Development of a Lightweight CNN Model for Metal Surface Defect Detection in Resource-Constrained Environments

Dennis Mashoko[1], Cross Gombiro[2]

Abstract

This research developed a customized, deep learning-based Convolutional Neural Network (CNN) model for automating metal surface defect detection in resource-constrained industrial environments, using the Zimbabwean steel industry as a case study. Manual inspection methods are labour-intensive and prone to error, emphasizing the need for an efficient, accurate solution. A research design approach combined with experimental design methods was employed to develop and evaluate the proposed CNN-based defect detection model. A lightweight CNN was designed and optimized for edge deployment, integrating transfer learning and active learning to improve performance under limited computational resources. The NEU Metal Surface Defects Database (1,800 images across six defect types) was used for training, and 200 Zimbabwean steel images were incorporated to adapt the model to local conditions. Transfer learning using MobileNetV2 enhanced both feature extraction and fine-tuning. Active learning with Monte Carlo dropout uncertainty sampling iteratively refined the model by selecting and re-labelling uncertain samples. The experimental results show significant performance improvements; the baseline L-CNN achieved 97.22% accuracy, transfer learning raised accuracy to 98.61%, and the final model with both techniques reached 99.4% with minimal loss (0.0042). Moreover, the results reflect that the L-CNN model is well-suited for contexts in developing countries like Zimbabwe, where challenges with computational resources and dataset availability are prevalent. The customized L-CNN sets a new benchmark for automated steel defect detection in developing regions, thus reducing inspection costs and improving quality control with high accuracy and efficiency.

Keywords: Convolutional Neural Networks, Steel Industry, Machine Learning, Transfer Learning, Active Learning

1. INTRODUCTION

The steel industry in Zimbabwe plays a pivotal role in the development of the country's construction, manufacturing, and infrastructure sectors (Gudukeya et al., 2019) Recently, the investment by Tsingshan Holding Group's Dinson Iron and Steel Company of about $800 million in the country reflects that the industry has a high potential to grow in the future. However, despite these investments, the industry is experiencing challenges in maintaining consistent quality standards, specifically in the detection of metal surface defects on flat steel products.

Traditional quality control methods are usually made up of manual visual inspections, are labour-intensive, subjective, and prone to errors due to both human fatigue and inconsistencies (Ghobakhloo et al., 2021). The reliance on outdated techniques leads to increased rejection rates, production delays, and high operational costs (Gudukeya et al., 2019) . On the other hand, advanced economies have adopted automated machine vision systems to enhance metal surface defect detection accuracy and efficiency. Convolutional Neural Networks (CNNs) have emerged as the leading model for visual recognition tasks, including industrial metal surface defect detection (Aboulhosn et al., 2023). The introduction of LeNet-5 in the late 1980s has seen CNNs experiencing a significant evolution, with milestones like AlexNet in 2012 demonstrating deep learning's potential. Modern lightweight architectures like the MobileNet and EfficientNet, are optimized for resource-constrained environments, making them ideal for applications in developing countries. Recent studies have reported accuracy rates exceeding 95% in automated defect detection systems. However, such advancements are largely absent in resource-constrained environments like the Zimbabwe Steel Industry because of resource limitations, lack of large annotated datasets, high costs of industrial Artificial Intelligence Systems, and a shortage of local expertise.

The scenario highlights a clear research gap, that is, the need for a cost-effective, lightweight CNN solution tailored for resource-constrained environments like the Zimbabwean Steel Industry, where computational infrastructure is limited and real-time inspection is essential. The existing generic solutions usually require high-end GPUs, extensive training of data, and stable high-speed networks,

[1] Systems Engineer, Information and Communication Technology Department, University of Zimbabwe, *dsmashoko@gmail.com*
[2] Lecturer, Computer Science Department, University of Zimbabwe, *cgombiro@ceic.uz.ac.zw*

which are often unavailable in resource-constrained environments. To address this gap, the proposed research aims to design and implement a lightweight CNN model optimized for resource-constrained environments. The model integrates transfer learning and active learning techniques into the CNN model, as a way to reduce training costs and also leverage small, localized datasets focused on defect types prevalent in Zimbabwe. This approach aspires to enhance metal surface defect detection accuracy and processing speed, thereby promoting affordable Artificial Intelligence adoption in resource-constrained environments like the Zimbabwe Steel Industry and ultimately improving quality control and competitiveness in the region.

This research endeavours to discover how effective are lightweight CNNs in detecting metal surface defects in resource-constrained environments compared to traditional inspection methods, to what extent can transfer learning and active learning can improve the detection accuracy when they are used on both small and localized datasets, and what is the trade-off between the computational efficiency and accuracy of the proposed lightweight CNN model. The researchers propose that integrating transfer learning and active learning into a lightweight CNN architecture will significantly improve the defect detection accuracy, at the same time reducing computational and data requirements in resource-constrained environments.

2. LITERATURE REVIEW

Steel production is crucial for the economic growth of many developing countries, especially in Africa. Detecting and addressing surface defects in metal products is a major difficulty (Gudukeya et al., 2019). The researchers (Ghobakhloo et al., 2021) explain the disadvantages of the manual metal surface defect systems as laborious, subjective and error-prone. The introduction of an automated solution that employs Convolutional Neural Networks (CNNs) has the potential to transform the quality control procedures in the metal industry (Zubayer et al., 2023). The researchers used the Inclusion and Exclusion Criteria. The criteria incorporated papers that focused on metal surface defect detection and classification using deep learning and convolutional neural networks. The researchers also included papers that talked about manual quality inspection of steel and papers that talked about challenges in the Zimbabwe Steel Industry. Papers that did not specifically report on steel surface defect detection or that used traditional machine learning methods instead of deep learning were excluded by the researchers. Furthermore, papers that did not specifically talk about challenges in the Zimbabwe Steel Industry and about manual inspections of metal were also excluded from the research.

A qualitative systematic review was carried out and Library Databases and journals that were used for the Literature review were taken from a variety of sources, namely IEEE Access, The International Journal of Advanced Manufacturing Technology, IEEE Region 10 Symposium (TENSYMP), Journal of Imaging, Optics and Lasers in Engineering, Scientific Reports, Journal of Manufacturing Technology Management, International Journal of Production Research, IEEE Transactions on Instrumentation and Measurement, Metals, Applied Sciences (Switzerland), IEEE Transactions on Cybernetics, Journal of Non-destructive Evaluation, Procedia Manufacturing, Revue d'Intelligence Artificielle, Algerian Journal of Signals and Systems, Journal of Physics: Conference Series, Journal of Big Data, ISIJ International, and Journal of Electrical Systems, 10th International Conference on Intelligent Control and Information Processing (ICICIP), American Institute of Physics, Springer Link, and Research Gate. The varied sources, namely journals, conference proceedings, and databases, reflect the comprehensive nature of the research in metal surface defect detection in manufacturing industries using deep learning techniques.

The search strategy that was used incorporated search terms which included keywords such as steel surface defect, deep learning, convolutional neural network, metal surface inspection and surface metal defect detection, challenges in the Zimbabwe Steel Industry, and manual inspection of metal or steel. The search was focused on finding the most relevant and recent research papers published in reputable journals and conference proceedings. The literature on CNN-based metal surface defect detection is extensive, with numerous studies exploring various models, techniques, and applications. This section summarizes key findings from recent research, highlighting the methodologies used and their limitations in the context of resource-constrained environments like the Zimbabwean Steel Industry.

2.1 SUMMARY OF LITERATURE REVIEW

The literature review reveals a significant focus on the general application of CNN models in metal surface defect detection. However, most studies fail to address the unique challenges of resource-constrained environments that are similar to the Zimbabwean Steel Industry. These environments have limited computational resources, evolving manufacturing processes that need customized solutions tailored to their local conditions.

Table 1, below shows the summary of the key findings of the research.

Table 1. Summary of the literature review.

Study and Key Findings and Methodologies Used	Limitations
(Aboulhosn et al., 2023) evaluated several CNN-based semantic segmentation approaches, including U-Net and Feature Pyramid Network (FPN) architectures with various backbones. The study also incorporated data augmentation techniques to improve the robustness of the models. The proposed approaches demonstrated superior performance in accurately detecting and segmenting different types of steel surface defects.	The study relied on a large labeled dataset that was used with high-end GPUs. However, in low-resource environments, such as small-scale steel plants, the resources and infrastructure may not be available. There is a need for a lightweight, data-efficient CNN model that can be deployed on low-power devices without sacrificing accuracy.
(Chen et al., 2025) proposed a lightweight steel surface defect detection framework based on YOLOv9s, which integrates the C3Ghost module, SCConv module, and CARAFE up-sampling operator to improve both accuracy and efficiency. Experimental results showed that the model achieved a high accuracy and robustness, effectively detecting multi-scale steel surface defects.	The model heavily relies on deep learning with moderately high computational requirements, which limits deployment in resource-constrained or edge device environments. The study does not employ transfer or active learning, which could greatly reduce dependency generic datasets
(Cumbajin et al., 2023) focused on classifying surface defect detection approaches based on the type of surface (metal, building. Classification was the most prevalent problem. Transfer learning was used in 83.05% of the studies, and data augmentation in 59.32%.	The research focused on regions with mature industrial automation, it lacked guidance for scenarios with limited datasets, unstable power or low-end hardware.
(Amin & Akhter., 2020) the use of U-NET and Deep Residual U-NET for high-accuracy metal surface defect detection.	The research required costly, expert-labeled datasets and high-resolution imaging. There is a need for the adoption of Active learning and supervision for training on limited labeled data to reduce annotation effort and costs.
(Fu et al., 2019) came up with a CNN model that focuses on training low-level features and incorporates multiple receptive fields, achieving high-accuracy recognition on a diverse testing dataset.	The Performance of the CNN model depends on controlled lighting and curated datasets. capable of handling heterogeneous conditions. The research failed to address the detection of diverse metal surfaces and low-quality imaging.
(Guan et al., 2020) formulated a novel recognition algorithm that leverages deep learning, feature visualization, and quality evaluation.	Lab experiments assumed uniform process conditions, the heterogeneous manufacturing processes employed do not show case the typical real-world scenarios. There is a need of an adaptive model that can learn across variable

