ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1263

Intelligent Resource Optimization: Enhancing Component Reuse through AI-Driven Image Classification

Kunal G. Borase[1]; Dhanashree Meshram[2]; Sowmiya Radhakrishnan[3]; Praveen Kumar Burra[4]; Bharani Kumar Depuru[5]

[1;2;3;4;5]Aispry

Publication Date: 2025/04/11

Abstract: This paper presents an AI-powered system designed to automate the identification and cataloging of electric switchgear components, improving inventory management and minimizing errors caused by manual classification. Traditional identification methods rely on human efforts, which are labor-intensive and prone to misclassification, leading to inefficiencies in warehouse operations. To overcome these challenges, we leveraged YOLO-based deep learning models to classify switchgear components accurately while ensuring seamless integration with inventory records. Our approach involved training YOLO models to classify switchgear components based on their unique visual features. The model matches each identified component against a Master Data Sheet containing essential details such as part numbers, dimensions, weight, and material specifications. By leveraging YOLO's advanced feature extraction and classification capabilities, our system achieves high precision in distinguishing visually similar components, ensuring reliable and real-time processing suitable for industrial deployment. During model development, we addressed critical challenges such as variations in lighting conditions, different orientations of components, and cluttered warehouse environments. Extensive data augmentation techniques[10] and model fine-tuning were applied to enhance robustness and maintain high classification accuracy across diverse scenarios. The final AI model achieves up to 95% accuracy, significantly reducing manual identification efforts by 70%, demonstrating its effectiveness in real-world applications. By automating switchgear component identification, our system significantly enhances inventory tracking, minimizes human errors, and optimizes warehouse efficiency. This research highlights the transformative potential of YOLO-based AI automation in industrial inventory management, paving the way for future advancements in intelligent spare part classification and cataloging.

Keywords: *Image Classification, Component Identification, YOLOv8n-cls, YOLOv8s-cls, YOLOv8m-cls, Deep Learning, Data Augmentation, CRISP-ML(Q), Inventory Management, Resource Optimization, AI Automation, Warehouse Efficiency.*

How to Cite: Kunal G. Borase; Dhanashree Meshram; Sowmiya Radhakrishnan; Praveen Kumar Burra; Bharani Kumar Depuru (2025). Intelligent Resource Optimization: Enhancing Component Reuse through AI-Driven Image Classification. *International Journal of Innovative Science and Research Technology*, 10(3), 2613-2622. https://doi.org/10.38124/ijisrt/25mar1263

I. INTRODUCTION

Electric switchgear is made up of different components, each playing an important role in controlling and distributing electrical power safely. These components, such as circuit breakers, switches, contactors, fuses, and protective relays, work together to prevent overloads, short circuits, and voltage fluctuations. Once all the parts are assembled, they form a complete switchgear unit, ensuring the safe and efficient operation of electrical systems. However, during the assembly process, some extra or unused parts are left behind, making it difficult to track and manage them properly. Identifying these leftover components is necessary to reduce waste, improve inventory management, and make better use of resources. Traditional methods like manual checking and barcode scanning often lead to misplaced items, wrong labeling, and delays in stock updates, and as the number of components increases, these issues become harder to handle. To solve this problem, this research focuses on developing an AI-powered system that can automatically recognize and classify both assembled and leftover components. This system will enable real-time inventory tracking, helping warehouses keep accurate records, reduce manual effort, and minimize errors. By using AI for classification, warehouse operations[11] can become faster, more efficient, and better organized, leading to improved productivity and resource management.

Artificial intelligence has become a powerful tool for automating inventory processes, significantly reducing human effort and errors. The proposed system employs YOLO(You Only Look Once) single shot image classification model to classify electric switchgear components even in challenging warehouse conditions, such as poor lighting, and varying orientations. Unlike traditional inventory tracking systems, AI-powered solutions provide faster, more accurate, and scalable inventory management, enabling seamless warehouse operations[11].

To build a robust and reliable classification system, a comprehensive dataset of switchgear component images was collected and annotated. These images, captured from multiple angles and under different environmental conditions, were carefully labeled to train the AI model effectively. To improve generalization, data augmentation techniques[10] such as brightness normalization, contrast adjustments, rotation, and noise filtering were applied, ensuring that the model maintains high classification accuracy across diverse warehouse settings.

A structured methodology, CRISP-ML(Q)[5], was followed to develop the AI-based inventory system. This approach provides a systematic framework for building machine learning models, ensuring efficient execution of each phase from data collection and preprocessing to model training, evaluation, and deployment. The system was evaluated using key performance metrics, including precision, recall, confusion matrix analysis, and classification accuracy. Testing different YOLO versions allowed us to select the most efficient model for deployment, ensuring high accuracy in component classification and seamless inventory tracking.

By implementing this AI-powered vision system, warehouses can reduce manual stock verification efforts, improve tracking accuracy, and optimize inventory workflows. This study demonstrates how deep learning-based classification can revolutionize warehouse management [11]by providing businesses with automated, real-time component identification, minimal human errors, and enhanced operational efficiency. Through this approach, organizations can achieve cost savings, increased productivity, and better resource utilization, making AI-driven inventory automation an essential advancement in modern warehouse operations[11].

