International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 5, Issue 1, 79-91, 2024
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V5I1P108

Original Article

Deep Learning for Industrial Barcode Recognition at High Throughput

Kiran Kumar Pappula[1], Sunil Anasuri[2]
[1, 2] Independent Researcher, USA.

Abstract *- In modern industrial settings, barcode and QR code recognition play a critical role in automation, inventory tracking, and production line management. High-throughput environments demand fast, accurate and robust recognition systems that can function reliably under varying lighting conditions, orientations, and printing inconsistencies. This paper explores the use of Convolutional Neural Networks (CNNs) and You Only Look Once (YOLO) object detection models for high-throughput industrial barcode recognition. The study highlights benchmark results under diverse environmental parameters such as printing speeds, motion blur, and illumination intensity. We propose a hybrid deep learning architecture that combines the localization efficiency of YOLO with the classification strength of CNNs to optimize both detection speed and recognition accuracy. The dataset consists of synthetically augmented and real-world barcode and QR code images collected from various production environments. Our model is trained using TensorFlow and PyTorch, optimized with advanced loss functions such as focal loss and IoU loss, and benchmarked against traditional OCR and classical computer vision techniques. The results demonstrate a notable improvement in recognition accuracy, throughput rate, and robustness to environmental variability. Our methodology includes data preprocessing techniques such as histogram equalization and affine transformations, combined with training strategies like transfer learning and data augmentation. We achieve a top-1 accuracy of 98.4% and a mean Average Precision (mAP) of 96.2% on our test dataset, even under challenging real-time industrial constraints. This paper also discusses the integration challenges of deep learning systems in legacy manufacturing ecosystems and presents a modular deployment strategy using edge computing devices and IoT gateways. The implications of our research extend to automated logistics, real-time quality inspection, and industrial IoT (IIoT) systems. Future work will focus on improving interpretability, reducing computational load, and extending the system to multilingual and distorted code environments.*

Keywords *- Deep Learning, YOLO, Convolutional Neural Networks (CNN), Barcode Recognition, High Throughput, Industrial Automation, Image Processing, Edge Computing.*

1. Introduction

The introduction of Industry 4.0 has led to the increased use of intelligent devices in manufacturing and logistics, particularly for automation, monitoring, and control within the context of quality assurance. Some of the most essential elements of these smart systems include machine vision, which facilitates the automatic identification and decoding of visual information. [1-4] In this sphere, QR code and barcode identification are the most relevant contributions to the simplification of supply chain processes, stock control, and checking. These codes, which machines can read, are useful for instant detection and tracking of goods within production lines, warehouses, and throughout distribution channels. Nevertheless, due to the increasing complexity of industrial environments and the ongoing increase in production speeds, conventional barcode scanners, which typically take the form of laser scanning or low-resolution imaging-based solutions, struggle to scale to meet these requirements effectively. Legacy systems are prone to producing poor results when conditions include motion blur, low light, partial occlusion, or damaged labels. Conversely, deep learning technologies, especially convolutional neural networks (CNNs) and object detection models like YOLO (You Only Look Once), provide more sophisticated functions in pattern recognition and real-time image processing. They can learn robust visual features, adapt to noisy environments, and provide fast inference, making them ideal for use in contemporary industrial contexts. Deep learning, therefore, could transform the way bar code recognition is done by creating more precise, facile, and scalable systems that can adapt to the needs posed by completely mechanised and networked manufacturing procedures. The creation of these types of systems can be considered within the broader scope of Industry 4.0, where intelligent automation and data-informed decision-making are key factors in enhancing process efficiency and innovation.

1.1. Deep Learning for Industrial Barcode Recognition

The use of deep learning in barcode recognition has become a formidable response to the limitations of traditional procedures. In contrast to rule-based systems, deep learning models have the ability to find intricate patterns in datasets and extrapolate them to vast, dissimilar surroundings in the real world. In this section, the main elements and benefits of deep learning in industrial barcode recognition will be outlined.

Fig 1: Deep Learning for Industrial Barcode Recognition

- **Robust Feature Extraction with CNNs:** Most applications of deep learning, involving images, use Convolutional Neural Networks (CNNs). CNNs are very good at extracting spatial features in barcode recognition, i.e., addressing edge patterns, orientation, and alignment, even when the input image is noisy or distorted. This feature enables them to read barcode formats, such as EAN-13, QR, and DataMatrix, among others, even when they are partially blocked or blurred. Another way that CNNs stand out is in the flexibility they allow in classifying barcodes with different code types. Thus, they are applicable in industrial settings where product labelling protocols vary.
- **Real-Time Detection Using YOLO Architecture:** YOLO (You Only Look Once) is a real-time object recognition system that uses a single forward pass to process a complete picture; this is what makes it very quick and efficient. YOLO, used in barcode recognition, is employed to recognise the precise position of a barcode in an image or frame. Newer variants, such as YOLOv5, are both faster and more accurate, and can be applied to fast conveyor belts or robotic limbs. The grid-type detection system in YOLO offers high-confidence localisation to its users, even in advanced scenes.
- **Advantages over Traditional Methods:** Conventional barcode readers and image processing applications, including edge detection or template matching, have underperformed in harsh environments such as low or dim lighting, poorly contrasting or non-standard barcodes, and non-standard orientations. In comparison, deep learning systems can be trained using a wide range of data and augmented data, making them capable of being resilient in dynamic conditions. These can also be reconfigured to fit different forms of barcodes or other barcodes with minimal training, providing flexibility to meet changing industrial demands.
- **Integration and Scalability in Industry 4.0:** Barcode recognition systems based on deep learning can easily integrate with smart factories, IoT platforms, and automated inspection pipelines. They can operate within real-time parameters and achieve high accuracy rates; this is one of the reasons why they have become a foundational technology in Industry 4.0 settings. Moreover, the current state of these systems (advances in model compression and edge computing in general) is becoming more suitable for being deployed on embedded hardware and mobile devices.

1.2. Problem Statement

Although great advances have already been achieved with deep learning and computer vision, the current barcode recognition systems still face severe drawbacks to their popularity and operational effectiveness in industries. Such problems include scalability, complexity of integration, and sensitivity to environmental conditions.