	production conditions using incremental updates.
(He et al., 2020) developed an end-to-end defect detection method for steel plate surfaces, combining deep learning with multilevel feature fusion. Defect Detection Network (DDN) was used, and it employs a baseline Convolutional Neural Network (CNN) to generate feature maps, followed by a Multilevel Feature created a new defect detection dataset called NEU-DET was created for training and evaluating the proposed system.	The Dataset that was used was small and controlled; also, generalization to noisy industrial conditions is not clear. There is a need for Data-efficient models capable of generalizing from small datasets that use both transfer learning and active learning to maximize accuracy on small, localized datasets.
(Hosseinzadeh et al., 2025) evaluated various CNN-based image classification models, namely MobileNet, ShuffleNet, DenseNet, RegNet, NasNet, ResNet, GoogleNet, and a custom CNN for detecting steel surface defects using the NEU-CLS-64 dataset. Performance is assessed via metrics such as accuracy, precision, sensitivity, and F1 score. The results showcase that RegNet achieves the highest overall performance.	The study provides a comprehensive benchmarking of CNN models, it mainly focused primarily on computationally heavy models and does not address deployment in resource-constrained industrial environments.
(Kateb et al., 2020) came up with a novel deep learning-based defect detection system that uses a baseline convolutional neural network (CNN) to generate feature maps, after a multilevel feature fusion network (MFN) that effectively combines hierarchical features for improved location details of defects.	The model gains accuracy at the cost of high memory and computing resources. Edge devices in small steel plants cannot handle this efficiently, thus a need of a Lightweight CNN architectures with reduced computational requirements.
(Kateb et al., 2024) formulated a deep learning approach based on a pre-trained NASNet-Mobile CNN to address the challenges of steel surface defect classification, including limited memory, long execution times, and insufficient faulty samples. The modified model achieved 99.51% accuracy in detecting 6 defect types.	The model assumes GPU-assisted fine-tuning and curated datasets; the research lacks real-time inference on low-edge CPUs and adaptation to new defect types with minimal data.
(Khebli et al., 2024) came up with a deep CNN model using the pre-trained NasNet-Mobile backbone architecture to classify defects in hot-rolled steel strips. The model achieved 99.30% accuracy using 26 times less data than previous approaches, addressing challenges of limited memory, long execution times, and scarce faulty samples.	The research used a controlled dataset; however, due to this, the performance may drop with varying sensors, surfaces, or environmental noise. There is a need for a robust, edge-deployable model with minimal retraining requirements,
(Konovalenko et al., 2020) investigated the use of residual neural networks, looking specifically at the CNN ResNet50 model, for classifying surface defects. The proposed models can be used to detect surface defects with high accuracy, with the best model based on ResNet50 achieving an average classification accuracy of 96.91% on the test data.	The model requires high computational resources and power; also, there is limited evidence of domain generalization. There is a need for a pruned or lightweight architecture for low-power deployment.
(Lee et al., 2019) proposed a deep convolutional neural network (CNN) approach for diagnosing steel surface defects, along with the use of class	The research incorporated interpretability but was not tested on edge devices or under variable lighting/motion blur. There is a need for a

activation maps to provide visual explanations. The deep CNN model achieved very high performance, with 99.44% accuracy and 0.99 F-1 score, outperforming traditional machine learning algorithms like support vector machines and logistic regression.	deployable, interpretable CNN suitable for operator-in-the-loop scenarios and environments.
(Li et al., 2025) came up with an improved YOLOv8-based model for metal surface defect detection and introduced modules such as C2f_GhostDynamic for reduced model size, CARAFE and SPPELAN pyramid pooling for enhanced feature extraction, and RFAHead/AKConv for adaptive detection of defect shapes	Accuracy remains moderate compared to state-of-the-art lightweight CNNs; the model was evaluated solely on NEU-DET without adaptation to localized or industry-specific data. mechanisms to enhance its performance under limited labelled data.
(Lin and Wibowo., 2021)collected datasets with varying lighting conditions and calculated the comprehensive assessment score for each. Trained YOLO, SSD, and Faster R-CNN models on the datasets and evaluated their F2-scores.	The model assumes the ability to recapture datasets in controlled conditions; however, it fails in inconsistent factory settings. There is a need for a self-supervised or adaptive-pipelines for variable lighting and capture conditions.
(Lu et al., 2025) proposed FEP-YOLO, a lightweight CNN framework for steel surface defect detection to be used on resource-constrained devices. The model employs a FasterNet backbone, EMA attention mechanism, and PC-Detect detection head to reduce model complexity.	CNN models like FEP-YOLO show promise for defect detection on edge devices, but are limited because of low accuracy (80.9% mAP). They also lack transfer and active learning and have no adaptation to localized industrial data.
(Lu and Lee., 2023) developed a deep learning model that is deployable on devices without GPUs and on edge computing environments. The model achieves a balance between accuracy, real-time performance, and lightweight design.	The model has primarily been evaluated on publicly available datasets and may not generalize to steel surfaces in specific industrial contexts, resource-constrained environments with unique textures, lighting variations, or rare defect types.
(Ma et al., 2025) proposed a lightweight YOLOv8-based algorithm for steel surface defect detection. The model addressed issues of low precision, high model complexity, and computational cost. The model integrated the GhostNet as the backbone to reduce parameters, the MPCA attention mechanism for enhanced multi-scale feature extraction, and the SIoU loss function for improved bounding box regression.	The improved YOLOv8 model has not been fully validated in resource-constrained industrial environments that have low-quality, noisy, or low-contrast steel surfaces, which are common in developing regions. Small defects, subtle scratches, or localized surface irregularities may still be missed.

(Mao and Gong., 2025) proposed a LRT-DETR, a lightweight object detection algorithm for steel surface defect detection based on RT-DETR. Major innovations to note include replacing BasicBlock modules with MobileNetV3 for efficient feature extraction, integrating depth-wise separable convolution (DWConv) and VoVGSCSP for enhanced feature fusion, and using MPDIoU as a novel bounding box similarity measure to improve detection accuracy.	The accuracy remains moderate compared to modern lightweight CNN models, achieving less than 90% accuracy on similar datasets. Evaluation is limited to standard datasets without local industrial adaptation. The model does not employ transfer learning nor use active learning, which could improve performance with limited labeled data.
(Liu et al., 2020) designed a lightweight deep-learning model called Concurrent Convolutional Neural Network (ConCNN) for real-time steel surface defect classification.	The model was measured in real-time on desktop GPUs lacks benchmarks on low-power devices. There is a need for Real-time inference on low-resource hardware
(Mery., 2020) highlighted the importance of using a large dataset with synthetic defects and real background data for effective CNN model training.	The Synthetic-to-real gap remains as the model fails on novel surface finishes and low-quality camera images. There is a need for a data-efficient augmentation strategy and transfer learning to be used for small datasets.
(Otieno et al., 2023) explored the use of machine learning (ML) and deep learning (DL) techniques for detecting surface defects in hot-rolled steel. In machine learning, the GLCM feature extraction technique outperformed LBP and HOG. In the deep learning domain, the CNN with the VGG19 model exhibited the best performance, with precision and recall metrics of 0.97 and 0.96, respectively.	The model uses Proprietary datasets; it shows unclear reproducibility across heterogeneous environments. There is a need for a generalizable, lightweight CNN for variable industrial conditions.
(Ren et al., 2018) , established a generic deep-learning-based approach for automated surface inspection, utilizing feature transferring to reduce the need for large training datasets.	The model has a heavy backbone and limited edge optimization. The early transfer methods used are not practical for low-power devices. There is a need for Edge-friendly transfer learning and lightweight CNNs.
(Selamet et al., 2022) developed a novel method combining Saliency Fusion Segmentation and Faster R-CNN for defect detection. Achieved 0.83% mean average precision on NEU Surface and KSDD2 datasets.	The Pixel-level labeling technique that was used is costly and also pipeline complex. There is a need to employ Active learning to reduce the annotation burden.
(Shan et al., 2025) proposed the RSM-YOLOv11 model, which is an improved YOLOv11-based algorithm for steel surface defect segmentation, and they are focusing on both accuracy and computational efficiency. Notable achievements included the Space-to-Depth Convolution (SPD-Conv) for preserving fine-grained features.	The RSM-YOLOv11 improved segmentation accuracy and reduced computational cost, however, it still relies on high-quality annotated datasets for effective training. This may limit the models applicability in industrial settings that have limited labelled data.
(Shorten and Khoshgoftaar., 2019) highlighted the use of various augmentation techniques to enhance model performance and leverage limited datasets.	Many of the models' augmentations are label-preserving but not physics-aware. The model also fails to handle glare, secularities, and motion artifacts. There is a need for a Process-aware, physics-informed augmentation for low-cost cameras.

(Luyu Sun and Yujun Zhang., 2025) came up with an improved YOLOv10 model for steel surface defect detection, introducing several novel modules: C3_Star_EMA to enhance feature representation using star operations and EMA attention, MobileOneBlock to reduce parameters and computation through multi-branch convolution, and C2fAFF with AFF attention for optimized multi-scale feature fusion.	Accuracy remains moderate compared to state-of-the-art lightweight CNNs, and the model is tested only on the NEU-DET dataset without adaptation to localized industrial data. The model does not employ transfer learning or active learning to enhance performance with limited labeled samples.
(Tao et al., 2018) presents a unique cascaded autoencoder (CASAE) architecture for segmenting and localizing metallic surface defects. The cascading network transforms the input defect image into a pixel-wise prediction mask based on semantic segmentation.	The model requires Dense annotations; inference is computationally heavy. There is a need for Semi-supervised, lightweight models suitable for edge deployment.
(Wang et al., 2021) proposed an improved approach combining an enhanced ResNet50 model, deformable convolution networks, and improved cutout techniques, achieving 98.2% accuracy and faster performance.	The model has complex architectures that consume a lot of compute resources. They are not robust to motion blur or to line-speed variations. There is a need for Low-latency, motion-robust CNNs for resource-constrained environments
(Wang et al., 2025) introduced YOLO-LSDI, which is an improved YOLOv11-based algorithm for steel surface defect detection aimed at enhancing detection accuracy, generalization and robustness for industrial applications.	The model might encounter difficulties in environments with very low-contrast surfaces or defect types it hasn't seen before that aren't well represented in the training data.
(Wang et al., 2025) CSCP-YOLO presents a lightweight deep learning model for real-time steel surface defect detection, which addresses challenges such as multi-scale defects, small targets, and complex backgrounds. The framework integrates CGSF (C2f-GSConv-FasterNet) for small-target detection, SPPF-DW for efficient multi-scale feature fusion, and CCE (C2f-CIB-EMA) for enhanced feature extraction through optimized weight sparsity.	The CSCP-YOLO model achieves high-speed, accurate detection; However, it lacks evaluation on localized or developing-region steel datasets, limiting its applicability to real-world industrial environments like Zimbabwe.
(Wei et al., 2020) developed a method leveraging Faster R-CNN, Weighted RoI Pooling, deformable convolution, FPN, and Strict-NMS to address challenges like missed detection of small defects and adaptability to irregular-shaped defects.	Compute-intensive model that is task-specific. There is a need for a Pruned, lightweight architecture for small-scale deployment that is able to adapt to its environment.
(Xia et al., 2025) came up with a model called LSDNet, which is a specialized, lightweight deep learning model for strip-steel surface defect detection on resource-constrained embedded devices. The model was built on MobileNetV2, and it incorporates a new feature extraction module, including SPD-Conv for small-target feature learning and ECANet for enhanced feature extraction. A parameter-free attention mechanism (SimAM) is applied to improve recognition accuracy.	Although LSDNet achieves high accuracy with low computational cost, it does not incorporate transfer learning or active learning, which directly limits its adaptability to new or scarce datasets. The model has not been tested on localized industrial steel datasets, which may differ in defect types and image quality; devices in developing-region factories remain invalidated.