The project methodology followed here is the open source CRISP-ML(Q)[5] methodology from 360DigiTMG(ak.1) [Fig 1], which stands for Cross Industry Standard Process for Machine Learning with Quality Assurance. This structured methodology ensures a systematic approach to problem identification, data preprocessing, model training, evaluation, deployment and monitoring and maintenance. By following CRISP-ML(Q), the project follows a well-defined lifecycle from data collection to real-world implementation, ensuring robust and efficient model performance.

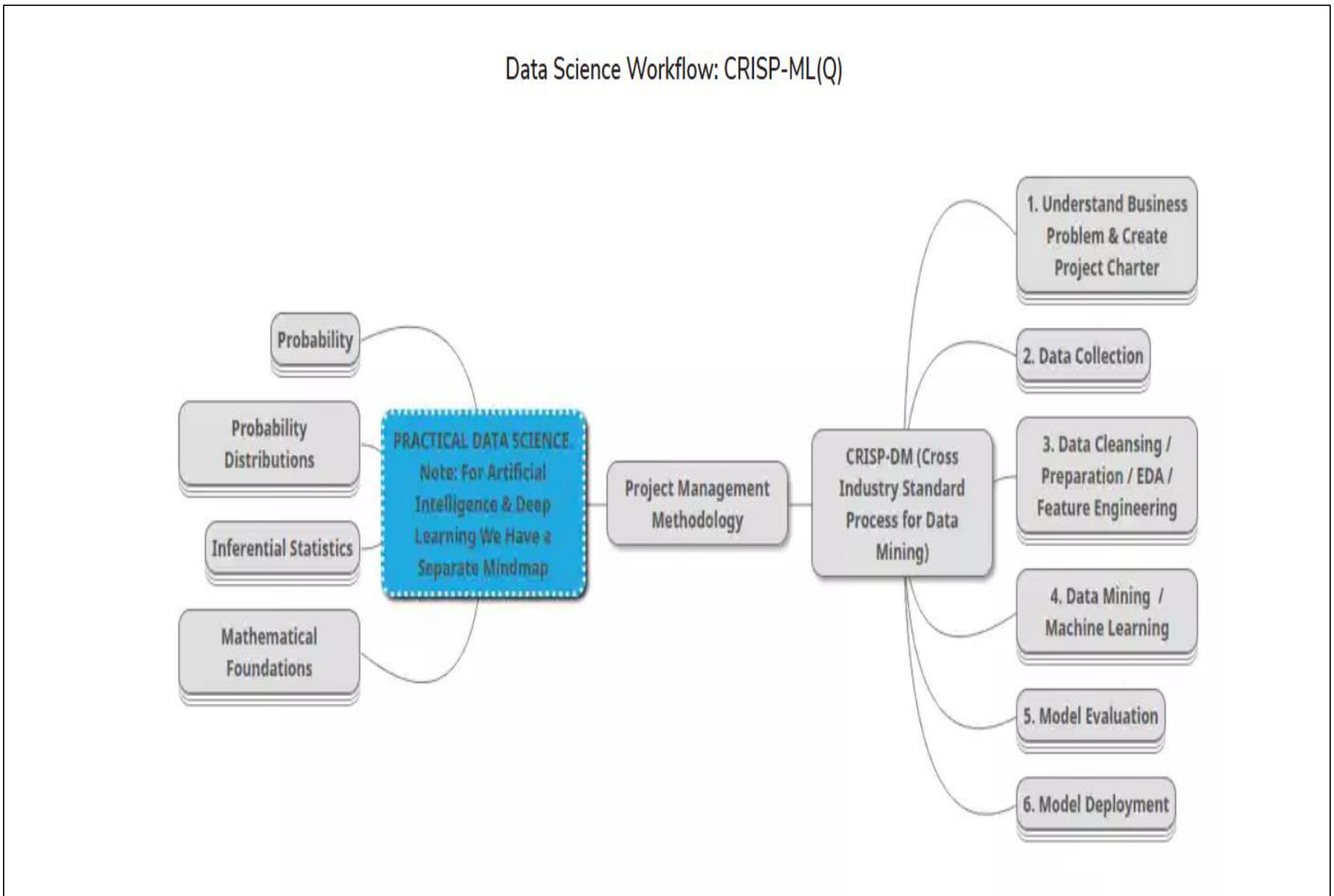

Fig 1 This Figure Depicts the CRISP-ML(Q) Architecture that We have followed for this Research Study.
(Source: Mind Map - 360DigiTMG)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1263

II. METHODOLOGY AND TECHNOLOGY

A. Data Collection

The dataset utilized in this research was collected directly from the client's supplier location, capturing images under various real-world environmental conditions to ensure robust model performance. Over 1000+ images of different assembly components were gathered, featuring diverse orientations[9], backgrounds, and lighting scenarios. To further enhance the dataset, images were also obtained from secondary sources, including open-source platforms. Captured images exhibited varying resolutions, such as 2560x1920, 3024x4032, and 4032x3024 pixels, providing a wide range of scale and quality to ensure effective model training.
Below table [Fig 2] shows data description and sample component images.