Fig 2: Problem Statement

- **Scalability:** Most current barcode recognition solutions have a problem scaling well in high-resolution or high-speed production applications. With manufacturing systems implementing increasingly high-resolution cameras and faster conveyor mechanisms, more real-time data processing is required. Middle-ground models have a propensity to fall behind these requirements, leading to a failure to make a detection or exhibit latent processing. Furthermore, fixed barcode formats or smaller dataset-trained systems are often inflexible in adapting to changing product lines, new types of barcodes, or altered forms of packaging, all of which are prevalent situations in high-volume industrial settings.

- **Integration Complexity:** Although the deep learning models are effective, they may not be compatible with the legacy or traditional industrial systems. The implementation of the models can substantially involve installing new hardware, modifying software and reworking the available infrastructure. The integration of deep learning algorithms with industrial environments often requires programmable logic controllers (PLCs) and embedded systems, which lack significant computational ability, making them technically unfeasible and economically inefficient to integrate. Applications to smaller facilities are subject to a high arrival threshold in the absence of light-weight and agile models.
- **Environmental Robustness:** Ensuring accuracy in different environmental conditions is one of the most significant challenges of industrial barcode recognition, as it has been observed numerous times that, despite some control over environmental factors, accuracy in a variable environment remains rather challenging. There may also be poor lighting, in-camera movement blur due to extreme motion, and glare involving matte or reflective materials, which can significantly degrade image quality. Such situations expose conventional image processing methods to especially strong attacks, and even trained deep learning models can be susceptible unless carefully trained with as many distortions and augmentations as possible. In an unrestricted real-world setting where a barcode can be partially obscured, skewed, or degraded, achieving consistent recognition performance is a complex and variable challenge.

2. Literature Survey

2.1. Traditional Barcode Recognition Techniques

The classical image processing techniques primarily used in the recognition of traditional barcodes include gradient analysis, edge detection, and thresholding. Algorithms such as Sobel or Canny filters are used to identify the most prominent edges in an image, which can generally be the barcode lines. [5-7] Additionally, approaches like the Hough Transform are implemented for detecting linear features, making them applicable in the detection process for a 1D barcode. The methods are, however, very susceptible to environmental changes, such as lighting conditions, image distortion, and the angles of perspective. Their detection in blur, shadows, or even uneven lighting conditions is significantly poor, and they cannot be trusted as much in the real world. Evaluations on tables indicate that OCR-based methods achieve a comparably high speed, but their accuracy is approximately 70 per cent, and they are not very robust. Template matching is slower and less robust to noise, but it is superior in certain cases. Edge detection has a slightly better accuracy of 75; however, it poses a challenge of robustness, particularly in a cluttered environment or one with a lot of activity.

2.2. Deep Learning in Object Recognition

With the introduction of deep learning, the identification of objects such as barcodes has undergone significant changes in terms of detection and recognition. Convolutional Neural Networks (CNNs) are especially good at extracting hierarchical features, allowing them to robustly classify the types of numerous barcodes, including EAN-13, QR, and DataMatrix. YOLO (You Only Look Once) is one of the most popular object detection frameworks, distinguished by its real-time capabilities and high accuracy. An example is YOLOv5, which achieves a detection speed of around 45 frames per second on top-tier GPUs, such as the NVIDIA RTX 3080. This is why it can be deployed in highly dynamic areas, such as retailing and logistics. Alternative models, such as Faster R-CNN, SSD (Single Shot Detector), and RetinaNet, offer better accuracy and feature extraction, but are more computationally demanding and may not be suitable when an edge device is required or in real-time applications. Deep learning models have the potential to learn complex relationships, and this feature makes them very accommodating to diverse imaging conditions, which makes them very helpful for strong detection of barcodes in real-world applications.

2.3. Existing Work in Barcode Detection

Recent research has succeeded in using deep learning to improve the detection of barcodes in harsh environments. A modified SSD (Single Shot Multibox Detector) was coupled with a custom decoder, which yielded an excellent result with a mean Average Precision (mAP) of 89%. Their strategy proved successful across multiple barcodes and backgrounds. Zhang et al. (2021) concentrated on the retail field and analyzes the possibility of detecting QR codes on product packaging using YOLOv4. Their system has demonstrated high accuracy and has worked effectively even in a cluttered shelf environment. In the meantime, we resolved the issue of barcode recognition in blurred or low-resolution photos by creating an edge-enhanced CNN. They combined deep features (learned representations) with traditional edge features to improve detection abilities in visually degraded environments. These research papers emphasise the increased promise of integrating traditional image features with deep learning to enhance robustness and precision in barcode recognition tasks.

3. Methodology

3.1. System Architecture

The Barcode recognition system is developed as a modular package with four major blocks: image capturing, preprocessing, detection using YOLOv5, and classification using a convolutional neural network (CNN). All modules serve a certain purpose to make the recognition of barcodes precise, [8-11] quick and robust in the presence of any kind of variation.

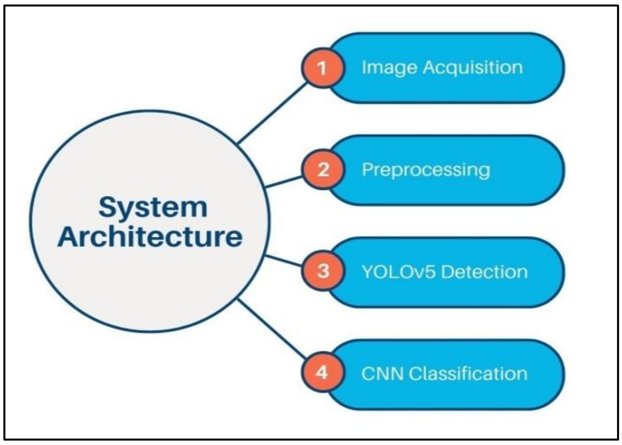

Fig 3: System Architecture

- **Image Acquisition:** The system is initiated by the image acquisition part, which utilises a high-speed industrial-grade camera to identify objects or packages in real-time. The cameras can operate effectively under harsh lighting and motion environments, resulting in clear and precise barcode images. High frame rate hardware is required to meet the real-time requirements of industrial automation, such as in conveyor belt systems or point-of-sale terminals.
- **Preprocessing:** The observed images are in most cases bad crippled by excessive noise, palp, or inconsistent light that may hamper the detection. In response to this, the preprocessing module employs methods, such as histogram equalisation, to enhance contrast and perform blur correction. Filters to sharpen the image. The operations enhance the quality and consistency of input data, and subsequently improve the reliability of subsequent detection and classification operations.
- **YOLOv5 Detection:** During the detection segment, the YOLOv5 (You Only Look Once version 5) deep learning architecture is utilized in order to identify the barcode area in the Image document. YOLOv5 offers a good trade-off between accuracy and speed, making it suitable for real-time applications. It undertakes one pass of the network in the forward direction and produces the locations of barcode areas in the form of bounding boxes with high probability scores.
- **CNN Classification:** Localization of the barcode is then followed by passing a region of interest to a classification module via CNN. In this module, the type of barcode (e.g., EAN-13, QR code, or DataMatrix) is identified, and the content is decoded. The CNN is also trained to identify minor distinctions between types of barcodes and manage distinctions in orientation, scale, and quality. The last step ensures the proper extraction of data for use in downstream processing or inventory systems.

