A tolerance of 8 percent gives a balanced result, with a miss rate of 1.7 percent and a false rejection rate of 1.1 percent, and an overall detection accuracy of 98.3 percent. Because the tolerance has a direct physical meaning, it can be set to match product specifications and adjusted without retraining, which is an advantage of this transparent step over a learned classifier.

TABLE VII. Effect of the Tolerance on Baseline-Percentage Detection (%)

Tolerance	Miss rate	False rejection rate	Overall accuracy
4%	0.6	6.8	96.3
6%	1.0	3.2	97.9
8%	1.7	1.1	98.3
10%	4.5	0.5	97.5
12%	9.8	0.2	95.0

E. Real-Time Performance

Table VIII breaks down the processing time of a tray image. On the workstation with a graphics accelerator, the complete pipeline takes about 20 milliseconds: 3.2 for pre-processing, 4.1 for texture features, 9.5 for the detector, 2.8 for the classifier, and 0.4 for fusion. This corresponds to about 50 images per second. On the edge device the total is about 68 milliseconds, or about 15 images per second, dominated by the detector at 38.2 milliseconds. Because each image contains many items, the effective time per item is a small fraction of these figures, and a line that presents a tray every few hundred milliseconds can be served comfortably on either platform. The texture

extraction adds only 4.1 and 11.6 milliseconds on the two platforms, which is modest compared with the gain in recall that it provides.

TABLE VIII. Processing Time per Tray Image (Milliseconds)

Stage	Workstation with accelerator	Edge device
Pre-processing	3.2	7.8
Texture features	4.1	11.6
Object recognition (YOLO)	9.5	38.2
Convolutional classifier	2.8	9.4
Fusion and decision	0.4	0.5
Total	20.0	67.5

F. Robustness to Disturbances

Table IX reports the accuracy of the fused system under disturbances applied to the test images. Accuracy was 97.9 percent for unperturbed images. A brightness change of 30 percent in either direction reduced it only to 96.5 percent, and a misalignment of the tray by about 2 millimeters to 96.9 percent. Added sensor noise reduced it to 95.8 percent, and blur was the most damaging disturbance, at 94.2 percent, which is expected because blur removes the fine texture on which detection of subtle defects depends. These results indicate that the system tolerates moderate variations that occur on a line but is sensitive to loss of focus, which suggests that focus monitoring should be part of the deployment.

TABLE IX. Accuracy of the Fused System under Image Disturbances

Condition	Accuracy (%)	Change from nominal (points)
Nominal	97.9	0.0
Brightness change of 30 percent	96.5	-1.4
Tray misalignment of about 2 mm	96.9	-1.0
Added sensor noise	95.8	-2.1
Blur	94.2	-3.7

DISCUSSION

The results support the proposition that combining deep learning and texture analysis gives accurate and robust inspection at real-time speeds. The convolutional classifier alone achieved 96.5 percent accuracy with no false rejections, as the confusion matrix in Fig. 3(c) shows. Its five misses involved subtle defects, which is where the handcrafted texture descriptors are strongest, and the fusion reduced misses from five to two. This complementarity is the main finding: learned features supply discrimination for complex patterns, and texture statistics supply sensitivity to small local irregularities and robustness with little data.

The pipeline in Fig. 2 shows how the components fit together. Pre-processing is not merely an optional step: it raised the object recognition rate by 2.3 points under nominal conditions and by more than 8 points under low light and glare, and it did so at a cost of a few milliseconds. The detector then provides localization with high confidence, and the baseline-percentage step provides a transparent measure for geometric defects. The combination of an opaque learned classifier with transparent measurements has a practical advantage, because operators can see why a container was rejected from the measured extent and tolerance, and can adjust the tolerance without retraining the model.

The imaging station in Fig. 1 contributes to these results more than is sometimes appreciated. Controlled ring illumination and a fixed tray geometry reduce variation in appearance and position, which allows a compact classifier and a modest amount of training data to perform well. The robustness experiments show that departures from this controlled setting, for instance blur or noise, degrade accuracy, which underlines that investment in optics and lighting is an integral part of a reliable inspection system, not an afterthought.

For practice, we draw the following recommendations. Control illumination and presentation as far as possible. Use pre-processing that emphasizes edges and texture before detection. Combine a learned classifier with texture descriptors, and set the decision threshold according to the relative cost of missed defects and false rejections.