Attribute	Description
Total Images	1000+ Original Images
Data Collection Source	Client's warehouse and Open-source platforms
Data Type	Unstructured Dataset
Image Format	JPG, JPEG, JFIF
Image Type	RGB
Component Diversity	Various sizes, shapes, orientations
Image Resolutions	2560x1920, 3024x4032, 4032x3024 pixels
Image View	Front and multiple angle views
Lighting Conditions	Daylight, low-light, artificial lighting
Background Complexity	Multiple object backgrounds, visual noise, varied colors

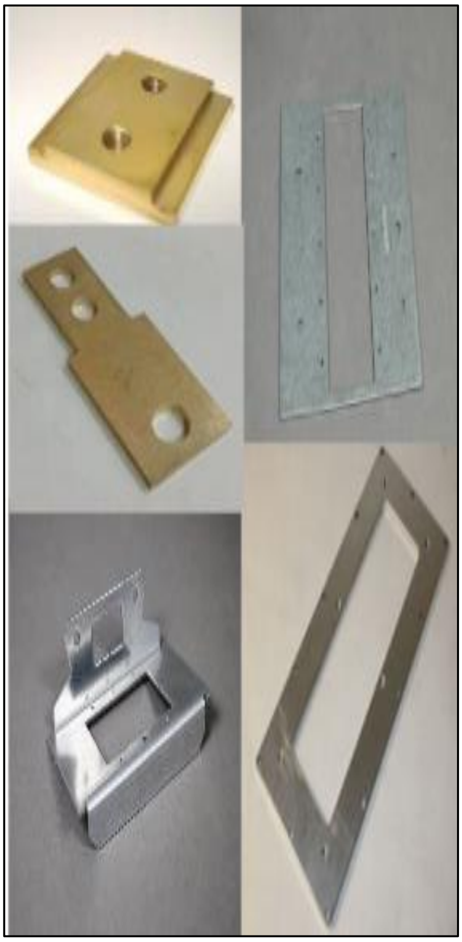

Fig 2 Data and Data Description

B. Data Preprocessing and Augmentation

Effective data preprocessing and augmentation significantly enhanced the performance and accuracy of the electric switchgear component[6] classification model. The preprocessing pipeline involved several critical steps:

- *Image Acquisition and Standardization:*

Images of switchgear components were systematically collected from diverse warehouse environments. To maintain dataset consistency and facilitate model training efficiency, all images were uniformly resized to dimensions of 320x320 pixels.

- *Data Cleaning:*

Images that were blurry, noisy, poorly illuminated, or of substandard quality were identified and either corrected or excluded from the dataset. This ensured high-quality input data, thereby enhancing the accuracy and reliability of the classification model.

- *Dataset Balancing:*

To address the inherent class imbalance[12], images across various switchgear component categories were balanced by ensuring equal representation. Each component class was standardized to a specific number of images, significantly improving model generalization and minimizing potential bias.

- *Data Augmentation:*

Data augmentation techniques[10] were systematically applied to increase the variability and robustness of the dataset. These augmentations simulated real-world operational conditions encountered in warehouse environments: Below table [Table 1] shows data Augmentation techniques applied in training dataset.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1263

Table 1 Augmentation Techniques Applied in Training Dataset

Dataset	Augmentation Techniques Applied (per image)
Training	rotate_image(image, 45°), rotate_image(image, 90°), shear_image(image, 0.05), shear_image(image, 0.10), adjust_brightness(image, 1.5), grayscale_image(image), hue_shift(image, 10), saturation_adjust(image, 1.2)

These preprocessing and augmentation steps collectively ensured high-quality, diverse data inputs, significantly contributing to the robustness and accuracy of the model's predictive capability in operational warehouse scenarios.

➢ *Data Splitting:*

The raw dataset comprised 1000+ original images categorized into 34 distinct classes. After applying comprehensive data augmentation[10], the dataset expanded substantially to a total of 10,336 images. The dataset was then strategically partitioned into three subsets to facilitate efficient model training and validation: Below table [Table 2] shows Data splitting after Augmentation techniques applied.

Table 2 Data Splitting after Augmentation Techniques

Dataset partition	Number of Images
Training set	7616
Validation set	1360
Testing set	1360

This structured split enabled robust training, rigorous validation, and thorough evaluation, ensuring accurate and reliable classification outcomes across diverse real-world warehouse conditions.

C. Model Architecture

The component detection and classification system leverages a comprehensive and structured architecture designed for seamless integration of multiple stages, encompassing data collection, preprocessing, model training, evaluation, deployment, and maintenance. This systematic workflow ensures robust and efficient real-time performance tailored specifically for industrial and operational environments.