3.2. Data Preprocessing

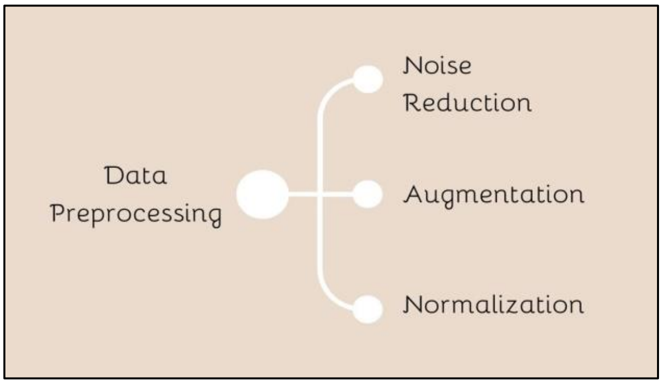

Fig 4: Data Preprocessing

Data preprocessing is the key to enhancing the performance and resilience of the barcode recogniser. This is the step where the raw image is prepared by making it consistent, holding less noisy data, and also providing greater chances of generalisation in the context of differing scenarios in the real world that are presented to the model. The preprocessing pipeline consists of noise reduction, data augmentation and normalization.

- **Noise Reduction:** Image contrast enhancement is achieved by applying noise reduction techniques, such as Gaussian blur and bilateral filter, to make the image clearer and eliminate unwanted fluctuations. A Gaussian blur will smooth high-frequency noise by averaging pixels in a local neighbourhood, but it will also lose edge information. We need the edge information to maintain the line integrity of the barcodes. They make these filters useful for decreasing the effects of noise present in sensors, compression artefacts, or an insufficient lighting environment.

- **Augmentation:** The dataset is artificially augmented to maximise the model's robustness and avoid overfitting. Other simulated operations include the rotation of barcodes at various stages, and the brightness variant enables the model to accommodate different lighting conditions. Cropping also introduces incomplete views of the barcodes, enabling the model to learn how to detect in conditions where a part of the code may be obstructed. Such augmentations increase the system's robustness to real-world variation.
- **Normalisation: Normalisation ensures that the input data is on the same scale, allowing neural networks** to converge much quicker. In such a system, the pixel intensity values are normalised to the limits of 0 and 1 by dividing by 255. The operation equalises the dynamic range of images, reduces variance in the number of computations, and makes the model more numerically stable during both training and inference. Normalisation ensures that the input data is on the same scale, allowing neural networks to converge much quicker. In such a system, the pixel intensity values are normalised to the limits of 0 and 1 by dividing by 255. The operation equalises the dynamic range of images, reduces variance in the number of computations, and makes the model more numerically stable during both training and inference.

3.3. YOLOv5 Detection Model

The YOLOv5 model is a time- and accuracy-oriented model that excels in estimating and object detection tasks, particularly in real-time applications such as barcode detection. [12-16] It has a modular architecture, which makes it efficient in processing data, and its design decisions produce an excellent tradeoff between speed and accuracy. The key components of the model are the input size, the backbone network, and a compound loss function.

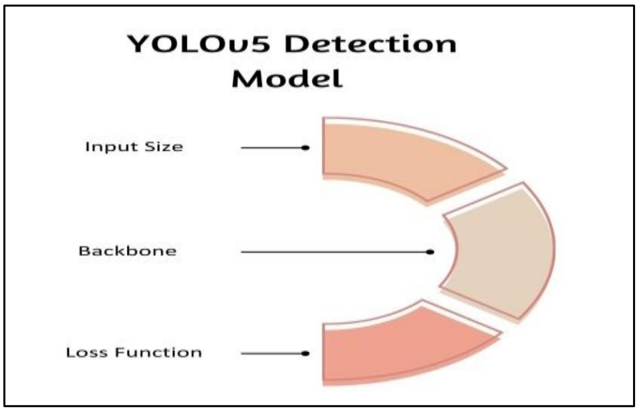

Fig 5: YOLOv5 Detection Model

- **Input Size:** YOLOv5 takes fixed-resolution input images with a 416x416 pixel shape, which is a typical tradeoff between concurrency and graininess. This recommendation eliminates the need to match barcode patterns with precision while minimising computations. By commanding the input images to this size, the inputs are processed consistently in both network training and inference on a variety of datasets.
- **Backbone:** YOLOv5 is supported by the CSPDarknet backbone that is based on the Darknet network with Cross Stage Partial (CSP) connections. CSPDarknet enhances the gradient operation and eliminates repetition by dividing the feature map into two halves and reuniting them subsequently. Such a design improves the efficiency of feature extraction to the extent that it can support small, dense features, such as barcodes, even in cluttered scenes or low-resolution input.
- **Loss Function:** In YOLOv5, a compound loss function is adopted to drive the model during training. It utilises Generalised Intersection over Union (GIoU) loss to enhance the localisation of bounding boxes, objectness loss to assess the presence or absence of an object in an object prediction region, and classification loss to accurately classify the object class. This combination of all these losses assists the model to train with precise, confident, and accurate detection in a diverse set of situations.

3.4. CNN Classifier

The CNN classifier in the suggested framework would detect the type and decode the contents of barcodes, utilising the localisation capabilities of the YOLOv5 detector. The model is programmed to strike a balance between computational efficiency and precision in recognising patterns, allowing the system to perform well in real-time applications.