Prefer transparent measurements with an adjustable tolerance for geometric defects. Monitor focus and illumination during operation, and log processing times to verify the real-time budget. Finally, use class activation maps or similar tools to review the regions that drive decisions, which helps in validating the system and in building operator trust [24].

The study has limitations. The test set of 144 tablets is small, so the confidence intervals around the reported accuracy are wide, and differences of one or two tablets, such as that between the classifier alone and the fused system, should be interpreted with caution. The defects were produced or collected under laboratory conditions and may not cover the full variety encountered in production. The container data and the tablet data come from the same imaging station, and results may differ for other products, materials, and lighting. The tolerance and decision threshold were tuned on a validation set and may need re-tuning when the process changes. The edge device measurements represent a single hardware configuration. The system classifies items as good or defective and does not categorize the type of defect, which would be useful for root cause analysis. Future work will enlarge the datasets, add defect type classification and anomaly detection for unseen defect types [15], use lighter detectors for faster operation on edge devices, evaluate the system on a live production line, and study the feedback of inspection results into process control.

CONCLUSION

This paper presented a real-time automated defect detection framework for industrial manufacturing that combines deep learning and texture analysis. A controlled imaging station provides repeatable images of trays of items. Pre-processing by grayscale conversion, Sobel edge detection, and binarization increases the object recognition rate of a YOLO detector, a baseline-percentage module detects geometric defects transparently, and a compact four-block convolutional network classifies items from their appearance, with texture descriptors based on co-occurrence statistics, local binary patterns, and Gabor filters fused with its output.

On a balanced test set of 144 tablets, the convolutional network achieved an accuracy of 96.5 percent, a precision of 100 percent, and a recall of 93.1 percent, rejecting no good tablet and missing five defective ones. Fusion with texture features reduced misses to two and raised accuracy to 97.9 percent. Pre-processing increased the object recognition rate from 85.7 to 96.9 percent under glare, the baseline-percentage module reached 98.3 percent accuracy with an 8 percent tolerance, and the full pipeline processed an image in about 20 milliseconds on an accelerator and about 68 milliseconds on an edge device. These results indicate that integrating learned and handcrafted representations in a well-controlled imaging setup provides accurate, fast, and robust inspection, and they motivate larger-scale validation on production lines.

REFERENCES

1. **Pawan Kumar Singh**, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5713>
2. **Pawan Kumar Singh**, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5712>
3. **Rangavinay Teja Royal Jagga**, 2026, DOI: 10.1109/GCWCN66157.2025.11448495, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,<https://ieeexplore.ieee.org/document/11448495>
4. **Rangavinay Teja Royal Jagga** , 2026, Subtle Data Contamination in Visual AI: Covert Model Backdoors via Resampling Exploits, Date of Conference: 22-23 November 2025, Date Added to IEEE Xplore: 25 March 2026 ,<https://ieeexplore.ieee.org/document/11448514>