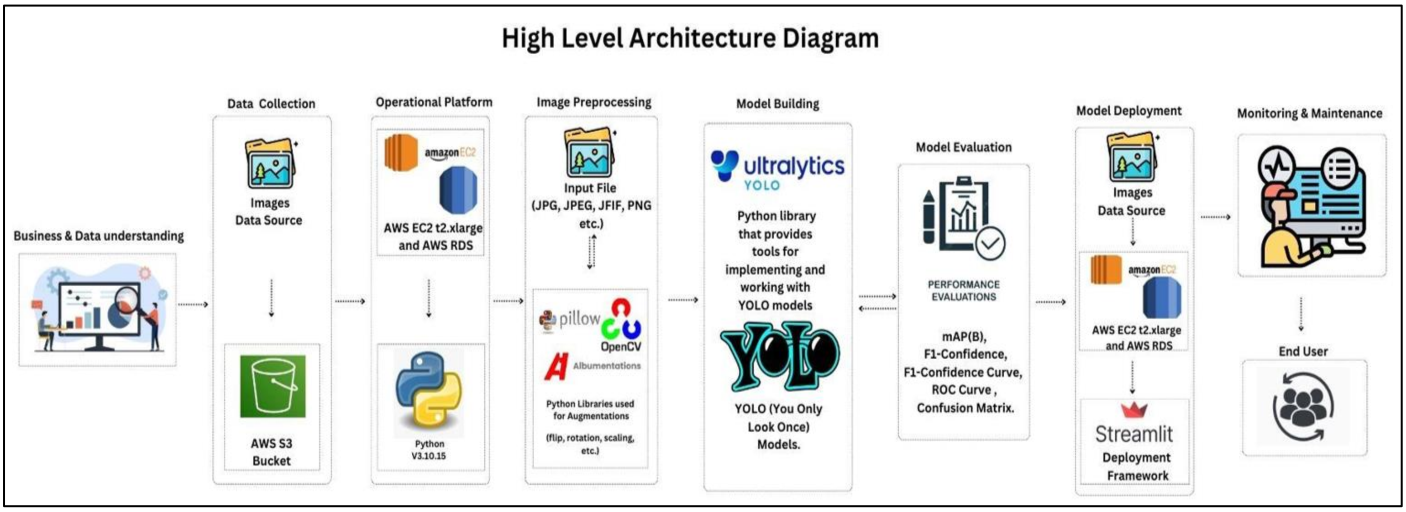

Fig 3 End-to-End High-Level Architecture of the Component Classification System

This architecture demonstrates the end-to-end pipeline, from image acquisition to model evaluation and deployment, ensuring that the system is scalable and efficient for real-world warehouse applications.

The architecture begins with an initial phase of client interaction and extensive business analysis, which clarifies the requirements and scopes the classification objectives. Subsequent data collection utilizes sources such as CCTV and local hardware systems to gather relevant visual data for analysis.

Data collected is then consolidated within an Operational Platform, leveraging tools and technologies like Python scripts and database systems for structured data handling. In the preprocessing phase, raw images undergo essential transformations including resizing, normalization, augmentation, and enhancement techniques to optimize their quality and suitability for model training. Python libraries like OpenCV and Pillow are extensively utilized to streamline and standardize the preprocessing procedures.

The preprocessed images are then fed into the model-building stage, where advanced deep-learning frameworks such as YOLO(You Only Look Once) from Ultralytics and PyTorch are employed. This step involves fine-tuning pre-trained CNN(Convolutional Neural Network) architectures, allowing the extraction of critical image features necessary for accurate detection and classification tasks.

Model evaluation follows a rigorous protocol to validate the effectiveness and accuracy of the developed model. Performance metrics such as precision, recall, F1-score, and mAP(mean Average Precision) are calculated to ensure the model’s reliability using tools like PyTorch evaluation modules.

Once validated, the model is deployed using AWS(Amazon Web Services) cloud environment where streamlit Framework[7] for building AI/ML application, was used enabling easy integration and scalability. Post-deployment, continuous monitoring and maintenance ensure model performance remains optimal, with real-time analytics and feedback loops for continual improvement and adaptation to new data or evolving scenarios.

This complete and detailed pipeline, illustrated in [Table 3], ensures an efficient, accurate, and highly adaptable component classification system suitable for real-world deployment.

D. Model Building

For the classification of components, a comprehensive approach was adopted using advanced deep learning models. The primary focus was on YOLO-cls(You Only Look Once classification)[1] models, specifically YOLOv8n-cls[8], YOLOv8s-cls[8], YOLOv8m-cls[8], ResNet50[2], MobileNetV2[2], MobileNetV3_small_100[3], ViT-base-patch 16-224[4], and ViT_B_16[4], due to their proven capability in real-time object detection and classification tasks. These models were trained using all available component classes, amounting to 34 distinct categories. YOLO's efficient architecture facilitated high accuracy and low latency, making it ideal for real-world industrial applications.

The diverse selection of architectures, especially YOLO variants trained across the full spectrum of 34 classes, ensured a robust comparative study. This systematic evaluation facilitated the identification of optimal models for deployment, balancing high accuracy, efficient inference, and scalability for industrial component classification.

➢ *Model Variants for Image Classification:*

- *YOLOv8n-cls:*

The nano (n) version of YOLOv8 is optimized for fast inference with minimal computational requirements. It consists of approximately 3.2 million parameters and 8.7 billion FLOPs(Floating Point Operations per Second). It is a lightweight model suitable for real-time applications where processing speed is crucial. In this project, it was evaluated to determine whether a compact architecture could still achieve accurate component classification while maintaining high processing speeds, making it viable for rapid operational decision-making.

- *YOLOv8s-cls:*

The small(s) variant of YOLOv8 offers a balanced approach, featuring approximately 11.2 million parameters and 28.6 billion FLOPs. This version aims to achieve a favorable trade-off between computational efficiency and classification accuracy. It was utilized to validate performance across a broader range of scenarios, ensuring scalability and reliability for general industrial deployments.

- *YOLOv8m-cls:*

The medium (m) variant of YOLOv8 provides enhanced performance capabilities with around 25.9 million parameters and 78.9 billion FLOPs. It was selected for scenarios requiring greater accuracy and robustness without sacrificing significant inference speed. Its evaluation aimed to confirm its suitability for complex component classifications where accuracy is paramount.

- *ResNet50:*

ResNet50[2], a residual neural network with 50 layers, is renowned for its ability to efficiently handle complex image classification tasks by mitigating the vanishing gradient problem through residual connections. Its deep architecture enables high accuracy, particularly suitable for detailed component differentiation.