(Yang et al., 2025b) presents LEIS-Net, a lightweight and efficient network for industrial surface defect detection. The key contributions include the Edge Information Extraction Module (EIEM) for optimized down-sampling and feature extraction, multi-scale down sampling (MSD) using Haar wavelets, and a Lightweight Feature Enhancement Module (LFEM).	The model's accuracy is dataset-dependent, and it has not been validated on localized industrial data, limiting generalizability. The model does not employ transfer learning or active learning, which could improve performance under limited labeled data.
(Yang., 2025a) proposed a lightweight CNN model for detecting surface defects in strip steel, which was optimized for resource-constrained embedded devices. The model integrated a parameter-free attention mechanism (SimAM), an inverse residual structure, and a depth-wise convolution to create a lightweight attention mobile (LAM) backbone network.	The model significantly reduces parameters and computational cost; however, it still relies on labeled datasets, which may be challenging for small-scale or developing industries with limited annotated data.
(Zhang et al., 2025) proposed a lightweight framework that improves multiscale feature extraction and interaction. The framework introduces Group Multiscale Bidirectional Interactive (GMBI) modules; the model uses a group-wise strategy for consistent computational cost across scales.	The model has not been validated with localized steel industry data, which may differ in defect types and image characteristics. The framework does not incorporate transfer learning or active learning, which could further improve performance with limited labeled data.
(Zeng et al., 2019) explored an image-based processing method for steel sheet defect detection, aiming to replace laborious manual inspection. Data augmentation techniques are used to expand the limited defect dataset for further training. Transfer learning is applied to both classification and regression networks to save training time and reduce the cost of dataset preparation. A hierarchical model ensemble approach is used to achieve better defect detection performance.	The models Ensemble technique increases compute resources, but the model has limited defect coverage. There is a need for the adoption of active learning and class-imbalance handling using lightweight CNNs.

2.2 SUMMARY OF LITERATURE REVIEW FINDINGS

The literature review revealed that CNN-based approaches have achieved high accuracy in metal surface defect detection. However, most existing studies focus on resource-rich settings with access to large annotated datasets, high-end GPUs and controlled imaging conditions. Heavy architectures like the ResNet50, NASNet and other multilevel feature fusion networks perform well in these contexts, but are unsuitable for resource-constrained industrial environments. The majority of studies rely on exhaustive pixel-level annotations, which limit their scalability and adaptability to real-world conditions.

Prior work often overlooks challenges that are found in low-resource settings, which include limited computational power, small and noisy datasets, heterogeneous surface finishes, variable lighting, and the absence of robust edge-deployable solutions. Few studies have managed to integrate adaptive mechanisms, such as active learning, which assist in the process of continuously improving the model performance with minimal annotation effort. The literature reveals a clear gap in the systematic integration of lightweight CNN architectures, transfer learning and active learning for models explicitly localised to developing-country industrial environments.

The reviewed literature highlights a clear gap in the development of context specific metal surface defect detection models suited for developing country industrial environments. Although high-performance CNN architectures dominate in existing research, their reliance on extensive labeled datasets, powerful computing infrastructure and controlled operating conditions limits their applicability in settings such as the Zimbabwean steel industry. There remains a clear need for a lightweight, data-efficient model that can be customized using localized data and be incrementally improved to reflect real-world industrial conditions.

This research addresses these gaps by proposing a customized lightweight CNN (L-CNN) framework that is specifically designed for resource-constrained environments. The model leverages transfer learning as a way to reduce dependency on large datasets, and at the same time incorporates active learning as a way to minimize annotation costs and enable continuous adaptation to new defect types. The model is optimized for edge deployment on low-power hardware. The framework is designed to be robust against variable industrial conditions, such as heterogeneous surfaces and fluctuating lighting, which are common in real-world manufacturing settings. The combination of these strategies makes the proposed study offer a novel, practical solution that overcomes the limitations of previous work and enables efficient, high-accuracy metal surface defect detection in environments with constrained resources.

2.3 LIMITATIONS OF THE LITERATURE REVIEW

The literature review highlights the successes of deep learning CNNs in metal defect detection. CNNs have the capacity to improve drastically on accuracy, efficiency, and real-time performance in various metal surface defect detection systems. However, a research gap exists in the context of resource-constrained environments like the Zimbabwean steel manufacturing industry, as it is lagging in terms of the automation of the metal surface defect detection process using machine learning. There is a need to implement an automated surface metal defect detection system that uses deep learning CNNs. The current CNN models lack the ability to prioritize the detection of specific metal surface defect types prevalent and critical in the Zimbabwean Steel Industry. The proposed solution of implementing a hybrid approach of combining both Machine learning and Active learning can help overcome the dataset challenge, reduce the data annotation burden, and accelerate the model development process, making it more efficient and cost-effective, improving the quality control process of steel in Zimbabwe.

This literature review has examined the state of research on the application of CNN-based approaches for automated metal surface defect detection, with a particular focus on the unique challenges and requirements of developing economy steel industries, such as the one in Zimbabwe. The review has identified the key advantages of CNN-based solutions, the characteristics and constraints of steel manufacturing in developing contexts, and the limited existing research on the adaptation and deployment of these technologies in resource-constrained environments.

To address the gaps in the literature, future research should aim to develop a deeper understanding of the steel industry in developing economies and to create customized CNN-based defect detection models that are tailored to the specific needs and constraints of these contexts. Through leveraging the power of CNN-based approaches and addressing the unique requirements of developing economies steel industries, researchers and practitioners can contribute to the improvement of product quality, process efficiency, and overall competitiveness in the global steel market.

3. RESEARCH APPROACH

Research Design Approach and experimental design methods were used together to develop and evaluate a CNN-based model for detecting metal surface defects. The Research Design Approach outlines a structured plan for conducting the study, ensuring that all aspects of the research align with the objectives and research questions (Ren et al., 2018). The Experimental Design method was used to come up with controlled experiments to test specific aspects of the CNN model (Cumbajin et al., 2023). The researchers managed to manipulate variables and observe their effects on model performance.

3.1 SYSTEM DEVELOPMENT APPROACH

The researchers identified a suitable dataset of steel surface images and defects that is similar to the defects found in the Zimbabwean Steel Industry. The dataset was created by (Islam, 2018) and the dataset was downloaded from https://www.kaggle.com/datasets/fantacher/neu-metal-surface-defects-data, NEU Metal Surface Defects Database which contains six kinds of typical surface defects of the hot-rolled steel strip are collected, rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc). The database includes 1,800 grayscale images. The researchers used computational resources, such as a GPU-enabled machine learning workstation platform, to support the model training and inference.

3.2 PREPROCESSING

The first step was to pre-process the dataset to guarantee compatibility with the Zimbabwean Steel Industry surface images. This adaptation does not assume that steel produced in Zimbabwe is materially different; rather it addresses the variations in imaging conditions, surface finishing and data availability typical of resource-constrained industrial environments. This process encompasses the resizing of images, standardizing pixel values, and implementing data augmentation techniques to diversify the available training data using the Image Data Generator class from the Keras library. The pixel values are rescaled to the range [0, 1] through division by 255. This is a common pre-processing step to standardize the pixel values. The images are randomly rotated up to 20 degrees, which makes the model learn to recognize objects at different orientations. The images are randomly shifted horizontally and vertically by up to 10% of the total width and height, respectively. This was done to make the model learn to recognize objects in different positions within the image. The images are randomly flipped horizontally to help the model learn to recognize objects that may appear in different orientations. Preprocessing is a perfect technique for visualizing the sample of the images in the dataset along with their corresponding labels. The method is employed to examine the data and verify that the data Preprocessing and augmentation processes are functioning correctly. The incorporation of locally collected Zimbabwean steel images allows the model to intentionally adapt and be more biased toward local operational conditions, a process further reinforced through active learning in order to improve performance under limited data and computational resources. The pre-trained model's weights and feature extraction layers serves as the initial stage for training. The model would be fine-tuned later on using the Zimbabwean metal steel surface images.

3.3 IMPLEMENTATION OF THE MODEL

The model is a lightweight Convolutional Neural Network (CNN) model for an image classification task that comprises six (6) classes. The first convolutional layer consists of 32 filters, each with a 3x3 kernel size. The activation function that was used is the ReLU (Rectified Linear Unit). The input shape is set to (200, 200, 3), which means the model expects 200x200 pixel images with 3 colour channels (RGB). The max pooling layer is a 2x2 pool size. This is important to reduce the spatial dimensions of the feature maps by a factor of 2. The second convolutional layer has 32 filters also and a 3x3 kernel size, followed by ReLU activation. The second max pooling layer has a 2x2 pool size. The layer flattens the 2D feature maps into a 1D vector, preparing the data for the fully connected layers. The first fully connected (dense) layer has 128 units and ReLU activation. The second fully connected layer has 32 units and uses the ReLU activation. The model has a dropout layer with a rate of 0.5 to prevent overfitting from occurring. The final output layer has 6 units (corresponding to the six (6) classes) and a softmax activation function, which outputs the class probabilities.

The Adam optimizer was utilized. It is a widely used optimization algorithm for training neural networks. The Adam optimizer incorporates an adaptive learning rate optimization technique that merges the advantages of RMSProp and Momentum. The categorical cross-entropy loss function has been activated. The specified loss function was used in the multi-class classification assignments, in which the model must forecast the likelihood of each class. The categorical cross-entropy loss function assesses the effectiveness of a classification model that generates a probability output within the range of 0 to 1. The metrics parameters were configured for accuracy. During the training and evaluation phases of the model, the classification accuracy was reported as a performance metric, along with the loss value.

When compared to larger CNN models, the lightweight CNN model has fewer filters and neurons. The main goal is to reduce the number of parameters in the model and increase efficiency by reducing the number of filters in the convolutional layers and the number of neurons in the fully connected layers. The feature maps spatial dimensions are reduced with the help of the max pooling layers. A dropout layer with a generally high rate of 0.5 is executed to both forestall overfitting and to regularize the model.

The Lightweight CNN model's smaller size comes with improved efficiency. The lightweight CNN model is ideal for applications requiring limited computational resources. However, it may have slightly lower performance compared to larger and more complex CNN architectures.

3.4 TRANSFER LEARNING

Transfer Learning is a machine learning technique that includes using a model trained on one task to perform a different but related task. In deep learning, this encompasses using a pre-trained model, for instance, a convolutional neural network trained on a large dataset, which was a starting point for a new model that was trained on a smaller but domain-specific dataset.

The noble idea behind Transfer Learning is that all the low-level features of the image edges and shapes learned by the pre-trained model can be useful for the new task. The technique can still work efficiently even if the target dataset and problem domain are different. The use of the pre-trained model's knowledge makes the system achieve better performance with less training data and computational resources, compared to going through the whole process of training a model.

3.5 INTEGRATING TRANSFER LEARNING WITH L-CNN

The following steps were covered by transfer learning. Firstly, the pre-trained Model Selection was done, and the MobileNetVer2 was selected as the pre-trained model. The MobileNetV2 is very efficient in terms of computational resources and memory usage, making it ideal for mobile and embedded applications and for developing countries like Zimbabwe that have challenges in computational resources. The architecture achieves faster inference times and smaller model sizes at the same time, maintaining high accuracy. The design is made up of depth-wise separable convolutions and inverted residuals, which guarantee versatility and effective performance when incorporating and using transfer learning.