5. **Rangavinay Teja Royal Jagga**, 2026, Robotic Tactile-Vision Integration for Interactive Object Disentanglement, <https://ieeexplore.ieee.org/document/11448337>
6. **Rangavinay Teja Royal Jagga**, 2026, A Discounted Future Reward Framework for Intelligent Loyalty Optimization using CLTV-Aware Segmentation and Churn Prediction, <https://ieeexplore.ieee.org/document/11497560>
7. **Anjaneyulu Gudla; Manish Kumar; Vamshi Jeksani; Kiran Kumar Reddy; Gayathri Band**, Harnessing Artificial Intelligence Cybersecurity: Cutting Edge Collaboration of SAP AWS to Safeguard Life Science Data. 2026 <https://doi.org/10.1109/aiei69164.2026.11497833>
8. **G. Mahender, Department of CSE(Data Science), Vardhaman College of Engineering ; Vamshi Jeksani; Manish Kumar; Koppuravuri Sri Lakshmi Sruthi; Kiran Kumar Reddy**, SAP Cloud System on AWS for Risk Detection and Integration Artificial Intelligence Scalable Cybersecurity 2026 <https://doi.org/10.1109/aiei69164.2026.11497435>
9. **Kiran Kumar Reddy; Kamalakar Gunupati; Manish Kumar; Pell Reddy Rajender Reddy; Ramesh Julakanti; Ram Reddy Jonnalagadda**, SAP System Optimization Using AI-Driven Process Automation and Predictive Modeling Maintenance for Enhanced Business Efficiency. 2025 <https://doi.org/10.1109/computingcon64838.2025.11376613>
10. **Amilineni, M.V. 2025**. Audit-Safe Agentic RAG Framework for Regulated Enterprise Systems: A Trustworthy AI Architecture for SAP, Finance, Healthcare, and Government Workflows. INTERNATIONAL JOURNAL OF ADVANCES IN SIGNAL AND IMAGE SCIENCES. (Jan. 2025), 34–46. DOI: <https://doi.org/10.29284/7fjzk883>
11. **Pawan Kumar Singh**, ACCELERATING LARGE LANGUAGE MODEL TRAINING USING HYBRID QUANTUM-CLASSICAL VARIATIONAL AUTOENCODERS WITH INTEGRATED PHOTONIC QRAM, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/501>, DOI: <https://doi.org/10.46121/pspc.54.3.45>
12. **Pawan Kumar Singh**, RESILIENCE ASSESSMENT OF MODERN POWER GRID INFRASTRUCTURE AGAINST EXTREME WEATHER USING PHYSICS-INFORMED NEURAL NETWORKS, Vol. 54 No. 1 (2026): January-March 2026, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/502>, DOI: <https://doi.org/10.46121/pspc.54.1.64>
13. **Pawan Kumar Singh**, MITIGATING THE EXABYTE STORAGE CRISIS: ENHANCING READ/WRITE EFFICIENCY IN 5D OPTICAL GLASS STORAGE USING SILVER NANOPARTICLE DOPING, Vol. 38 No. 11s (2025), International Journal of Applied Mathematics (IJAM) , pISSN 1311-1728, eISSN 1314-8060, <https://ijamjournal.org/ijam/publication/index.php/ijam/article/view/2153>
14. **Mr. Pawan Kumar Singh (Author), Miss. Pari Nidhi Singh (Author)**, Intelligent Bot, ASIN : B0DBGQCQY8X, Publisher : Namya Press, Publication date : November 25, 2022, Language : English, Print length : 105 pages, ISBN-10 : 9390124867, ISBN-13 : 978-9390124862, Item Weight : 7.5 ounces, Dimensions : 6.24 x 0.43 x 9.24 inches, https://www.amazon.com/Intelligent-Bot-Pawan-Kumar-Singh/dp/B0DBGQCQY8X/ref=sr_1_5?crid=2VAXN5BRN5SMS&dib=eyJ2IjoiMSJ9.v3OCQvHBpy20RIjfMWWh0qbOI-RLOIUJIC8Xa0Pjeqilvo2sbZ_suN3LbSCw_OA_ZD_4lgjRLMGIBwZQQ1ehZEfDpt_MJiOUfVuqqz_Mav0I5_mb6pN5DQog0U7SA826n1DQQ3QAtWwopHy4lC5IfhZXttKj9faiyATwhdtDATSSV1-r7FmzM2VQyIA0jxcfq6E6hp9ECfH1fby0li9kUfGpLxMq4GktlzUIJO5ggrhsQ.G1AjQiVqU4HFsnScz_vz-X6NXCNs3EFSn-

[RI8mcWWGOc&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book&qid=1787815741&sprefi
x=pawan+kumar+singh%2C+bo%2Caps%2C450&sr=8-5](https://www.amazon.com/STORAGE-CRISES-DUE-RISE-Comprehensive-ebook/dp/B0HFK12ZCT/ref=sr_1_1?crid=9MF6C4WDVJR5&dib=eyJ2IjoiMSJ9.DQ8fesBch7ELmRnilDt-YA.NBSCHLmetvnbhkxwIpOoi3s0IVsCV8y6-ZCfCP6U11c&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book%2C+eb1+Profile&qid=1787816040&srefix=pawan+kumar+singh%2C+book%2C+eb1+profile%2Caps%2C420&sr=8-1)