- *MobileNetV2:*

MobileNetV2[2] is optimized for mobile and embedded vision applications. Utilizing depthwise separable convolutions, it provides a highly efficient architecture with fewer parameters, making it ideal for scenarios demanding high-speed inference with minimal computational resources.

- *MobileNetV3_small_100:*

MobileNetV3_small_100[3] represents the smallest and most efficient variant of the MobileNetV3[3] family, designed for high efficiency and performance on resource-constrained devices. It combines innovative design techniques with advanced architecture search for optimized performance in industrial classification tasks.

- *ViT-base-patch 16-224:*

Vision Transformer (ViT)[4] base variant, with a patch size of 16 pixels and input image size of 224 pixels, utilizes self-attention mechanisms instead of traditional convolutional approaches. It excels at capturing global

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1263

context, making it highly effective for detailed component classification tasks requiring global spatial relationships.

- *ViT_B_16:*

Similar to ViT-base, ViT_B_16[4] employs transformer architectures for image recognition, providing robust performance and efficient context understanding through attention mechanisms. This model is particularly suitable for complex scenarios involving detailed visual component classification.

E. Model Evaluation

A rigorous model evaluation strategy was implemented to assess the performance, scalability, and applicability of various deep learning architectures. Models evaluated included YOLOv8 (variants: nano, small, and medium)[8], ResNet50[2], MobileNetV2[2], MobileNetV3_small_100[3], and Vision Transformer variants (ViT-base-patch 16-224 and ViT_B_16)[4]. These models were systematically compared across multiple critical metrics: accuracy, inference speed, resource utilization, ease of deployment, and scalability.

The YOLOv8 models consistently outperformed other architectures in terms of achieving a balance between accuracy and computational efficiency, demonstrating their suitability for real-time industrial image classification tasks. Specifically, the YOLOv8m-cls[8] variant exhibited the highest validation accuracy of 95.20%, clearly indicating its effectiveness in accurately classifying 34 distinct industrial components while maintaining moderate resource demands.

MobileNet variants, including MobileNetV2[2] and MobileNetV3_small_100, showcased notable efficiency and were optimal for scenarios involving constrained computational resources, such as edge or mobile devices. However, they displayed comparatively lower accuracy, limiting their suitability for critical deployments requiring very high precision.

ResNet50[2], although robust in general image recognition tasks, showed moderate performance in this application due to relatively high computational costs and lower achieved accuracy. Vision Transformer(ViT) models offered promising accuracy, particularly ViT_B_16[4], but their computational overhead and inference latency were significantly higher, restricting practical deployment to cloud environments with ample computational resources.

Based on this comprehensive evaluation, YOLOv8m-cls[8] emerged as the most suitable model for practical implementation, offering the optimal balance among accuracy, inference speed, resource efficiency, and scalability. Its performance aligns precisely with the objectives of enhancing component reuse through efficient and precise classification, thereby supporting intelligent resource optimization in industrial applications.

Table 3 Model Selection and Performance Evaluation using Decision Analysis and Resolution (DAR)

Criteria	YOLOv8n (100 epochs)	YOLOv8s (100 epochs)	YOLOv8m (100 epochs)	ResNet50 (70 epochs)	MobileNetV2 (100 epochs)	MobileNetV3_small_100 (100 epochs)	ViT-base-patch 16-224 (100 epochs)	ViT_B_16 (70 epochs)	Decision
Performance & Resource Utilization	Very lightweight, excellent inference speed, lowest GPU usage	Lightweight, fast inference speed, low GPU usage	Moderate, good inference speed, moderate GPU usage	Higher computational demand, moderate inference speed	Efficient computation, low GPU usage	Highly efficient, minimal GPU usage	High computational cost, slower inference	High computation al demand, moderate inference speed	YOLOv8m provides balanced performance suitable for deployment
Ease of Use	Very easy to train, fast training, fast inference	Easy to train, fast training, quick inference	Easy to moderate training, moderate inference speed	Moderate training complexity, slightly slower inference	Easy to train, fast inference, lower complexity	Easy to train, moderate speed	Moderate complexity, longer training times	Moderate complexity, longer training times	YOLOv8s & YOLOv8m offer best balance of ease and capability
Accuracy and Quality	mAP50: 0.9081, Validation Accuracy: 82.22%	mAP50: 0.932, Validation Accuracy: 90.11%	mAP50: 0.9937, Validation Accuracy: 95.20%	mAP50: 0.7022, Validation Accuracy: 50.72%	mAP50: 0.9255, Validation Accuracy: 78.20%	mAP50: 0.8995, Validation Accuracy: 70.31%	mAP50: 0.8150, Validation Accuracy: 77.96%	mAP50: 0.9330, Validation Accuracy: 87.99%	YOLOv8m achieved highest accuracy
System Requirements	Minimal GPU resources required	Low GPU resources required	Moderate GPU resources required	Moderate to high GPU requirements	Low GPU resource required	Very low GPU resources required	High GPU resources required	Moderate to high GPU resources required	YOLOv8n and MobileNet variants suitable for resource-limited environments
Scalability	Ideal for edge and mobile devices	Good scalability, suitable for general deployment	Excellent scalability, optimized for broad deployments	Moderate scalability	Good scalability, suitable for edge deployment	High scalability, optimized for mobile/edge deployments	Moderate scalability, optimal for cloud environments	Moderate scalability, ideal for cloud deployments	YOLOv8m & YOLOv8s best suited for scalable applications