Feature Extraction was done by using the pre-trained model as a feature extractor by removing the final classification layer. Data was inputted through the pre-trained model to obtain the feature representations. Fine-tuning was done by adding a new classification layer to the pre-trained model. Freezing the weights of the pre-trained layers, except for the final layers, was done. Training of the new layers on the task-specific dataset was done to allow the weights to be updated during the training process. The next step is hyperparameter tuning, which was implemented by experimenting with different hyperparameters, namely the learning rate, batch size, and regularization techniques, to optimize the performance of the fine-tuned model. Evaluation was done to efficiently evaluate the performance of the fine-tuned model on a held-out test set and through cross-validation. Comparison of the performance of the fine-tuned model to the original dataset and model was done.

The combination of transfer Learning with the L-CNN model and customizing the Zimbabwean steel image dataset will make the benefits of both techniques to improve the performance of the image classification task in as it can achieve better performance with a smaller dataset compared to training a model from scratch. The technique highly reduces the time and computational resources required for training, as the pre-trained model has already learned useful features. On top of that, it is beneficial when the target dataset is small, and beginning to train a model from the start to the end would result in overfitting. The use and employment of the knowledge learned by the pre-trained model, transfer learning, can be a powerful technique to improve the performance of the machine learning model, especially in this case when your dataset is limited.

3.6 ACTIVE LEARNING INTEGRATION

Active learning is an effective machine learning technique that improves the performance of a model by actively selecting the most informative samples for annotation and inclusion in the training dataset. Active learning involves the human expert or annotator in the learning process, where the model identifies the most uncertain or influential samples and requests the human to provide the correct labels for those samples.

Active learning was implemented by initial model training of the CNN model using the available labelled dataset. The next stage will activate the Sample Selection, where active learning techniques were applied to identify the most informative samples from the unlabelled data. There are various active learning techniques, such as uncertainty sampling, that were introduced, where the model selects the samples it is most uncertain about; this encompasses samples with low confidence scores or high entropy in the predicted class probabilities.

In this study, instead of using the Query by Committee technique, which involves training multiple versions of the model and selecting samples with the highest disagreement among committee members, the researchers used the Monte Carlo dropout evaluation method. This technique encompasses performing multiple forward passes through the network with dropout enabled. The method generates a distribution of predictions for each sample. The images that exhibited the highest uncertainty or poorest results were selected. These uncertain images were labelled and added to the training set. This was done to greatly improve the model's performance by addressing areas of weakness at the same time, enhancing its overall efficiency. After updating the training dataset with the newly annotated samples, the CNN model is retrained using the expanded dataset. The iterative process of active sample selection, annotation, and model retraining was repeated until the model's performance reached a satisfactory level or the available annotation budget was exhausted.

The integration of active learning with the CNN model iteratively improves the model's performance by allowing it to focus on the most informative samples, reducing the need for a large, fully labelled dataset. This can be particularly helpful when the dataset is limited or when the cost of data annotation is high.

3.7 ACTIVE LEARNING IMPLEMENTATION

The fine-tuned model was used as the initial model for the active learning process. The active learning cycle would involve the model making predictions on a small set of Zimbabwean metal surface images. The model's confidence scores were analysed to identify the most informative or challenging samples. The selected samples were prioritized for manual annotation by domain experts. The next stage involves adding the annotated data to the training dataset, and the model was retrained. The iterative process will continue, and at each stage, until the model gradually becomes more specialized and accurate for the Zimbabwean Steel Industry.

3.8 CNN MODEL VISUALIZATION SHOWING THE USE OF TRANSFER AND ACTIVE LEARNING

The diagram below shows the CNN Model Visualization showing the use of transfer and active learning that are incorporated into the light-weight CNN model

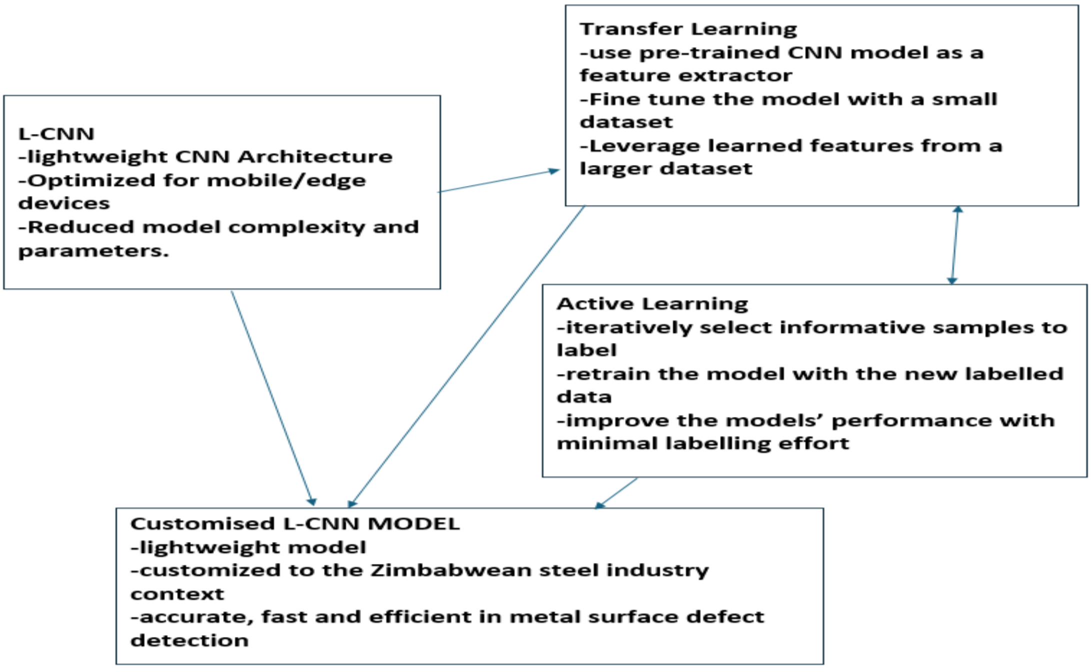

Figure 1. CNN Model Visualization showing the use of transfer and active learning.

4. EXPERIMENTAL SETUP

The experimental setup of this study involved the design and customization of a lightweight CNN model, leveraging transfer learning and active learning techniques to optimize it for the Zimbabwean steel industry's specific needs.

A comprehensive understanding of the results is crucial to review and understand the experimental setup used for this study. Initially, the researchers developed a lightweight CNN model and fine-tuned it to minimize computational resource requirements. The approach was taken to accommodate the Zimbabwean context, where limited computational resources can pose challenges to implementing and using the technology.

After verifying the model's effectiveness, the next step involved incorporating transfer learning. This technique leverages the existing knowledge from pre-trained models to enhance performance, efficiency, and resource utilization in new tasks. Transfer learning reduced training time, improved accuracy, enabled the effective use of limited data, leading to the development of robust models. The MobileNetV2 pre-trained model was employed for this purpose.

Lastly, hundred (200) images taken from the Zimbabwean steel industry were tested on the lightweight CNN model. Images with low probabilities across all classes were selected. These images indicated the model's uncertainty about its predictions. These images were correctly re-labelled and added to the dataset. Active learning was employed to customize the images to better fit the conditions of the Zimbabwean Steel Industry. Tests were conducted, and the results were recorded and analysed.

4.1 Diagram showing the stages of the Experimental Setup

The diagram below Figure 2, shows the stages of the experimental setup.

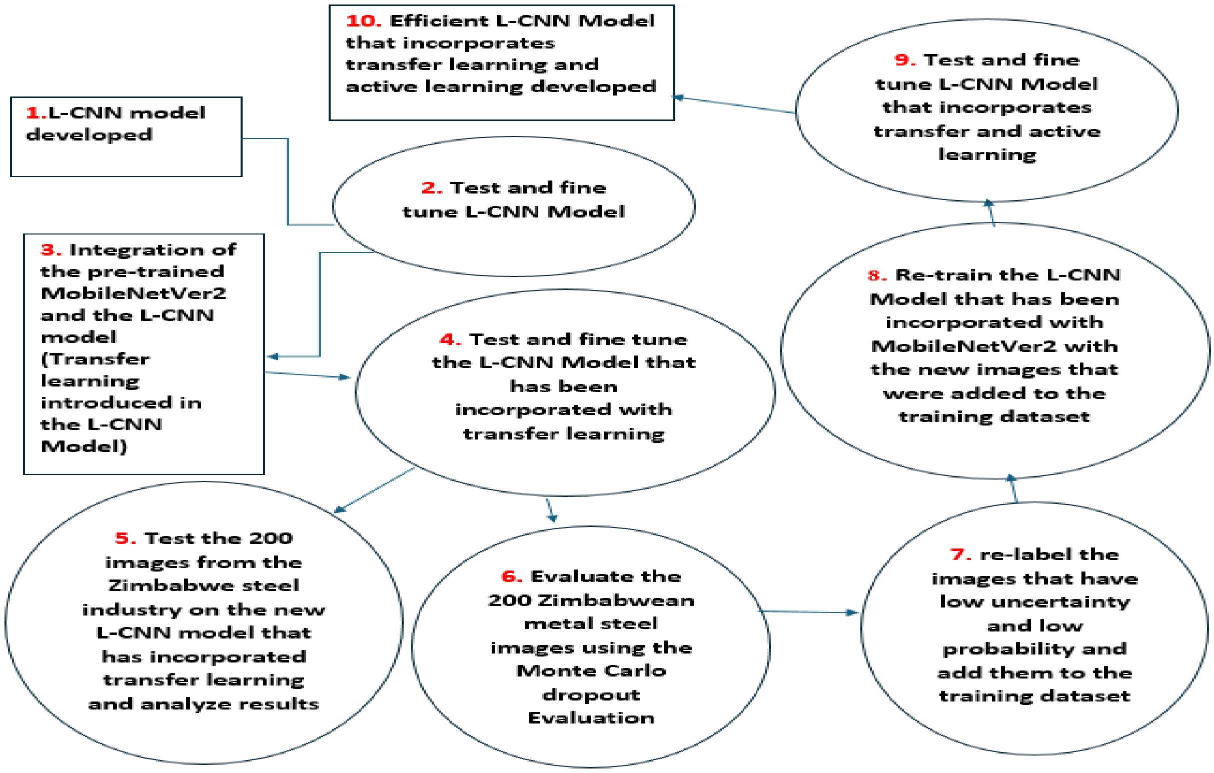

Figure 2. Experimental Setup Stages.