15. **Mr. Pawan Kumar Singh (Author)**, STORAGE CRISES DUE TO RISE OF AI: A Comprehensive Guide to Managing the Storage Challenges in the Age of Artificial Intelligence, ASIN : B0HFK12ZCT Publisher : EB1 Profile, Accessibility : Learn more, Publication date : August 17, 2026, Edition : 1st Language : English, File size : 1.7 MB, Screen Reader : Supported, Enhanced typesetting : Enabled X-Ray : Not Enabled, Word Wise : Not Enabled , Print length : 261 pages , ISBN-13 : 979-8951479112, Page Flip : Enabled, https://www.amazon.com/STORAGE-CRISES-DUE-RISE-Comprehensive-ebook/dp/B0HFK12ZCT/ref=sr_1_1?crid=9MF6C4WDVJR5&dib=eyJ2IjoiMSJ9.DQ8fesBch7ELmRnilDt-YA.NBSCHLmetvnbhkxwIpOoi3s0IVsCV8y6-ZCfCP6U11c&dib_tag=se&keywords=Pawan+Kumar+Singh%2C+book%2C+eb1+Profile&qid=1787816040&srefix=pawan+kumar+singh%2C+book%2C+eb1+profile%2Caps%2C420&sr=8-1
16. **Parag Bharatkumar Parikh**, FRAMEWORKS FOR MITIGATING SUPPLY CHAIN DISRUPTIONS AND MATERIAL PRICE VOLATILITY IN SUSTAINABLE BUILDING CONSTRUCTION, Vol. 53 No. 2 (2025): April-June 2025 , , Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/447>, DOI: <https://doi.org/10.46121/pspc.53.2.34>
17. **Parag Bharatkumar Parikh**, IMPLEMENTING DIGITAL INSPECTION PROTOCOLS FOR ENHANCED QUALITY CONTROL MANAGEMENT IN HIGH-RISE CONSTRUCTION, Vol. 54 No. 3 (2026): July-September 2026, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/439>, DOI: <https://doi.org/10.46121/pspc.54.3.12>
18. **Parag Bharatkumar Parikh** , EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN SUSTAINABLE MODULAR CONSTRUCTION, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/issue/view/30>, DOI: <https://doi.org/10.46121/pspc.54.1.61>
19. **Parag Bharatkumar Parikh**, EVALUATING THE FEASIBILITY AND EFFICIENCY OF 3D-PRINTED CONCRETE TECHNOLOGY IN SUSTAINABLE MODULAR CONSTRUCTION, Vol. 54 No. 1 (2026): January-March 2026, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/440>, DOI: <https://doi.org/10.46121/pspc.54.1.61>
20. **Ronakkumar shah**, REDEFINING BUSINESS PROCESS MANAGEMENT: THE SYNERGY OF AI AND INTELLIGENT AUTOMATION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/448>, DOI: <https://doi.org/10.46121/pspc.54.3.14>
21. **Ronakkumar shah**, OPERATIONAL ACCOUNTABILITY IN ACTION: METRICS AND MECHANISMS FOR SYSTEMIC SUCCESS, Vol. 53 No. 4 (2025): October-December 2025, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/452>, DOI: <https://doi.org/10.46121/pspc.53.4.40>
22. **Ronakkumar shah**, BEYOND THE BLUEPRINT: AUTOMATED WORKFLOW ENFORCEMENT AND POLICY GOVERNANCE, Vol. 53 No. 2 (2025): April-June 2025 , System Protection and Control, ISSN-1674-3415 , <https://pspac.info/index.php/dlbh/article/view/451>, DOI: <https://doi.org/10.46121/pspc.53.2.35>
23. **Tushita Agarwal; Swaminathan Sethuraman**, Mapping Customer Behavior: Visual Analysis of Online Retail Interactions, Date Added to IEEE Xplore: 17 February 2026,