This analysis visually highlights how each model compares regarding accuracy, resource efficiency, ease of use, system requirements, and scalability. YOLOv8m-cls[8] is clearly identified as the optimal model choice due to its superior balance of high accuracy, moderate computational requirements, ease of deployment, and broad scalability, thus confirming its suitability for real-time industrial applications aimed at component reuse and intelligent resource optimization.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1263

F. Model Hyperparameter Tuning

To improve the performance of the YOLOv8m-cls model, hyperparameter tuning was carried out by adjusting key settings such as learning rate, weight decay, optimizer type, and data augmentation techniques[10]. The goal was to increase the model's accuracy while keeping it stable and efficient for real-world industrial use. Different tuning strategies were tested by changing how the model learns, how much regularization is applied, and how image enhancements like flipping and brightness adjustments are used. These tuning methods are summarized in Table 4.

The objective of this hyperparameter tuning was to achieve the best possible trade-off between accuracy, computational efficiency, and model stability for practical deployment in industrial applications

Table 4 Hyperparameter Tuning for the Best Model (YOLOv8m-cls)

Sets	Parameters	Strategy Description
Set 1	epochs=100, imgsz=640, lr=1e-4, weight_decay=1e-4, optimizer=Adam, augmentation=True (Flip, HSV adjustments, Mosaic), warmup_epochs=5	Balanced learning rate with standard augmentation for accuracy
Set_2	epochs=100, imgsz=640, lr=5e-5, weight_decay=5e-4, optimizer=Adam, augmentation=True (HSV adjustments, Mosaic, Mixup), warmup_epochs=7	Lower learning rate, strong regularization, and augmented data for improved stability
Set_3	epochs=100, imgsz=640, lr=1e-4, weight_decay=1e-5, optimizer=Adam, augmentation=False, warmup_epochs=3	Minimal augmentation with a low learning rate for stability

After testing various tuning approaches, the best model setup showed significant improvements``in accuracy and detection performance. As seen in Table 5, the validation accuracy increased from 95.20% to 97.35%, making the model more reliable in correctly identifying switchgear components. The mAP50(Mean average precision calculated at an intersection over union (IoU) threshold of 0.50) score improved from 0.963 to 0.995, meaning the model became more precise in detecting objects. The precision increased from 0.926 to 0.964, reducing incorrect classifications, while the recall improved from 0.911 to 0.959, ensuring fewer missed detections. Although training time rose slightly by 12%, the model's inference speed improved, allowing faster real-time classification. The GPU usage remained moderate, meaning it still runs efficiently without requiring excessive computing power. Additionally, scalability improved from good to excellent, making the model more adaptable for large-scale industrial applications.

Table 5 Model Performance Metrics after Hyperparameter Tuning

Metric	Before Tuning (YOLOv8m baseline)	After Tuning (Set 1)
mAP50	0.963	0.995
Validation Accuracy (%)	95.20%	97.35%
Precision	0.926	0.964
Recall	0.911	0.959
Training Time	Moderate	Slightly increased (+12%)
Inference Speed	Moderate	Slightly faster
GPU Utilization	Moderate	Moderate
Scalability	Good	Excellent

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1263

These improvements show that fine-tuning the model's parameters significantly enhances its performance, making it highly accurate, efficient, and reliable for warehouse inventory tracking[11]. By reducing human errors and automating the classification process, this optimized AI model helps industries manage their components more effectively, improving productivity and resource utilization.

The optimized YOLOv8m-cls model (Set 1) achieved remarkable improvements in accuracy, precision, and recall, demonstrating its efficacy in real-time industrial applications. This tuning provided a robust model, capable of precise and rapid component classification, critical for enhancing component reuse and intelligent resource optimization.

III. MODEL DEPLOYMENT

After testing and fine-tuning different models, the YOLOv8m-cls (Set 1) model was deployed in an AWS cloud environment to ensure scalability, real-time processing, and easy accessibility. The Streamlit framework[7] was used to build an interactive AI-powered web application, allowing users to easily upload images of switchgear components and receive instant classification results. Figure 4 shows streamlit's lightweight and efficient design enabled seamless integration with the trained YOLO model, ensuring smooth real-time inference.