4.2 Sample Output images of the Experimental Setup

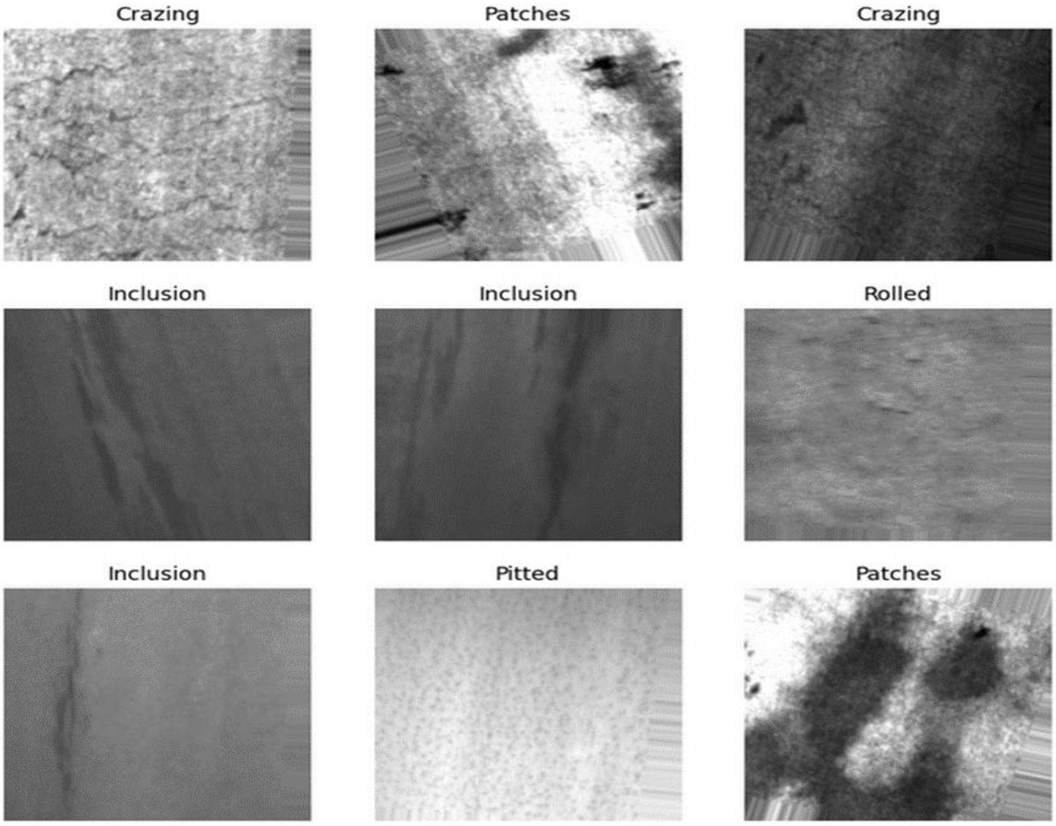

Figure 3. shows results of metal surface defects that were outputted by the model Light weight CNN Model.

4.3 Model Code

```
# Model layers (Modified)
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(200, 200, 3)),  # Reduced filters to 32
    MaxPooling2D((2, 2)),
    Conv2D(32, (3, 3), activation='relu'),  # Reduced filters to 32
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),  # Reduced neurons to 128
    Dense(32, activation='relu'),   # Reduced neurons to 32
    Dropout(0.5),                   # Increased dropout rate
    Dense(6, activation='softmax')  # 6 classes
])
```

Figure 4 shows the code of the light-weight CNN model.

4.4 Model Summary

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 198, 198, 32)      896

 max_pooling2d_4 (MaxPoolin  (None, 99, 99, 32)        0
 g2D)

 conv2d_5 (Conv2D)           (None, 97, 97, 32)        9248

 max_pooling2d_5 (MaxPoolin  (None, 48, 48, 32)        0
 g2D)

 flatten_2 (Flatten)         (None, 73728)             0

 dense_6 (Dense)             (None, 128)               9437312

 dense_7 (Dense)             (None, 32)                4128

 dropout_2 (Dropout)         (None, 32)                0

 dense_8 (Dense)             (None, 6)                 198

=================================================================
Total params: 9451782 (36.06 MB)
Trainable params: 9451782 (36.06 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 5. shows the summary of the Light-weight CNN Model showing its structure and architecture.

4.5 Model Structure

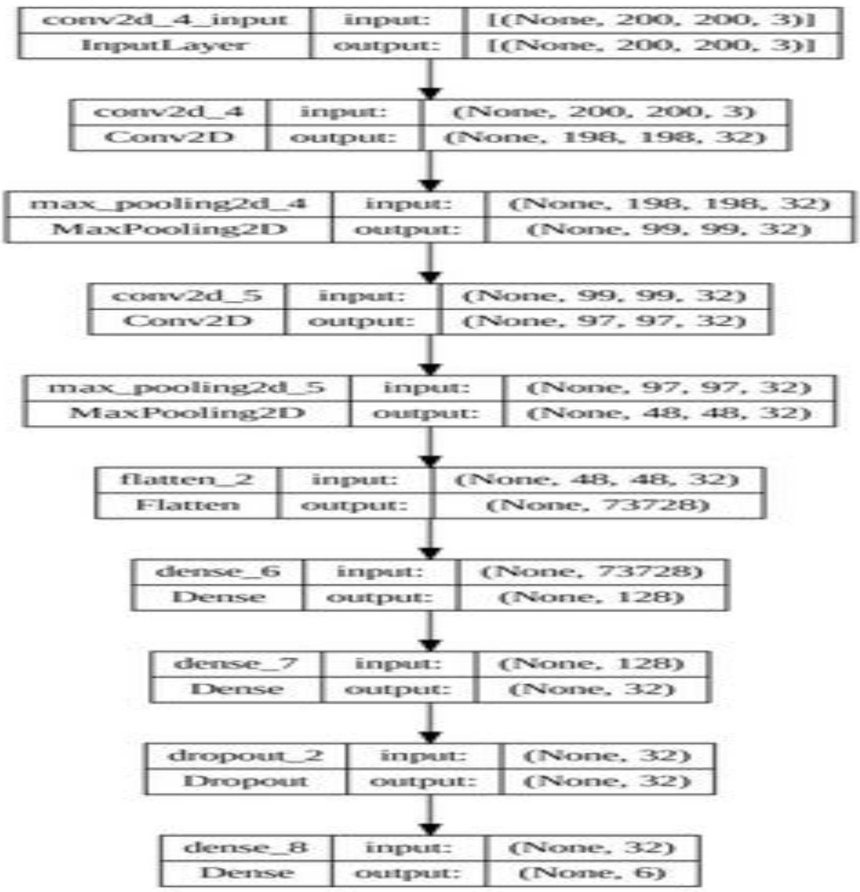

Figure 6 shows a pictorial diagram of the Light-weight CNN Model structure and architecture.

4.6 Model Accuracy Graph

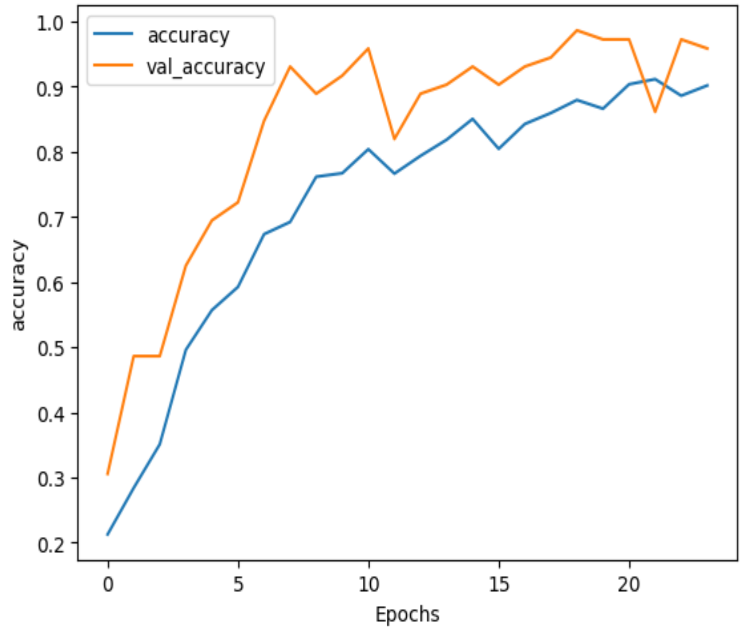

Figure 7 shows the Model accuracy graph of the Light-weight CNN Model that has 24 Epochs and a test accuracy of 97%.

4.7 Model Loss Graph

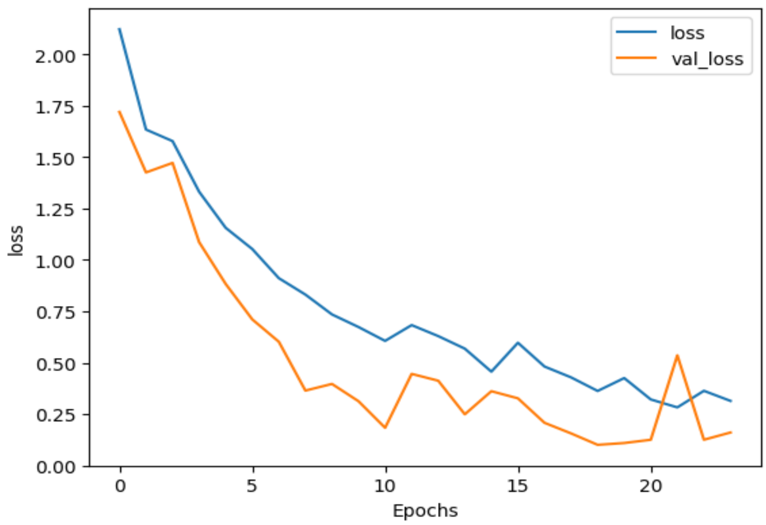

Figure 8 shows the Model Loss graph of the Light-weight CNN Model that has 24 Epochs and a test loss of 19%.

4.8 Model Summary that incorporates Transfer learning

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_224 (Func  (None, 7, 7, 1280)        2257984
 tional)