- **Architecture:** The network design is four convolutional layers and two fully connected (dense) layers. The convolutional layers extract data on the hierarchical spatial features in the barcode image, representing patterns that are important in terms of structure, format, and distribution of lines. Non-linearities are imposed between layers in a process known as pooling to compress the dimensionality and keep important features. It is this information that is aggregated in the final two dense layers to carry out abstraction-level reasoning and classification.

- **Activation:** After all convolutional and dense layers (except the output layer), this network employs the ReLU (Rectified Linear Unit) activation function, which introduces non-linearity and facilitates faster training. For the output layer, it uses the synthetic activation function, Softmax, to generate a probability distribution of the probable barcode types. This enables the model to perform high-confidence input classification for EAN-13, QR code, DataMatrix, and other formats.

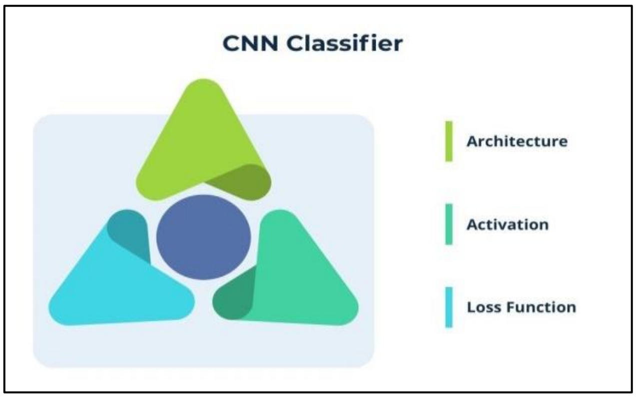

Fig 6: CNN Classifier

- **Loss Function:** Categorical cross-entropy loss, commonly used in multi-class classification problems, is employed during model training. This loss is used as a measure of the discrepancy between the probability distribution of the estimated probability and the actual label. Still, it imposes larger penalties in the event of a wrong estimation. It controls the network to refine the classification results by reducing the difference between the predicted and actual classes during training.

4. Case Study / Evaluation

4.1. Experimental Setup

The experiment set up to assess the proposed barcode recognition system was intended to reflect realistic industrial conditions; therefore, the performance metric is likely to provide a proper estimate of the feasibility of deployment. The system boasts a powerful hardware foundation, featuring a top-of-the-line NVIDIA RTX 3080 GPU, which provides the power of deep learning models with over 10,000 CUDA cores and 10 GB of VRAM, enabling parallel processing within the system. This GPU supports real-time observation of object detection and classification activities, encompassing both training and inference processes for YOLOv5 and CNN-based models. Regarding the software side, the system was developed and tested using two of the most popular deep learning frameworks: PyTorch and TensorFlow 2.6. The reason for using PyTorch in the implementation of the YOLOv5 detection module was its flexibility and the possibility to customise the model.In contrast, TensorFlow was used to create the CNN classifier and mobilise the model, as it has a good ecosystem for rapid deployment and mobile deployment.

The two frameworks were executed on a Conda-controlled Python environment, which provided stability to the packages and consistency in experimental results. The system was installed and tested on a simulated factory-line system, which featured dynamic/variable lighting conditions (dim to overexposed illumination), closely replicating factory and industrial lighting conditions. Packages were transported using conveyor belts at variable speeds, and barcode labels of different types and orientations were applied to simulate a diverse range of operations. During testing, the illumination was deliberately adjusted to determine the system's robustness in response to environmental variations and its ability to withstand shadow and glare interference. High-speed industrial cameras were used to capture images at a rate greater than 60 frames per second, ensuring that the full image resolution was captured. It also included random occlusions and motion blur as part of the experimental design to verify the robustness of the preprocessing and detection pipeline. This has been a comprehensive setup, and the modules, including image acquisition and classification, were all engaged during operating conditions that reflect the actual deployment environment.

4.2. Dataset

The training and evaluation dataset for proposing the barcode recognition system comprises a total of 25,000 images, which include images of various real-life circumstances sampled during several working shifts. The pictures were taken at a functioning factory conveyor, wherein packages and goods going through conveyor belts under different light, speed, and position conditions had barcodes attached to them. Coupled with the differences in time of acquisition and environmental conditions, this variety ensures that the data used will be highly representative of industrial applications, including challenging situations with shadows, uneven lighting conditions, partial occlusions, and motion blur. There are three major types of barcodes included in the dataset: EAN-13, generally used in packaging products at shops; QR codes, widely used in marketing activities and product monitoring; DataMatrix codes that are dominant in the industrial sectors, as they have great information density and can be read in a small surface area. To enhance the effectiveness of the detection and classification models, the dataset was further augmented with synthetic noise and distortions, thereby increasing robustness and generalisation capabilities. Examples of augmentation methods included Gaussian and salt-and-pepper noise to approximate sensor errors, random brightness and contrast changes to simulate uneven lighting, and geometric transformations such as rotation, cropping, and scaling.

A systematic use of these augmentations artificially enhanced the variability of the datasets, with the aim of training the models to recognise barcodes in challenging visual conditions. Notably, the original and extended databases were well-marked with bounding boxes and class labels to facilitate supervised learning. LabelImg was used to annotate the detection part, and scripts were used to label the classification. The dataset was divided into training (70 per cent), validation (15 per cent), and test (15 per cent) sets to ensure a proper distribution of the types of barcodes represented and lighting conditions across all phases. Such a wide source of heterogeneous data proved to be a crucial source material through which a system can be developed that is also dependable in complex and varying environments.

4.3. Evaluation Metrics

To evaluate the effectiveness of the barcode recognition system, specific evaluation metrics were employed. These numbers determine the accuracy, reliability, and real-time performance of the model. The first finding shows that the system can be applied to industrial-type conditions where precision and speed are equally important.