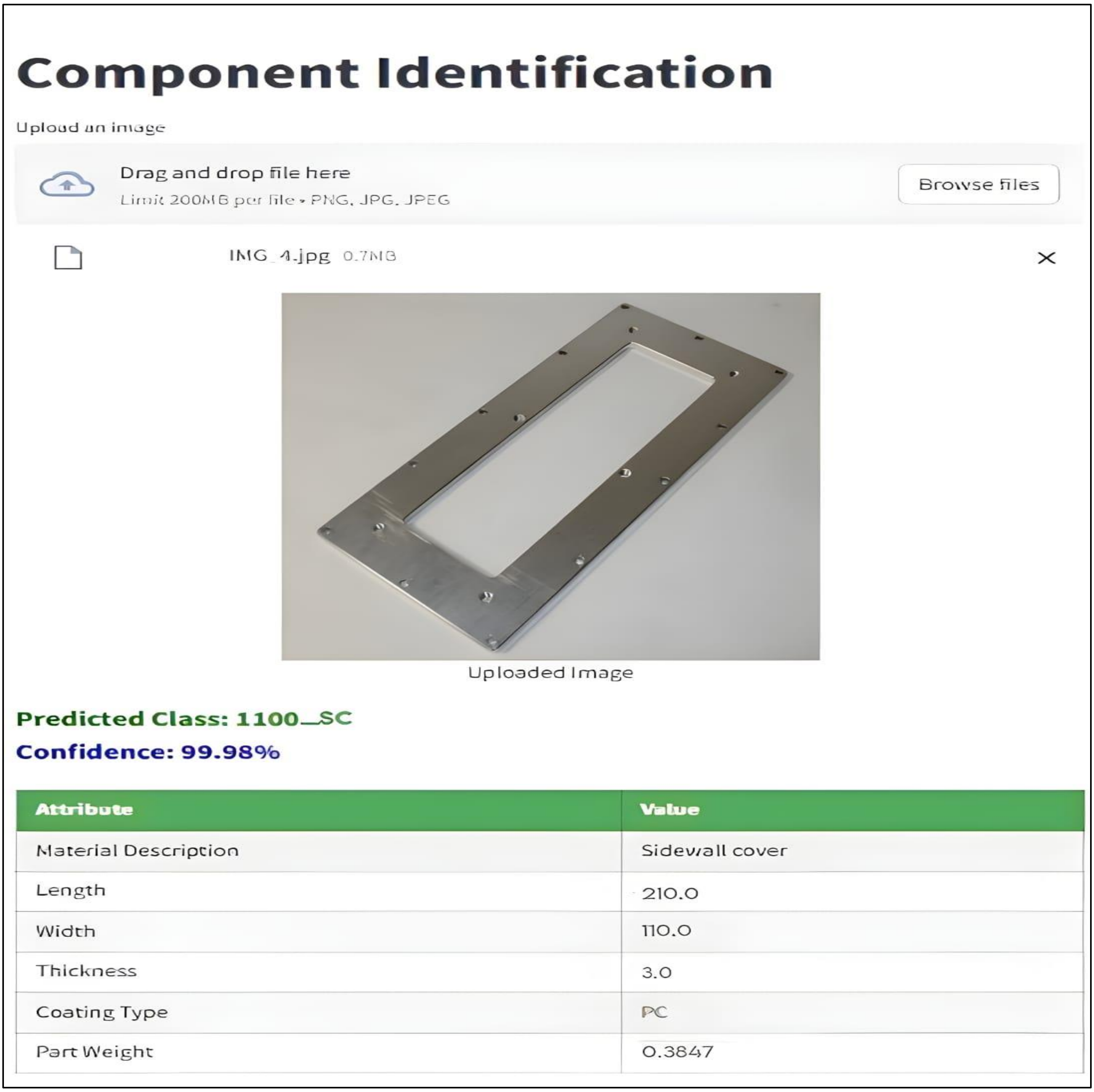

Fig 4 A Streamlit Framework for Automated Component Identification and Attribute Extraction

The deployed system allows warehouse staff to upload images, which are processed in real time to classify the component. Along with classification, the system retrieves and displays key attributes such as Material Description, Length, Width, Thickness, Coating Type, and Part Weight to provide comprehensive inventory insights.

This cloud-based deployment ensures that warehouses can automate component classification, reduce manual efforts, and improve inventory accuracy while maintaining seamless and efficient operations. By leveraging AWS services, the system supports real-time tracking, minimizes classification errors, and enhances overall warehouse management[11], making AI-driven inventory automation an essential advancement in modern industrial operations.

IV. CONCLUSION

This research successfully developed an AI-powered system for classifying and identifying electric switchgear components, significantly improving warehouse inventory management. Traditional manual classification methods often result in misidentification, misplaced inventory, and operational inefficiencies. By leveraging YOLO-based deep learning models, the proposed system automates component identification with high precision, reducing human errors and increasing efficiency.

The model was trained on a comprehensive dataset, incorporating diverse environmental conditions to ensure robustness. Data augmentation techniques further enhanced the model's generalization, making it effective in real-world warehouse settings. Among the tested models, YOLOv8m-cls(Set1) emerged as the best-performing model after hyperparameter tuning, significantly improving classification performance.

Before tuning, the YOLOv8m-cls baseline model achieved a validation accuracy of 95.20%, with a precision of 0.926 and a recall of 0.911. The mAP50 score was 0.963, indicating strong detection performance. The model exhibited moderate inference speed and scalability, making it suitable for deployment but with room for optimization.

After tuning (Set 1), the model demonstrated notable improvements, achieving a validation accuracy of 97.35%, precision of 0.964, and recall of 0.959. The mAP50 score increased to 0.995, showcasing enhanced detection performance. Although training time increased by 12%, inference speed improved, and scalability was enhanced from good to excellent.

To ensure scalability and accessibility, the model was deployed in an AWS cloud environment using the Streamlit framework, allowing warehouse staff to upload images and receive instant classification results. This cloud-based approach enables real-time tracking, seamless inventory updates, and smooth integration with warehouse management systems.

By implementing this AI-driven system, warehouses can reduce manual effort, enhance tracking accuracy, and optimize workflows. The findings of this research demonstrate the transformative impact of AI in industrial inventory management, paving the way for future advancements in automated component classification, predictive analytics, and mobile-friendly AI applications.

FUTURE SCOPE

The model can also be used on tablets or mobile phones by turning this AI application into a mobile app. This would allow workers to scan and classify switchgear components instantly, making inventory management easier and more efficient. The mobile version would run smoothly using lightweight AI models, ensuring fast results even without an internet connection.