 flatten (Flatten)           (None, 62720)             0

 dense (Dense)               (None, 128)               8028288

 dense_1 (Dense)             (None, 32)                4128

 dropout (Dropout)           (None, 32)                0

 dense_2 (Dense)             (None, 6)                 198

=================================================================
Total params: 10290598 (39.26 MB)
Trainable params: 8032614 (30.64 MB)
Non-trainable params: 2257984 (8.61 MB)
```

Figure 9 above shows the summary of the Light-weight CNN Model that incorporates transfer learning, showing its structure and architecture.

4.9 Accuracy Graph that incorporates Transfer learning

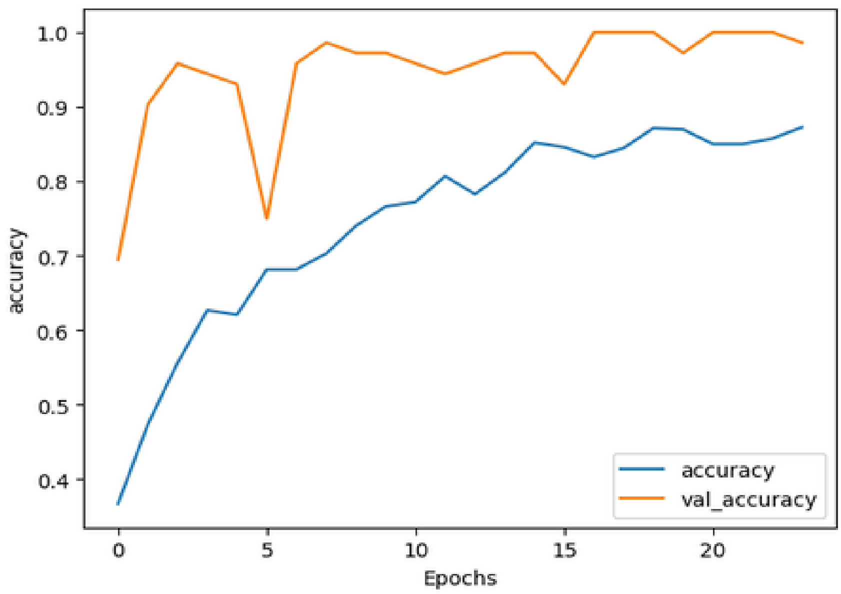

Figure 10 shows the Model accuracy graph of the Light-weight CNN Model that has 24 Epochs and a test accuracy of 98%.

4.10 Loss Graph that incorporates Transfer learning

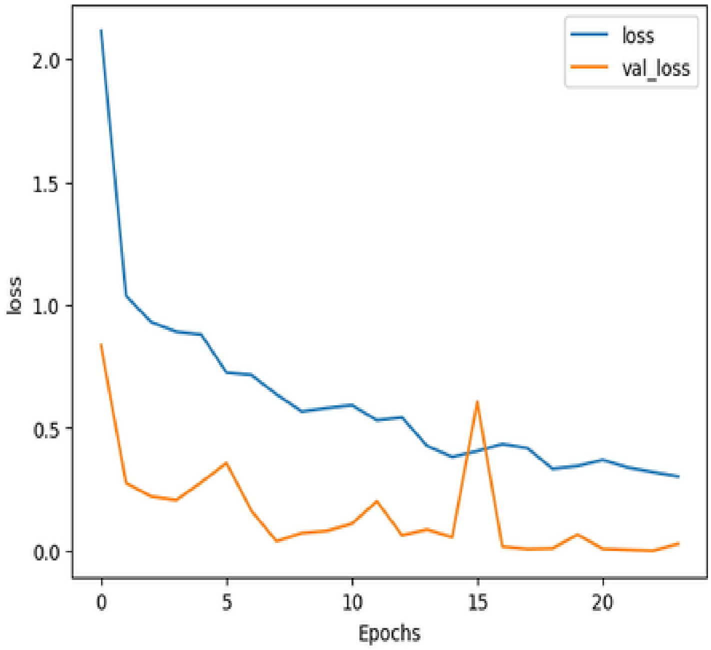

Figure 11 shows the Model Loss graph of the Light-weight CNN Model that has 24 Epochs and a test loss of 2%.

4.11 Active Learning Implementation

Active Learning is an effective technique to enhance model performance by purposefully selecting the most informative samples from unlabelled data and requesting labels for those samples from an oracle. The main goal is to achieve high performance with fewer labelled samples, and this is particularly useful when labelling data is expensive or time-consuming.

The researchers obtained 100 different types of images from the Zimbabwean Steel Industry and evaluated them using the Monte Carlo method. The active learning approach is effective because it employs the Monte Carlo dropout evaluation, which allows the model to quantify its own uncertainty. This is useful for identifying ambiguous and even samples that are difficult to classify, targeting them for further training and fine-tuning to improve the model's overall performance. Furthermore, the technique is essentially helpful in customizing the dataset and model to the specific conditions and requirements of the Zimbabwean context, so as to ensure that the model is better suited to local industry needs. The images were placed in a folder titled "mypictures," uploaded to Google Drive, and labelled as picture1.bmp to picture200.bmp, respectively

4.12 Explanation of Active Learning

Active learning aims to identify the most valuable instances in a dataset for improving a model through labelling or re-labelling. Based on the results provided, a systematic approach was used to determine which images might need to be re-labelled. The indicators for re-Labelling firstly are low confidence predictions. The method involves identifying images where the model's confidence in its predictions is below a pre-defined threshold. However, most of the predictions had a high confidence. The step whereby any low-confidence predictions are flagged for potential re-labelling. High uncertainty can also be looked at where by prediction uncertainty values are analysed. Both high uncertainty and low probabilities for the predicted class for all classes indicate that the model is unsure about its prediction. When this occurs, there is a need for re-labelling or further investigation of the images.

Another technique is to identify instances where the probabilities for the predicted class are close to the certainty threshold, that is, around 0.5. While probabilities are generally high in the provided results, the step identifies cases where the model's confidence is marginal, indicating potential struggle. The application of the indicators of the model can be continuously improved by focusing on the most informative and challenging instances, therefore improving the overall performance and reliability.

4.13 Analysis of results of Sample images selected after Active Learning

The table 2, below shows summarised results after Active Learning; a few images were selected.

Table 2 summarised results of the Images after Active Learning

Picture Name, Predicted Class, and Prediction Probability	Prediction Uncertainty	Reason for Re-Labelling
picture4.bmp Class 2 (0.9369) High	Higher compared to others	Slightly higher uncertainty for the predicted class.
picture7.bmp Class 2 (0.9998) High	Slightly higher	Higher uncertainty for the predicted class.
picture9.bmp Class 2 (0.9680) High	Low but significant	Potential ambiguity despite high confidence.

4.14 Summary of sample images selected for re-labelling

The table 3 below shows summarised results of some of the images that were selected for re-labelling after the Monte Carlo Dropout Evaluation.

Table 3 Summarises the results of some selected images after the Monte Carlo Dropout Evaluation

Prediction	Prediction Uncertainty	Reason
Picture 4 [3.2090859e-09, 3.2827008e-09, 9.3690240e-01, 9.3354993e-06, 7.2657194e-08, 6.3088179e-02]	[2.2204460e-16, 2.2204460e-16, 0.0000000e+00, 9.0949470e-13, 7.1054274e-15, 0.0000000e+00	The model is confident about its prediction (high probability for class 2), but the uncertainty for the predicted class is higher compared to other images. This might suggest that while the prediction is accurate, it could benefit from additional review or validation
Picture 7 [3.0901397e-09, 1.1797434e-09, 9.9978381e-01, 1.0894874e-05, 9.2806942e-05, 1.1246854e-04]	[0.000000e+00,0.000000e+00, 0.000000e+00, 0.000000e+00, 7.275958e-12, 0.000000e+00]	Similar to picture 4, this image has high confidence, but the uncertainty for the predicted class is slightly higher than in other cases
Picture 9 [9.4034442e-09, 6.1839858e-09, 9.6801889e-01, 1.4414158e-04, 5.3136813e-07, 3.1836364e-02]	[9.4034442e-09, 6.1839858e-09, 9.6801889e-01, 1.4414158e-04, 5.3136813e-07, 3.1836364e-02]	The predicted class has a high probability, but the uncertainty for other classes, though low, suggests potential ambiguity.

4.15 Code Snippet showing one of the images (picture4.bmp) that was chosen for re-labelling

```python
# Set the path for the image you want to classify
image_path = '/content/drive/MyDrive/mypictures/picture4.bmp'

# Load and preprocess the image
img = load_img(image_path, target_size=(200, 200))
img_array = img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)

# Normalize the image
img_array = img_array / 255.0

# Make the prediction
prediction = model.predict(img_array)
```

Figure 12 shows the code of inputting picture4.bmp in the L-CNN Model to test what the type of defect it has.

4.16 System output of (picture4.bmp) that was chosen for re-labelling

```
1/1 [==============================] - 1s 1s/step
The image is classified as: inclusive
```

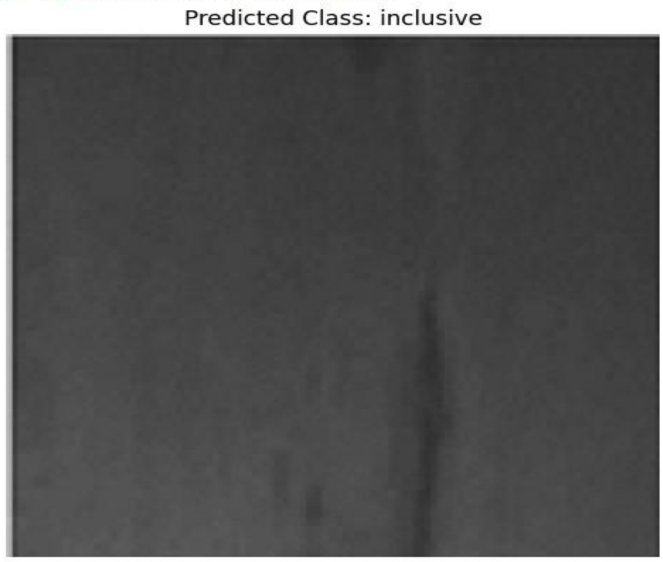

Figure 13 shows that the picture has scratches and is inclusive, but is not clear.

4.17 Actionable steps to implement Active Learning

The 200 pictures were re-sized to a width of 200 and a length of 200 pixels, they were originally had a width of 188 and a height of 202 pixels, so they had to be re-sized to match the size of the pictures in the original training dataset. They were placed in the folders that belonged to their classes.

The new dataset, which had the new pictures, was uploaded on Google Drive, and the model was re-trained using the updated dataset.

4.18 Code Snippet showing the new dataset being uploaded

```
#import os

# Assuming 'train' is a directory containing images
train_dir = '/content/drive/MyDrive/customizeddataset/train'

# List all files and directories within the 'train' directory
contents = os.listdir(train_dir)

# Print the contents to inspect them
print(contents)

test_dir ='/content/drive/MyDrive/customiseddataset/test'
valid_dir='/content/drive/MyDrive/customiseddataset/valid'
```

Figure 14 shows the updated dataset being uploaded into the model

4.19 Code Snippet showing the new CNN Model being saved