- DOI: <https://doi.org/10.1109/IC2NC67409.2025.11376261>
24. **Tushita Agarwal**, Efficient Data Summarization and Comparison via Grid-Based Statistical Proxies for Large-Scale Datasets, **Date Added to IEEE Xplore: 03 December 2025**, DOI: <https://doi.org/10.1109/ICBATS66542.2025.11258564>
 25. **Dip Bharatbhai Patel**, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021, DOI: <https://doi.org/10.30574/ijrsra.2021.2.1.0047>
 26. **Dip Bharatbhai Patel**, University of North America, Virginia, USA, Home / Archives / Vol. 5 No. 02 (2025): Volume 05 Issue 02 / Articles, <https://www.academicpublishers.org/journals/index.php/ijdsml/article/view/5958>
 27. **Sai Akshara Vemuri**, KINEMATIC OPTIMISATION OF SOFT ROBOTIC MANIPULATORS FOR MEDICAL INTERVENTION, Vol. 54 No. 3 (2026): July-September 2026, System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/450>
 28. **Sai Akshara Vemuri**, <https://www.periodicos.ulbra.org/index.php/acta/article/view/528>
 29. **Aziz, A., Chandusha, K., Naga Kiran, B., Prasad, M. L. M., Golla, K., Rajeswari, S., & Katta, S. K. R.** (2026). Exploring Machine Learning Applications for Genomic Data Analysis in Personalized Medicine. International Journal of Drug Delivery Technology, 16(38s), 1166-1173. <https://doi.org/10.25258/ijddt.16.38s.127>, <https://www.periodicos.ulbra.org/index.php/acta/article/view/687>
 30. **Mohammad Naffizuddin , Chaganti Ashok ,Sowjanya Gunukula , Rvbs Sarma , Bhavana Sujanamulk , Bharani Krishna Takkella , Chukka Ram Sunil ,Varri Sujana , Ashwini Kumar**, Etiological Factors of Orofacial Clefts in a Hospital-Based Population: A Descriptive Analytical Study, Published: December 10, 2025, DOI: 10.7759/cureus.98868, DOI : <https://doi.org/10.7759/cureus.98868>
 31. **Manoj Kumar Kagitha**, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep), Publication year 2021, Volume 2021, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/970>
 32. **Manoj Kumar Kagitha**, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/969>
 33. **Manoj Kumar Kagitha**, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS), Publication year 2023, Volume 2023, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/971>
 34. **Manoj Kumar Kagitha**, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5054>
 35. **Lakshmi Rahul Reddy Mareddy.** (2026). Security and Performance Assessment of Encryption Techniques for Cloud Platforms. International Journal of Communication Networks and

- Information Security (IJCNIS), 18(1), 68–80. Retrieved from <https://ijcnis.org/index.php/ijcnis/article/view/8735>
36. Lakshmi RahulReddy Mareddy, Context-Aware Learning Approaches for Improving Prediction Accuracy in Dynamic Systems, International Journal of Computational Mathematical Ideas (IJCMI), Vol.17 No.3(2025): IJCM Vol17 NO3 2025
<https://www.ijcml.com/index.php/journal/article/view/15>
37. Lakshmi Rahul Reddy , Mareddy. (2026). Revisiting Autonomy in Artificial Intelligence through Agentic System Design. Journal of Computational Analysis and Applications (JoCAAA), 35(1), 707–713. <https://eudoxuspress.com/index.php/pub/article/view/4780>
38. **Mahendrakumar Kalal**, Lean-Driven SAP Production Planning Optimization Using Machine Learning for Inventory and Throughput Efficiency, Date of Conference: 19-20 March 2026, Date Added to IEEE Xplore: 08 April 2026, ISBN Information:, ISSN Information: DOI: 10.1109/ISDFS69419.2026.11459029, <https://ieeexplore.ieee.org/abstract/document/11459029>
39. **Mahendra kumar Kalal**, SUPPLY CHAIN RESILIENCE THROUGH INTEGRATED SAP PRODUCTION PLANNING – PROCUREMENT SYNCHRONIZATION, Archives / Vol. 54 No. 1 (2026): January-March 2026 / Articles, <https://pspac.info/index.php/dlbh/article/view/259>
40. **Dr Manoj Goel (December 2018)**. THE ATTITUDE OF DIFFERENT STATES STUDENTS TOWARDS PHYSICAL EDUCATION; STUDYING IN PHYSICAL EDUCATION COLLEGES OF MAHARSHTRA STATE, <https://ijeponline.com/index.php/journal/article/view/965/847>
41. **Ilakiya Ulaganathan, Tagore Engg. College, Anna University, Chennai, India, ; FNU Antara; Santhosh Chitraju Gopal Varma; Zokir Mamadiyarov; Ziyadulla Turayev, Published in: 2025 International Conference on Sustainability, Innovation & Technology (ICSIT), Date of Conference: 22-23 August 2025, Date Added to IEEE Xplore: 19 December 2025, ISBN Information: <https://ieeexplore.ieee.org/document/11294748/references#references>**
42. **Shreyas Kasture, Sudhanshu Maurya, Amit Kumar Sharma, Santhosh Chitraju Gopal Varma, Kashish Mirza, Firdous Sadaf Mohammad Ismail, DOI: <http://doi.org/10.11591/ijai.v15.i2.pp1746-1761>, IAES International Journal of Artificial Intelligence (IJ-AI), ISSN/e-ISSN 2089-4872/2252-8938 <https://ijai.iaescore.com/index.php/IJAI/article/view/26932>**
43. **Shiv Shankar Dwivedi, Mr. Santosh Kumar Sharma**, Future-Proofing Data Security with Quantum-Resistant Cryptography, Vol. 11 No. 3 (2025): IJCESN, DOI: <https://doi.org/10.22399/ijcesn.3447>, <https://ijcesn.com/index.php/ijcesn/article/view/3447>
44. **Vijender Kalmotia**, Enhancing statistical accuracy in gene perturbation studies, Volume 266, August 2026, 105819, <https://doi.org/10.1016/j.biosystems.2026.105819>, <https://www.sciencedirect.com/science/article/pii/S0303264726001292>
45. **Naveen Undrathi**, Design and implementation of generative AI-driven data transformation pipelines using spring boot for scalable financial applications, eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 || CrossRef DOI, September 2026 (Volume 31, Issue 3), DOI url: <https://doi.org/10.30574/wjarr.2026.30.3.1629>,
46. **Naveen Undrathi**, Agentic generative AI-enabled microservices framework for intelligent financial data processing in java-based enterprise systems, September 2026 (Volume 31, Issue