In the future, the system could also be improved to identify multiple components in a single image. This would help warehouse staff scan entire shelves at once instead of classifying parts one by one, saving time and reducing manual work. Additionally, AI-powered image enhancement techniques could be integrated to improve detection accuracy in low-light or cluttered environments.

Another potential enhancement is the incorporation of augmented reality (AR) to overlay real-time classification and inventory details directly on the device's screen. This would help workers quickly locate and verify components without needing to cross-check with manual records.

Furthermore, integrating cloud connectivity and predictive analytics could enable real-time stock monitoring, automated restocking alerts, and data-driven decision-making for inventory optimization. Future versions might also support voice commands, making hands-free operation possible for increased efficiency in fast-paced warehouse environments.

These advancements will make the system more powerful, user-friendly, and indispensable for modern warehouse operations.

ACKNOWLEDGMENTS

We sincerely thank 360 DigiTMG for providing us with the opportunity to work on this project. We also appreciate the guidance and support of our partners throughout this research. Additionally, we acknowledge the use of the CRISP-ML(Q) framework and ML Workflow, which are openly available on the official 360 DigiTMG website and used with their explicit consent.

REFERENCES

[1]. Naqif Fared Nor, Hazlyna Harun. An Experiment on Lung Disease Classification using YOLOv8. DOI: https://doi.org/10.58915/amci.v13i3.626.
[2]. Mohammad Rafka Mahendra Ariefwan, I Gede Susrama Mas Diyasa, Kartika Maulidya Hindrayani.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1263

InceptionV3, ResNet50, ResNet18 and MobileNetV2 Performance Comparison on Face Recognition Classification. DOI:https://doi.org/10.56480/jln.v4i1.990.

[3]. Mohamed Abd Elaziz 1,2,*ORCID,Abdelghani Dahou 3,Naser A. Alsaleh 4ORCID,Ammar H. Elsheikh 5,*ORCID,Amal I. Saba 6 andMahmoud Ahmadein 4,5 ORCID. Boosting COVID-19 Image Classification Using MobileNetV3 and Aquila Optimizer Algorithm. DOI: https://doi.org/10.3390/e23111383.

[4]. Chu Myaet Thwal1 , Ye Lin Tun1 , Minh N. H. Nguyen2 , Eui-Nam Huh1 , Choong Seon Hong1* 1Kyung Hee University 2Vietnam-Korea University of Information and Communication Technology. CLIP-PING: Boosting Lightweight Vision-Language Models with Proximus Intrinsic Neighbors Guidance. DOI: https://doi.org/10.48550/arXiv.2412.03871.

[5]. Stefan Studer , Thanh Binh Bui , Christian Drescher , Alexander Hanuschkin , Ludwig Winkler , Steven Peters and Klaus-Robert Müller Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology . DOI:https://doi.org/10.3390/make3020020.

[6]. R. Umamaheswari a & R. Sarathi a a Department of Electrical Engineering , Indian Institute of Technology Madras , Chennai, India Published online: 07 Oct 2011. Identification of Partial Discharges in Gas-insulated Switchgear by Ultra-high frequency Technique and Classification by Adopting Multi-class Support Vector Machines. DOI: https://doi.org/10.1080/15325008.2011.596506.

[7]. Himangi Dani, Pooja Bhople, Hariom Waghmare, Kartik Munginwar, Prof. Ankush Patil Review on Frameworks Used for Deployment of Machine Learning Model DOI:https://doi.org/10.22214/ijraset.2022.40222.

[8]. Utsha Saha∗† , Imtiaj Uddin Ahamed‡, Md Ashique Imran§, Imam Uddin Ahamed¶, Al-Amin Hossain‡, Ucchash Das Gupta YOLOv8-Based Deep Learning Approach for Real-Time Skin Lesion Classification Using the HAM10000 Dataset. DOI:10.1109/HealthCom60970.2024.10880715.

[9]. Ilia V. Safonov, Ilya V. Kurilin, Michael N. Rychagov, Ekaterina V. Tolstaya, "Content-Based Image Orientation Recognition," in *Adaptive Image Processing Algorithms for Printing*, 2018. DOI: 10.1007/978-981-10-6931-4_12.

[10]. Marcus D. Bloice, Christof Stocker, Andreas Holzinger, "Augmentor: An Image Augmentation Library for Machine Learning," *arXiv preprint arXiv:1708.04680*, 2017. DOI: 10.48550/arXiv.1708.04680.

[11]. Niels Faber, René de Koster, and Ale Smidts, "Organizing Warehouse Management," *International Journal of Physical Distribution & Logistics Management*, vol. 43, no. 9, pp. 764-784, 2013. DOI: 10.1108/IJPDLM-11-2011-0203.

[12]. Haibo He and Edwardo A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009. DOI: 10.1109/TKDE.2008.239.

[13]. Chawla, Nitesh V., et al. "SMOTE: Synthetic Minority Over-sampling Technique." *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. DOI: 10.1613/jair.953.

[14]. Japkowicz, Nathalie, and Shaju Stephen. "The Class Imbalance Problem: A Systematic Study." *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429-449, 2002. DOI: 10.3233/IDA-2002-6504.

[15]. He, Haibo, and Yunqian Ma. "Imbalanced Learning: Foundations, Algorithms, and Applications." *IEEE Press/Wiley*, 2013. DOI: 10.1002/9781118646106.