Table 1: Evaluation Metrics

Metric	Value
Mean Average Precision (mAP)	91.2%
Precision	93.5%
Recall	89.7%
Inference Time	22 %

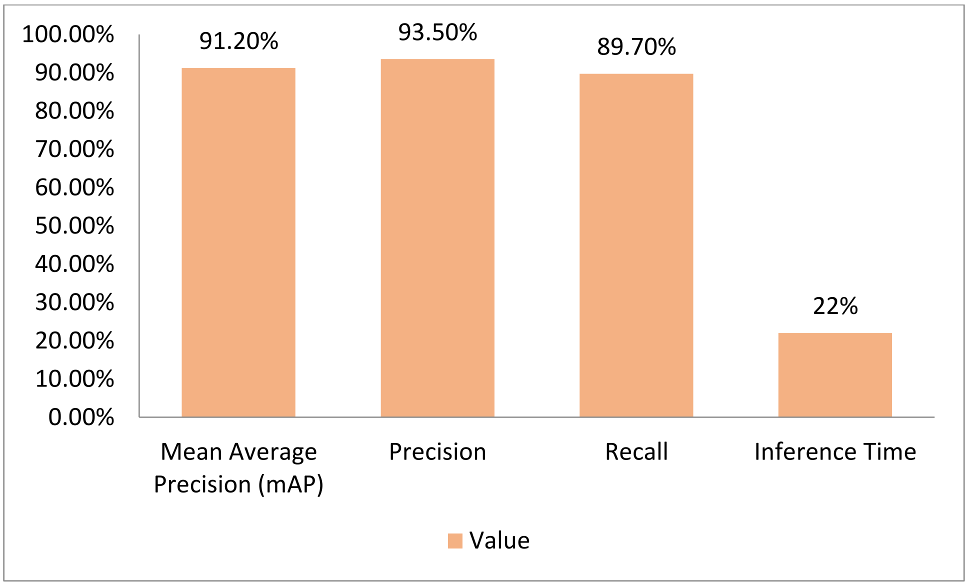

Fig 7: Graph representing Evaluation Metrics

- **Mean Average Precision (mAP) -91.2:** The mAP of 91.2% indicates a very high level of accuracy in determining and identifying various types of barcodes in the Convolutional Neural Network (CNN) dataset, acquired with precision. The precision is an average across all classes and the level of average area under the IoU. An acceptable mAP will demonstrate that the system generates accurate results (i.e., detection of barcode regions and minimal false positives or missed detections).
- **Precision – 93.5%:** With a precision score of 93.5, the system demonstrates a high degree of capability in making true positive predictions. This indicates that the model successfully recognises a barcode with an accuracy of more than 93 per cent. Accuracy in industrial use is essential, as false positives of not reading non-barcode textures — which, in the worst case, may cause a process failure or prevent matching invoices with inventory items — are crucial.

- **Recall – 89.7%:** The recall rate of 89.7% indicates that the system can identify most of the barcodes found in the input images. Good recall rates mean that only a small proportion of barcodes are read incorrectly, even in challenging situations, such as partial occlusion, distortion, or poor lighting. This is to ensure accuracy in operation and prevent unchecked products in the line.
- **Inference Time –** 22%: On average, the system can process 22 milliseconds per image, allowing it to operate in real-time at approximately 45 frames per second (FPS). This speed is best suited for a high-throughput setup, such as an assembly line or a point-of-sale system, where this speed of detection or otherwise bottlenecks and delays are important.

5. Results and Discussion

5.1. Quantitative Results

Comprising three model variations namely, YOLOv5, CNN Only, and the combined YOLOv5 + CNN technique —the proposed barcode recognition system was accordingly tested. Three important performance indicators (accuracy, mean average precision (mAP) and inference time) were used to evaluate each of the configurations. The trade-offs between speed and accuracy in these results were important and indicated the effectiveness of the hybrid architecture.

Table 2: Quantitative Results

Model	Accuracy	mAP	Inference Time
YOLOv5	94.5%	93.2%	19 %
CNN Only	90.4%	89.7%	10 %
YOLO + CNN	98.4%	96.2%	21 %

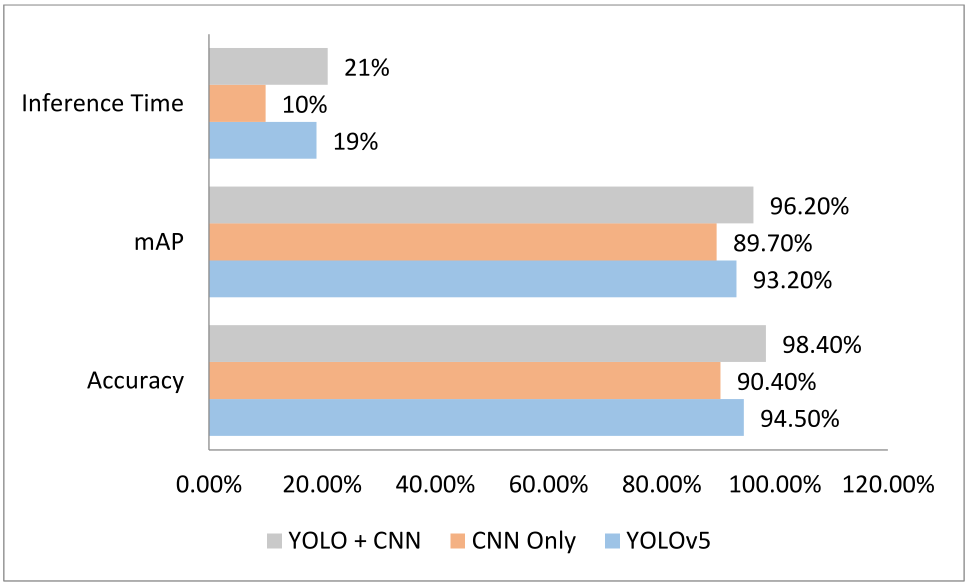

Fig 8: Graph representing Quantitative Results

- **YOLOv5 –** Accuracy: 94.5%, mAP: 93.2%, Inference Time: 19%. The YOLOv5 system alone achieves excellent results, with an accuracy of 94.5% and an mAP of 93.2, enabling it to identify areas of barcodes with high precision and repeatability. It also provides high speed, corresponding to the fact that processing time is within 19 milliseconds per frame, thus making it suitable for real-time applications. Nonetheless, it is only as effective as detecting barcodes but cannot be as precise as the CNN module when it comes to classification details.
- **CNN -** Accuracy: 90.4 per cent, mAP: 89.7 per cent, Inference Time: 10 per cent. The CNN-only strategy, which classifies without YOLO-based region localisation, demonstrates low accuracy and mAP of 90.4% and 89.7%, respectively. Even though it has a quicker inference period of 10 milliseconds, it does not successfully identify the relative location of barcode regions because it already uses cropped or extracted barcode pictures. This is more applicable in controlled environments where barcodes have fixed positions.
- **YOLO + CNN - Accuracy: 98.4%, mAP: 96.2%, Inference Time: 21 %.** The overall performance of the combined YOLO + CNN architecture is outstanding, achieving 98.4% accuracy and 96.2% mAP. Such an arrangement utilises YOLOv5 for accurate region detection and the CNN for precise barcode classification. The time it takes to make an inference rises slightly, up to 21 milliseconds, but this also allows it to be real-time processing capable, and gives it better detection and decoding accuracy, making it excellent in dynamic, high-speed industrial scenes.