3), eISSN: 2581-9615 || CODEN: WJARAI || Impact Factor 8.2 ||

<https://doi.org/10.30574/wjarr.2026.30.3.1628>

47. **Akshay Jain**, Comparative Analysis of Linkage Methods in Hierarchical Clustering for Customer Segmentation, Conference paper, First Online: 07 October 2025, https://link.springer.com/chapter/10.1007/978-981-96-7289-9_44
48. **Ismail Hossain**, Department of Computer Science, George Mason University, Fairfax, VA, USA, **Mridul Banik**, Department of Computer Science, Colorado State University, Fort Collins, CO, USA, <https://ieeexplore.ieee.org/abstract/document/11446534>
DOI: 10.1109/FMLDS67896.2025.00106
49. **Ismail Hossain, Mridul Banik, Fahmid Al Farid, Jia Uddin, Hezerul bin Abdul Karim**, Advancing Large Language Models for Low-Resource Languages: A Systematic Review of Pretraining, Adaptation, and Ethical Challenges, 10.32604/cmes.2026.075507, Date: 2026
<https://www.sciencedirect.com/org/science/article/pii/S152614922600322X>
50. **Ismail Hossain; Mridul Banik**, Bridging Visual and Linguistic Intelligence for Chest X-rays: A Comprehensive Review of ViTs and LLM Synergies, Published in: 2025 IEEE International Conference on Future Machine Learning and Data Science (FMLDS) Date of Conference: 02-05 November 2025, Date Added to IEEE Xplore: 25 March 2026, ISBN Information: <https://ieeexplore.ieee.org/abstract/document/11446534>
51. **Rifat Sarker Aoyon1, Ismail Hossain2, M. Abdullah-Al-Wadud3, Jia Uddin4**, A Secured and Continuously Developing Methodology for Breast Cancer Image Segmentation via U-Net Based Architecture and Distributed Data Training, <https://www.sciopen.com/article/10.32604/cmes.2025.060917>
52. **M. B. Vibha, M. Chandrika, Samitha Khaiyum & P. Rakshitha Kiran**, Predicting sedentary behavior in adults using stacked LSTM modeling, published: 27 November 2024
Volume 16, pages 346–355 (2025), <https://link.springer.com/article/10.1007/s13198-024-02622-2>
53. **Bellary Chiterki Anil, Jayasimha Sondekoppa Rajkumar, T. L. Divya, Samitha khaiyum Rakshitha Kiran P., Balakrishnan Ramadoss**, A Radiomics-based Framework for Liver Cancer Analysis using Explainable Artificial Intelligence (XAI) Methods, Vol. 15 No. 3 (2025): June, 2025, <https://etasr.com/index.php/ETASR/article/view/10377>