```
from keras.models import load_model

# Load the saved model
model.save('/content/drive/MyDrive/finalzimsteellcnn.keras')
```

```
from keras.models import load_model

# Load the saved model
model.save('/content/drive/MyDrive/finalzimsteellcnn.h5')
```

Figure 15 shows the new model being saved, the model has transfer learning and active learning incorporated into it, and has also been trained on the new updated dataset.

4.20 Graph Showing test accuracy

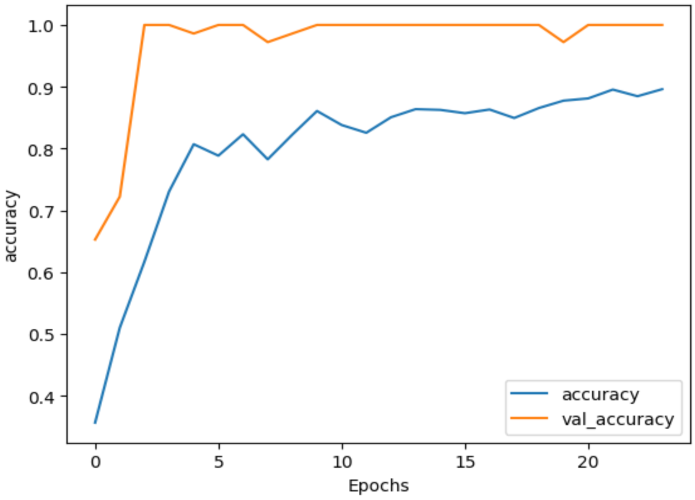

Figure 16 shows the Model accuracy graph of the Light-weight CNN Model that incorporates both transfer and active learning that has 24 Epochs and a test accuracy of 99.4%.

4.21 Graph Showing test loss

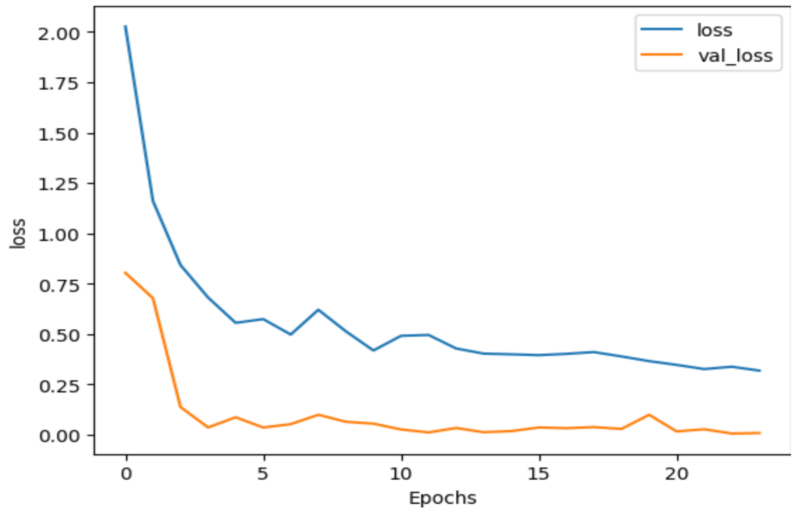

Figure 17 shows test loss results of the L-CNN model that is incorporated with both transfer and active learning, which has a loss of 0.0042.

4.22 COMPARISON OF THE THREE MODELS

Table 4, below, shows a comparison of three models for testing accuracy and test loss.

Table 4 shows a summary of the test accuracy and test loss of the L-CNN model, L-CNN integrated with MobileNetV2 (Transfer Learning) and the L-CNN Model integrated with Transfer learning and Active Learning.

Name of Model	Test Accuracy	Test Loss
L-CNN	97.22%	0.1999
L-CNN integrated with MobileNetV2 (Transfer Learning)	98.61%	0.02078
L-CNN Model (Transfer learning and Active Learning incorporated)	99.4%	0.0042

5. RESULTS AND ANALYSIS

The results in Table 4 clearly show a continuing improvement in both test accuracy and test loss as we move from the basic L-CNN model to the more advanced models integrating transfer learning and active learning techniques.

The initial L-CNN model achieved a test accuracy of 97.22% with a test loss of 0.1999. The metrics reveal a high level of performance however, room for improvement is there, especially in reducing the test loss, which is important as it is a benchmark that measures the model's prediction error on unseen data.

The incorporation of the pre-trained MobileNetV2 through Transfer Learning extensively enhanced the model's performance, and this resulted in a test accuracy of 98.61% and a substantially reduced test loss of 0.02078. The improvement highlights the efficacy of using and leveraging pre-trained models to transfer learned features therefore enhancing the L-CNN's ability to generalize better to when encountering new data.

The integration of both Transfer Learning and Active Learning techniques brought about a very notable improvement, which made the model achieve a perfect test accuracy of 99.4% and an impressively low-test loss of 0.0042. The remarkable performance showcases the advantages of using an updated dataset together with active learning strategies, as it allows the model to learn from the most informative data points, making it possible for the model to refine its predictive capabilities to the optimal level.

The gradual enhancement observed across the models demonstrates the profound impact of integrating advanced transfer and active learning techniques and leveraging the strengths of pre-trained models. The transition from the L-CNN to its integration through transfer learning with MobileNetV2 pre-trained architecture, and further pre-training it with an updated dataset that incorporated active learning, greatly improved test accuracy. All of this demonstrated the prospective power of combining Transfer Learning and Active Learning for achieving a superior model performance.

In summary, the empirical results validate the hypothesis that model performance can be significantly enhanced through the strategic integration of Transfer Learning and Active Learning techniques. The final model that achieved perfect accuracy and a minimal loss sets a new benchmark for future research and application in this domain. Moreover, the results reflect that the L-CNN model is well-suited for contexts in developing countries like Zimbabwe, where challenges with computational resources and dataset availability are prevalent. The results clearly support the idea of developing a customized model for Zimbabwe, as the labelled images obtained through Active Learning make this customization feasible and effective.

5.1 SUMMARY OF FINDINGS

The research successfully developed a succession of lightweight CNN models custom-made to the specific needs of the Zimbabwean Steel Industry. The initial L-CNN model demonstrated a solid baseline performance with a test accuracy of 97.22% and a test loss of 0.1999. However, the incorporation of Transfer Learning significantly enhanced the model’s performance. The introduction of the pre-trained MobileNetV2 through Transfer Learning led to an outstanding increase in test accuracy to 98.61% and a reduction in test loss to 0.02078. Further refinement through combining Transfer Learning and Active Learning culminated in achieving a perfect test accuracy of 99.4% and an exceptionally low-test loss of 0.0042.

The study offers significant insights into the application of Transfer Learning and Active Learning for metal surface defect detection. The research demonstrated how leveraging pre-trained models using transfer learning and continuously updated datasets through active learning can considerably enhance performance metrics. The study advances the theoretical understanding of deploying advanced machine learning techniques in specialized industrial applications. This research bridges the gap between general deep learning methodologies and their specific adaptation for industrial defect detection, providing a framework that can be generalized to similar contexts.

Practically, this research provides a solution by automating of metal surface defect detection process. This addresses the challenges of manual visual inspections that are done in the Zimbabwean Steel Industry, which are labour-intensive and error-prone. The developed model not only improves accuracy but also reduces operational costs, offering a practical tool that meets the industry’s requirements. The incorporation of Transfer Learning and Active Learning enables the model to continuously adapt to real-world data, therefore ensuring sustained performance improvements and operational efficiency in resource-constrained environments.

The research introduces a novel framework that integrates a lightweight CNN model with transfer learning and active learning to address metal surface defect detection in resource-constrained environments. Whilst individual elements of these techniques have been explored, this framework uniquely combines them to overcome challenges in computational power and data labelling. The model's ability to iteratively update with real-world data ensures continuous improvement and adaptation, therefore enhancing performance and feasibility in environments with limited resources. This contribution is significant for deploying advanced AI technologies in developing countries, improving the accessibility and effectiveness of automated inspection systems, and adding valuable insights to the field of computer science by demonstrating a cost-effective, optimized approach to metal surface defect detection.

5.2 SYSTEM ARCHITECTURE DIAGRAM SHOWING COMPONENTS OF THE FRAMEWORK

The diagram below shows the system architecture diagram showing components of the framework.

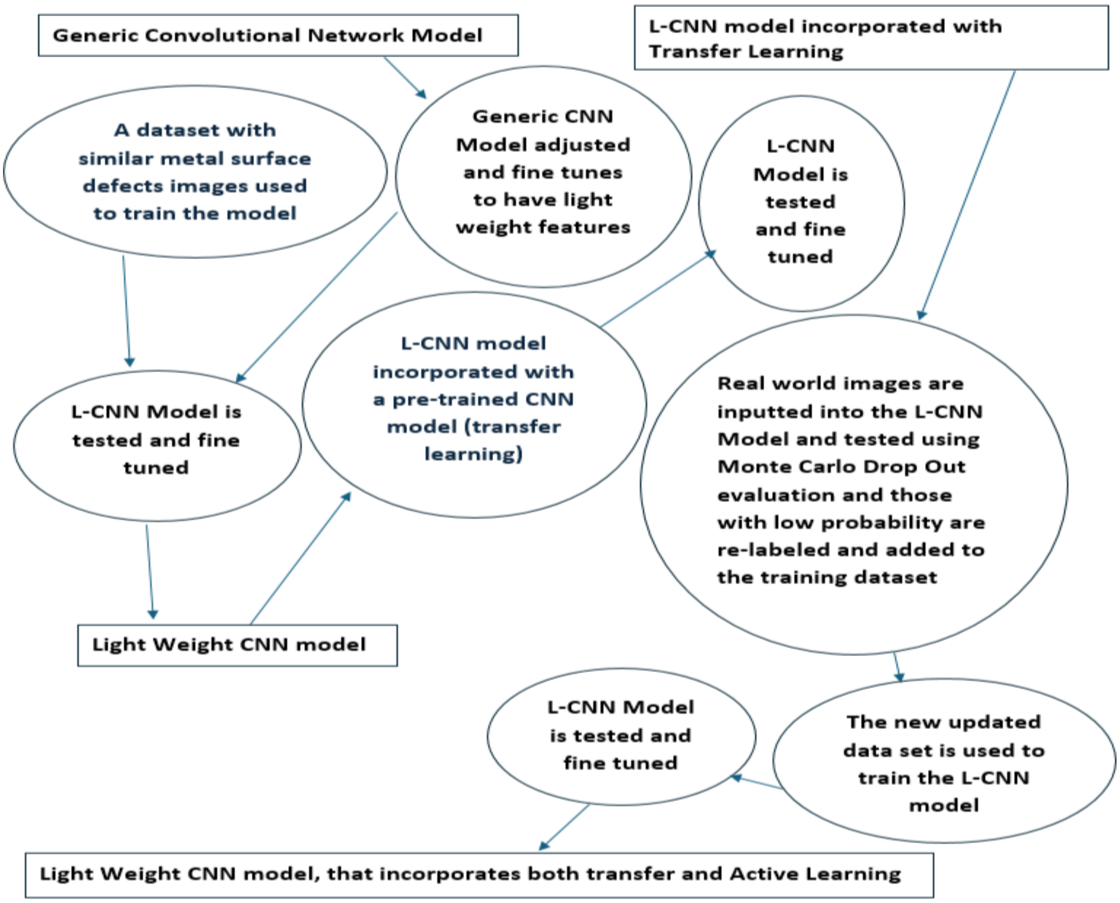

Figure 18 System architecture diagram showing the components of the Framework

5.3 LIMITATIONS

While the research achieved high performance metrics, there are inherent limitations in the model's design and implementation. The study relied on the MobileNetV2 architecture and the availability of labelled images through Active Learning, which may not be universally applicable in all contexts or industries. The theoretical assumptions supporting the use of Transfer Learning and Active Learning may not fully account for all variables affecting metal surface defect detection, such as variations in defect types or production processes.

6. CONCLUSIONS AND FUTURE WORK

In conclusion, the research demonstrated the significant impact of integrating advanced machine learning techniques on the performance of metal surface defect detection models. The successful development and validation of a model achieving near-perfect accuracy and minimal loss set a new benchmark in this domain. The results underline the potential of customized models in addressing specific industrial needs, particularly in resource-constrained environments like Zimbabwe.

The study validated the hypothesis that the strategic incorporation of Transfer Learning and Active Learning can extensively enhance model performance. This approach not only contributes to the field of machine learning but also offers practical solutions for industrial applications. The findings advocate for continued exploration and development of specialized models to address both current and future challenges in defect detection and beyond.

Future research could explore the application of alternative pre-trained models and architectures on metal surface defect detection and extend the study to include a broader range of defect types and industrial contexts, all of which could provide additional insights and improvements.

The model was tested primarily on data from the Zimbabwean Steel Industry. Therefore, its generalizability to other industries or regions with different types of metal surface defects may be limited. The research focused on specific types of defects prioritized by local needs, thus potentially excluding other relevant defect types.