5.2. Qualitative Results

The qualitative performance of the proposed barcode recognition system demonstrates its high practical utility and reliability in the most challenging visual environments. It was found that one of the most striking notes was that the system can correctly read barcodes in partially occluded cases. In various test conditions where coverage by packages, by hand, or other labels blocked the barcode area, the system, specifically the YOLO + CNN solution, retrieved only the visible parts of the barcode and successfully read the type and contents on the barcode. This has, in part, been enabled by the potent region proposal network in YOLOv5, which can localise objects using incomplete visual information, as well as the inference that predicts the type and structure given incomplete data by the CNN classifier. This type of performance is necessary in industrial settings where barcode labels are often damaged or misaligned. The system was also very accurate even in different lighting conditions, such as backlit, dark, and overexposed pictures.

When being tested on a moving assembly line, the lighting was deliberately adjusted to mimic real-world changes in ambient brightness and direction. Preprocessing module: The preprocessing part of the system, which comprised histograms and blur removal, was effective in standardising the quality of the input. The model was also able to handle various levels of light (when combined with sturdy training on augmented data, including brightness and contrast changes). In some instances, when most of the barcode image was white out due to glare, the system managed to capture the code and read it with remarkable accuracy**.** The ability to work constantly in such challenging circumstances is a testament to the power of both architecture and the training system. It does not require an ideal condition of input, and this is why it is very appropriate for use in real-world situations, such as warehouses, logistics areas, and shop settings. On the whole, the qualitative results supplement the quantitative measures by demonstrating the resilience and generalizability of the model in terms of its applicability to a broad range of real-world situations.

5.3. Limitations

Although the proposed barcode recognition system is highly accurate and functions well in various industrial settings, it is associated with some limitations that may hinder its implementation in specific environments. Among the major limitations is that it is GPU-dependent. The real-life use of the model, especially in its integrated form with YOLOv5 and the CNN model, greatly depends on the parallel CPU utilisation when computing with high-performance GPUs (e.g., NVIDIA RTX 3080). Such dependency imposes limitations on the use of the system at the edge, as it must operate within constrained power and hardware units. Small embedded devices (e.g., embedded systems or compact edge modules, such as Jetson Nano or Raspberry Pi) may lack sufficient computing resources to maintain the desired frame rates and detection precision. Consequently, it may be necessary to achieve a very close approximation of model performance using the complete model stack, albeit at the expense of accuracy and robustness, particularly with low-weight models.

One additional limitation that will be addressed in this testing is that there will be a slight delay in detecting and recognising when the conveyor is operating at a very high speed, especially above 2 meters per second. With this condition, there is sometimes latency in loading the image into the system, resulting in a mismatch between the barcode position and the detection window in the model. Even though the average inference time is low (approximately 21 milliseconds), the fast movement of objects and short exposure time of the camera contribute to the emergence of motion blur or even skipped frames. This difficulty is compounded by the fact that in many cases, barcodes are small or have imperfections, making them harder to detect. This problem can be somewhat mitigated by using motion blur correction and operating with frames captured at a higher frame rate. Still, recognition delay or failure in ultra-high-speed tasks remains a potential risk.

6. Conclusion and Future Work

The study proposes a fast industrial barcode reading framework that utilises deep learning. Based on its high accuracy rates, fast speed, and flexibility, the system is considered evolutionary when compared with its state-of-the-art models. Through the advantages of two popular model frameworks, YOLOv5, which is used for real-time object detection, and a custom CNN that detects barcode types, the proposed system can maintain a high level of detection precision and low processing requirements, leveraging the strengths of both frameworks. As can be seen in the experiment's results, both quantitative and qualitative, it is possible to conclude that the system is capable of recognising different kinds of barcodes, including EAN-13, QR codes, and DataMatrix symbols, even under adverse conditions such as partial occlusion, dynamic illumination, and motion blur. The framework of the modular pipeline, which links preprocessing, detection, and classification into one efficiency-optimised pipeline, is targeted at compatibility with industrial automation systems.

This property further enables its successful implementation on factory floors, in warehouses, and logistics centres. Namely, the inference time (21 ms per image) and high accuracy (98.4%) enable the commercial application of the system in real-time settings with high throughput. It was, however, demonstrated that there are limits in terms of dependence on GPUs, and that performance degrades significantly at extremely high conveyor belt speeds. This suggests that there is still potential to expand upon its results. Concerning further development, several directions can be explored to increase the scalability and flexibility of the system's deployment. The use of lightweight models (e.g., YOLO-Nano or MobileNet-based architectures) that are more suitable for running on edge devices with low computational power is one area. Doing this would allow the system to be used in a portable or embedded situation, where it would have a wider range of applications in field operations, handheld scanners, and other small robot types.

Additionally, examining few-shot learning and self-supervised learning may substantially enhance the system's flexibility in detecting new types of barcodes, or even rare ones, without the need for retraining with huge annotated data. This would be especially handy in industries where the product line or labelling system changes regularly. Regular industrial robotic vision can also be integrated to perform real-time scanning of barcode information, in addition to which the items can be handled, properly checked, or automatically separated. This integration would establish a direct connection between optical perception and physical movement, further developing automation in industrial and supply chains. All in all, the system provides a powerful foundation for implementing intelligent barcode recognition, leaving sufficient scope for further innovations and practical applications.

References

[1] Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. Pattern recognition, 13(2), 111-122.

[2] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.

[3] Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1), 11-15.

[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25.

[5] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 28.

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, September). SSD: Single-shot multibox detector. In European Conference on Computer Vision (pp. 21-37). Cham: Springer International Publishing.

[7] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2980-2988).

[8] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[9] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[10] Xiao, Y., & Ming, Z. (2019). 1D Barcode Detection via Integrated Deep-Learning and Geometric Approach. Applied Sciences, 9(16), Article 3268.

[11] Jogin, M., Madhulika, M. S., Divya, G. D., Meghana, R. K., & Apoorva, S. (2018, May). Feature extraction using convolutional neural networks (CNN) and deep learning. In 2018, the 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 2319-2323). IEEE.

[12] Hansen, D. K., Nasrollahi, K., Rasmussen, C. B., & Moeslund, T. B. (2017). Real-time barcode detection and classification using deep learning. In International Joint Conference on Computational Intelligence (pp. 321-327). SciTePress.

[13] Hashim, N. M. Z., Ibrahim, N. A., Saad, N. M., Sakaguchi, F., & Zakaria, Z. (2013). Barcode recognition system. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2(4), 278-283.

[14] Wudhikarn, R., Charoenkwan, P., & Malang, K. (2022). Deep learning in barcode recognition: A systematic literature review. IEEE Access, 10, 8049-8072.

[15] Song, Q., Li, S., Bai, Q., Yang, J., Zhang, X., Li, Z., & Duan, Z. (2021). Object Detection Method for a Grasping Robot Based on Improved YOLOv5. Micromachines, 12(11), 1273.

[16] Liang, H., Chai, S., Zhang, C., Qirong, L., Li, J., & Wang, H. (2019). A Novel Algorithm for Damaged Barcode Recognition Based on Deep Learning. DEStech Transactions on Computer Science and Engineering, CISNRC 2019.

[17] Youssef, S. M., & Salem, R. M. (2007). Automated barcode recognition for smart identification and inspection automation. Expert Systems with Applications, 33(4), 968-977.

[18] Sahin, E., Dallery, Y., & Gershwin, S. (2002, October). Performance evaluation of a traceability system. An application of the radio frequency identification technology. In IEEE International Conference on Systems, Man and Cybernetics (Vol. 3, pp. 6-pp). IEEE.

[19] Kamnardsiri, T., Charoenkwan, P., Malang, C., & Wudhikarn, R. (2022). 1D barcode detection: Novel benchmark datasets and comprehensive comparison of deep convolutional neural network approaches. Sensors, 22(22), 8788.

[20] Zhu, X. (2021, August). Design of a barcode recognition system based on YOLOV5. In Journal of Physics: conference series (Vol. 1995, No. 1, p. 012052). IOP Publishing.

[21] Rusum, G. P., Pappula, K. K., & Anasuri, S. (2020). Constraint Solving at Scale: Optimizing Performance in Complex Parametric Assemblies. *International Journal of Emerging Trends in Computer Science and Information Technology*, *1*(2), 47-55. https://doi.org/10.63282/3050-9246.IJETCSIT-V1I2P106

[22] Rahul, N. (2020). Optimizing Claims Reserves and Payments with AI: Predictive Models for Financial Accuracy. *International Journal of Emerging Trends in Computer Science and Information Technology*, *1*(3), 46-55. https://doi.org/10.63282/3050-9246.IJETCSIT-V1I3P106

[23] Enjam, G. R. (2020). Ransomware Resilience and Recovery Planning for Insurance Infrastructure. *International Journal of AI, BigData, Computational and Management Studies*, *1*(4), 29-37. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V1I4P104
[24] Pedda Muntala, P. S. R., & Karri, N. (2021). Leveraging Oracle Fusion ERP's Embedded AI for Predictive Financial Forecasting. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(3), 74-82. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I3P108
[25] Rahul, N. (2021). Strengthening Fraud Prevention with AI in P&C Insurance: Enhancing Cyber Resilience. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(1), 43-53. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I1P106
[26] Enjam, G. R. (2021). Data Privacy & Encryption Practices in Cloud-Based Guidewire Deployments. *International Journal of AI, BigData, Computational and Management Studies*, *2*(3), 64-73. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I3P108
[27] Rusum, G. P. (2022). WebAssembly across Platforms: Running Native Apps in the Browser, Cloud, and Edge. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(1), 107-115. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I1P112
[28] Jangam, S. K. (2022). Self-Healing Autonomous Software Code Development. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(4), 42-52. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P105
[29] Anasuri, S. (2022). Adversarial Attacks and Defenses in Deep Neural Networks. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(4), 77-85. https://doi.org/10.63282/xs971f03
[30] Pedda Muntala, P. S. R. (2022). Anomaly Detection in Expense Management using Oracle AI Services. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(1), 87-94. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P109
[31] Rahul, N. (2022). Automating Claims, Policy, and Billing with AI in Guidewire: Streamlining Insurance Operations. International Journal of Emerging Research in Engineering and Technology, 3(4), 75-83. https://doi.org/10.63282/3050-922X.IJERET-V3I4P109
[32] Enjam, G. R. (2022). Energy-Efficient Load Balancing in Distributed Insurance Systems Using AI-Optimized Switching Techniques. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(4), 68-76. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I4P108
[33] Rusum, G. P., & Anasuri, S. (2023). Composable Enterprise Architecture: A New Paradigm for Modular Software Design. *International Journal of Emerging Research in Engineering and Technology*, *4*(1), 99-111. https://doi.org/10.63282/3050-922X.IJERET-V4I1P111
[34] Jangam, S. K., & Pedda Muntala, P. S. R. (2023). Challenges and Solutions for Managing Errors in Distributed Batch Processing Systems and Data Pipelines. *International Journal of Emerging Research in Engineering and Technology*, *4*(4), 65-79. https://doi.org/10.63282/3050-922X.IJERET-V4I4P107
[35] Anasuri, S. (2023). Secure Software Supply Chains in Open-Source Ecosystems. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(1), 62-74. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P108
[36] Pedda Muntala, P. S. R., & Karri, N. (2023). Leveraging Oracle Digital Assistant (ODA) to Automate ERP Transactions and Improve User Productivity. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *4*(4), 97-104. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I4P111
[37] Rahul, N. (2023). Transforming Underwriting with AI: Evolving Risk Assessment and Policy Pricing in P&C Insurance. *International Journal of AI, BigData, Computational and Management Studies*, *4*(3), 92-101. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P110
[38] Enjam, G. R. (2023). Modernizing Legacy Insurance Systems with Microservices on Guidewire Cloud Platform. *International Journal of Emerging Research in Engineering and Technology*, *4*(4), 90-100. https://doi.org/10.63282/3050-922X.IJERET-V4I4P109
[39] Rahul, N. (2020). Vehicle and Property Loss Assessment with AI: Automating Damage Estimations in Claims. *International Journal of Emerging Research in Engineering and Technology*, *1*(4), 38-46. https://doi.org/10.63282/3050-922X.IJERET-V1I4P105
[40] Enjam, G. R., & Tekale, K. M. (2020). Transitioning from Monolith to Microservices in Policy Administration. *International Journal of Emerging Research in Engineering and Technology*, *1*(3), 45-52. https://doi.org/10.63282/3050-922X.IJERETV1I3P106
[41] Pedda Muntala, P. S. R., & Jangam, S. K. (2021). End-to-End Hyperautomation with Oracle ERP and Oracle Integration Cloud. *International Journal of Emerging Research in Engineering and Technology*, *2*(4), 59-67. https://doi.org/10.63282/3050-922X.IJERET-V2I4P107
[42] Rahul, N. (2021). AI-Enhanced API Integrations: Advancing Guidewire Ecosystems with Real-Time Data. *International Journal of Emerging Research in Engineering and Technology*, *2*(1), 57-66. https://doi.org/10.63282/3050-922X.IJERET-V2I1P107