To adequately address the limitations of this study, future work could focus on incorporating more diverse datasets and defect types to enhance model robustness. Furthermore, exploring methods to optimize computational efficiency without compromising accuracy would be beneficial, especially for deployment in settings with limited resources.

The integration of advanced Deep Learning techniques, such as reinforcement learning and unsupervised learning, could offer new opportunities for improving metal surface defect detection models. Investigating the potential of these emerging technologies and combining them with Transfer Learning and Active Learning may further advance the field.

7. REFERENCES

Aboulhosn, Z., Musamih, A., Salah, K., Omar, M., & Aung, Z. (2023). Date of publication 2023, date of current version 2023, Detection of Manufacturing Defects in Steel Using Deep Learning with Explainable Artificial Intelligence. *Digital Object Identifier 10.1109/ACCESS.2023.1120000*. https://doi.org/10.1109/ACCESS.2023.1120000

Amin, D., & Akhter, S. (2020). Deep Learning-Based Defect Detection System in Steel Sheet Surfaces. *2020 IEEE Region 10 Symposium (TENSYMP)*, 444–448. https://doi.org/10.1109/TENSYMP50017.2020.9230863

Chen, C., Chen, M., Lee, H., Li, Y., & Yu, J. (2025). *A Steel Surface Defect Detection Method Based on Lightweight Convolution Optimization*. https://doi.org/10.14569/IJACSA.2025.0160619

Cumbajin, E., Rodrigues, N., Costa, P., Miragaia, R., Frazão, L., Costa, N., Fernández-Caballero, A., Carneiro, J., Buruberri, L. H., & Pereira, A. (2023). A Systematic Review on Deep Learning with CNNs Applied to Surface Defect Detection. In *Journal of Imaging* (Vol. 9, Issue 10). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/jimaging9100193

Fu, G., Sun, P., Zhu, W., Yang, J., Cao, Y., Yang, M. Y., & Cao, Y. (2019). A deep-learning-based approach for fast and robust steel surface defects classification. *Optics and Lasers in Engineering*, *121*, 397–405. https://doi.org/10.1016/j.optlaseng.2019.05.005

Ghobakhloo, M., Iranmanesh, M., Grybauskas, A., Vilkas, M., & Petraitė, M. (2021). Industry 4.0, innovation, and sustainable development: A systematic review and a roadmap to sustainable innovation. *Business Strategy and the Environment*, *30*(8), 4237–4257. https://doi.org/10.1002/bse.2867

Guan, S., Lei, M., & Lu, H. (2020). A Steel Surface Defect Recognition Algorithm Based on Improved Deep Learning Network Model Using Feature Visualization and Quality Evaluation. *IEEE Access*, *8*, 49885–49895. https://doi.org/10.1109/ACCESS.2020.2979755

Gudukeya, L., Mbohwa, C., & Mativenga, P. T. (2019). Industrial sustainability in a challenged economy: the Zimbabwe steel industry. *Procedia Manufacturing*, *33*, 562–569. https://doi.org/10.1016/j.promfg.2019.04.070

He, Y., Song, K., Meng, Q., & Yan, Y. (2020). An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. *IEEE Transactions on Instrumentation and Measurement*, *69*(4), 1493–1504. https://doi.org/10.1109/TIM.2019.2915404

Hosseinzadeh, A., Shahin, M., Lopez, E. C., Chen, F. F., & Maghanaki, M. (2025). Performance evaluation of CNN models for steel surface defect detection in lean manufacturing. *Manufacturing Letters*, *44*, 1266–1273. https://doi.org/10.1016/j.mfglet.2025.06.146

Kateb, Y., Khebli, A., & Meglouli, H. (2024). Classification of Surface Defects in Steel Sheets Using Developed NasNet-Mobile CNN and Few Samples. *Revue d'Intelligence Artificielle*, *38*(2), 681–691. https://doi.org/10.18280/ria.380232

Kateb, Y., Meglouli, H., & Khebli, A. (2020). Steel Surface Defect Detection Using Convolutional Neural Network. *Algerian Journal of Signals and Systems*, *5*(4), 203–208. https://doi.org/10.51485/ajss.v5i4.122

Khebli, A., Kateb, Y., Meglouli, H., Aguib, S., Salah, M., & Touhami, K. (2024). Classifying Surface Fault in Steel Strips Using a Customized NasNet-Mobile CNN and Small Dataset. In *J. Electrical Systems* (Vol. 20, Issue 1). https://www.researchgate.net/publication/378876350

Konovalenko, I., Maruschak, P., Brezinová, J., Viňáš, J., & Brezina, J. (2020). Steel surface defect classification using deep residual neural network. *Metals*, *10*(6), 1–15. https://doi.org/10.3390/met10060846

Lee, S. Y., Tama, B. A., Moon, S. J., & Lee, S. (2019). Steel surface defect diagnostics using deep convolutional neural network and class activation map. *Applied Sciences (Switzerland)*, *9*(24). https://doi.org/10.3390/app9245449

Li, R., Zhao, L., Wei, H., OuYang, B., Zhang, M., Fang, B., Hu, G., & Tan, J. (2025). Optimized YOLOv8 for lightweight and high-precision metal surface defect detection in industrial applications. *Machine Learning*, *114*(10), 226. https://doi.org/10.1007/s10994-025-06857-3

Lin, H. I., & Wibowo, F. S. (2021). Image Data Assessment Approach for Deep Learning-Based Metal Surface Defect-Detection Systems. *IEEE Access*, *9*, 47621–47638. https://doi.org/10.1109/ACCESS.2021.3068256

Liu, Y., Yuan, Y., Balta, C., & Liu, J. (2020). A light-weight deep-learning model with multi-scale features for steel surface defect classification. *Materials*, *13*(20), 1–13. https://doi.org/10.3390/ma13204629

Lu, J., & Lee, S.-H. (2023). Real-Time Defect Detection Model in Industrial Environment Based on Lightweight Deep Learning Network. *Electronics*, *12*(21), 4388. https://doi.org/10.3390/electronics12214388

Lu, S., Liang, Y., Ren, Z., Yu, X., & Wang, X. (2025). FEP-YOLO: a lightweight steel surface defect detection method for resource-constrained devices. *Measurement Science and Technology*, *36*(7), 076016. https://doi.org/10.1088/1361-6501/aded27

Luyu Sun, & Yujun Zhang. (2025). Lightweight Steel Surface Defect Detection Based on YOLOv10. *Engineering Letters*, *33*(7), 2673–2685.

Ma, S., Zhao, X., Wan, L., Zhang, Y., & Gao, H. (2025). A lightweight algorithm for steel surface defect detection using improved YOLOv8. *Scientific Reports*, *15*(1), 8966. https://doi.org/10.1038/s41598-025-93469-5

Mao, H., & Gong, Y. (2025). Steel surface defect detection based on the lightweight improved RT-DETR algorithm. *Journal of Real-Time Image Processing*, *22*(1), 28. https://doi.org/10.1007/s11554-024-01611-9

Mery, D. (2020). Aluminum Casting Inspection Using Deep Learning: A Method Based on Convolutional Neural Networks. *Journal of Nondestructive Evaluation*, *39*(1). https://doi.org/10.1007/s10921-020-0655-9

Otieno, S., Kotak, N., & Parmar, C. (2023). *Defect Detection and Classification on Hot-Rolled Steel Using Machine Learning and Deep Learning Algorithms: A Comparative Analysis*. https://doi.org/10.21203/rs.3.rs-3281494/v1

Ren, R., Hung, T., & Tan, K. C. (2018). A Generic Deep-Learning-Based Approach for Automated Surface Inspection. *IEEE Transactions on Cybernetics*, *48*(3), 929–940. https://doi.org/10.1109/TCYB.2017.2668395

Selamet, F., Cakar, S., & Kotan, M. (2022). Automatic Detection and Classification of Defective Areas on Metal Parts by Using Adaptive Fusion of Faster R-CNN and Shape From Shading. *IEEE Access*, *10*, 126030–126038. https://doi.org/10.1109/ACCESS.2022.3224037

Shan, Z., Haoyan, H., Zhu, C., Du, S., Jing, H., & Haibin, W. (2025). RSM-YOLOv11: Lightweight Steel Surface Defect Segmentation Algorithm Research Based on YOLOv11 Improvement. *IEEE Access*, *13*, 111681–111698. https://doi.org/10.1109/ACCESS.2025.3583752

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0197-0

Tao, X., Zhang, D., Ma, W., Liu, X., & De Xu. (2018). Automatic metallic surface defect detection and recognition with convolutional neural networks. *Applied Sciences (Switzerland)*, *8*(9). https://doi.org/10.3390/app8091575

Wang, C., Wang, H., Jiang, Y., Yu, L., & Wang, X. (2025). CSCP-YOLO: A Lightweight and Efficient Algorithm for Real-Time Steel Surface Defect Detection. *IEEE Access*, *13*, 113517–113528. https://doi.org/10.1109/ACCESS.2025.3581024

Wang, S., Xia, X., Ye, L., & Yang, B. (2021). Automatic detection and classification of steel surface defect using deep convolutional neural networks. *Metals*, *11*(3), 1–23. https://doi.org/10.3390/met11030388

Wei, R., Song, Y., & Zhang, Y. (2020). Enhanced faster region convolutional neural networks for steel surface defect detection. *ISIJ International*, *60*(3), 539–545. https://doi.org/10.2355/isijinternational.ISIJINT-2019-335

Xia, X., Guo, J., Zhang, Z., Wang, L., & Guo, Y. (2025). LSDNet: Lightweight strip-steel surface defect detection networks for edge device environment. *Optics and Lasers in Engineering*, *186*, 108857. https://doi.org/10.1016/j.optlaseng.2025.108857

Yang, C., Lv, D., Tian, X., Wang, C., Yang, P., & Zhang, Y. (2025a). A lightweight convolutional neural network for surface defect detection in strip steel. *Measurement Science and Technology*, *36*(4), 046001. https://doi.org/10.1088/1361-6501/adbc10

Yang, C., Lv, D., Tian, X., Wang, C., Yang, P., & Zhang, Y. (2025b). LEIS-net: lightweight and efficient industrial surface defect detection network. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-025-02653-x

Z. Aboulhosn, A. Musamih, K. Salah, M. Omar, & Z. Aung. (2023). Detection of Manufacturing Defects in Steel Using Deep Learning with Explainable Artificial Intelligence. *IEEE Access*.

Zeng, W., You, Z., Huang, M., Kong, Z., Yu, Y., & Le, X. (2019). Steel Sheet Defect Detection Based on Deep Learning Method. *10th International Conference on Intelligent Control and Information Processing, ICICIP 2019*, 152–157. https://doi.org/10.1109/ICICIP47338.2019.9012199

Zhang, Y., Chen, C., Gao, Q., Wang, Y., & Fang, B. (2025). *A Lightweight Group Multiscale Bidirectional Interactive Network for Real-Time Steel Surface Defect Detection*.

Zubayer, M. H., Zhang, C., & Wang, Y. (2023). Deep Learning-Based Automatic Defect Detection of Additive Manufactured Stainless Steel. *Metals*, *13*(12). https://doi.org/10.3390/met13121987