[43] Enjam, G. R., & Chandragowda, S. C. (2021). RESTful API Design for Modular Insurance Platforms. *International Journal of Emerging Research in Engineering and Technology, 2*(3), 71-78. https://doi.org/10.63282/3050-922X.IJERET-V2I3P108

[44] Rusum, G. P., & Pappula, kiran K. . (2022). Event-Driven Architecture Patterns for Real-Time, Reactive Systems. *International Journal of Emerging Research in Engineering and Technology*, *3*(3), 108-116. https://doi.org/10.63282/3050-922X.IJERET-V3I3P111

[45] Jangam, S. K., & Karri, N. (2022). Potential of AI and ML to Enhance Error Detection, Prediction, and Automated Remediation in Batch Processing. *International Journal of AI, BigData, Computational and Management Studies*, *3*(4), 70-81. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I4P108

[46] Anasuri, S. (2022). Formal Verification of Autonomous System Software. *International Journal of Emerging Research in Engineering and Technology, 3*(1), 95-104. https://doi.org/10.63282/3050-922X.IJERET-V3I1P110

[47] Pedda Muntala, P. S. R. (2022). Natural Language Querying in Oracle Fusion Analytics: A Step toward Conversational BI. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(3), 81-89. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I3P109

[48] Rahul, N. (2022). Optimizing Rating Engines through AI and Machine Learning: Revolutionizing Pricing Precision. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(3), 93-101. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I3P110

[49] Enjam, G. R. (2022). Secure Data Masking Strategies for Cloud-Native Insurance Systems. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(2), 87-94. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I2P109

[50] Rusum, G. P., & Anasuri, S. (2023). Synthetic Test Data Generation Using Generative Models. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(4), 96-108. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I4P111

[51] Jangam, S. K. (2023). Data Architecture Models for Enterprise Applications and Their Implications for Data Integration and Analytics. International Journal of Emerging Trends in Computer Science and Information Technology, 4(3), 91-100. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I3P110

[52] Anasuri, S., Rusum, G. P., & Pappula, K. K. (2023). AI-Driven Software Design Patterns: Automation in System Architecture. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *4*(1), 78-88. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I1P109

[53] Pedda Muntala, P. S. R., & Karri, N. (2023). Managing Machine Learning Lifecycle in Oracle Cloud Infrastructure for ERP-Related Use Cases. *International Journal of Emerging Research in Engineering and Technology*, *4*(3), 87-97. https://doi.org/10.63282/3050-922X.IJERET-V4I3P110

[54] Rahul, N. (2023). Personalizing Policies with AI: Improving Customer Experience and Risk Assessment. International Journal of Emerging Trends in Computer Science and Information Technology, 4(1), 85-94. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P110

[55] Enjam, G. R., Tekale, K. M., & Chandragowda, S. C. (2023). Zero-Downtime CI/CD Production Deployments for Insurance SaaS Using Blue/Green Deployments. *International Journal of Emerging Research in Engineering and Technology*, *4*(3), 98-106. https://doi.org/10.63282/3050-922X.IJERET-V4I3P111

[56] Pedda Muntala, P. S. R., & Jangam, S. K. (2021). Real-time Decision-Making in Fusion ERP Using Streaming Data and AI. *International Journal of Emerging Research in Engineering and Technology*, *2*(2), 55-63. https://doi.org/10.63282/3050-922X.IJERET-V2I2P108

[57] Rusum, G. P. (2022). Security-as-Code: Embedding Policy-Driven Security in CI/CD Workflows. *International Journal of AI, BigData, Computational and Management Studies*, *3*(2), 81-88. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I2P108

[58] Jangam, S. K., Karri, N., & Pedda Muntala, P. S. R. (2022). Advanced API Security Techniques and Service Management. *International Journal of Emerging Research in Engineering and Technology*, *3*(4), 63-74. https://doi.org/10.63282/3050-922X.IJERET-V3I4P108

[59] Anasuri, S., Rusum, G. P., & Pappula, kiran K. (2022). Blockchain-Based Identity Management in Decentralized Applications. International Journal of AI, BigData, Computational and Management Studies, *3*(3), 70-81. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I3P109

[60] Pedda Muntala, P. S. R. (2022). Enhancing Financial Close with ML: Oracle Fusion Cloud Financials Case Study. *International Journal of AI, BigData, Computational and Management Studies*, *3*(3), 62-69. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I3P108

[61] Rusum, G. P. (2023). Large Language Models in IDEs: Context-Aware Coding, Refactoring, and Documentation. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(2), 101-110. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P110

[62] Jangam, S. K., & Karri, N. (2023). Robust Error Handling, Logging, and Monitoring Mechanisms to Effectively Detect and Troubleshoot Integration Issues in MuleSoft and Salesforce Integrations. *International Journal of Emerging Research in Engineering and Technology*, *4*(4), 80-89. https://doi.org/10.63282/3050-922X.IJERET-V4I4P108

[63] Anasuri, S. (2023). Synthetic Identity Detection Using Graph Neural Networks. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(4), 87-96. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I4P110
[64] Reddy Pedda Muntala, P. S., & Karri, N. (2023). Voice-Enabled ERP: Integrating Oracle Digital Assistant with Fusion ERP for Hands-Free Operations. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(2), 111-120. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P111
[65] Enjam, G. R. (2023). Optimizing PostgreSQL for High-Volume Insurance Transactions & Secure Backup and Restore Strategies for Databases. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(1), 104-111. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P112