Research Article

Lightweight Deep Learning Model for TFT-LCD Pixel Defect Detection Using MobileNetV2

J. Sheikshabjan[1], K. Dhanakodi[1], A.T. Rajamanickam[1], C. Josephine ArockiaMary[1] and S. Athinarayanan[2]

[1]*Department of Electronics, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore*-641020, *India*
[2]*Department of CSE, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, India*

Article history
Received: 30-06-2025
Revised: 04-12-2025
Accepted: 07-08-2026

Corresponding Author:
J. Sheikshabjan
Department of Electronics,
Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore-641020, India
Email: s.shabjan@gmail.com

Abstract: The proposed methodology aims to achieve high classification accuracy while maintaining computational efficiency and minimizing inference latency, making it highly suitable for real-time industrial inspection scenarios. This study evaluated five traditional transfer learning models (VGG16, ResNet50, MobileNetV2, Xception, and InceptionV3) for comparison in pixel defect detection. MobileNetV2-based implementation demonstrated a competitive average test accuracy of 0.9916, outperforming several baselines while maintaining significantly lower computational complexity. Based on our investigation, MobileNetV2 offers an optimal balance between accuracy and efficiency, making it a preferred choice over heavier architectures such as VGG16 and ResNet50, which, despite achieving comparable accuracy of 1.0, incur substantially greater computational overhead and longer inference times. These results underscore the suitability of MobileNetV2 as a lightweight and reliable alternative for automated visual quality inspection in display manufacturing, particularly where real time performance and operational efficiency are critical.

Keywords: Pixel Defect, Defect Detection, Transfer Learning, MobileNetV2

Introduction

Thin-Film Transistor Liquid Crystal Displays (TFT-LCDs) are now standard components in modern life, powering displays for smartphones, televisions, laptops, and a wide range of other electronic devices (Wu et al., 2023). The ability to deliver crisp visuals, vibrant colors, and compact form factors has revolutionized how we interact with technology. However, the intricate manufacturing process of TFT-LCDs can be susceptible to various defects that compromise display quality and user experience.

TFT-LCD defects come in two categories: macro (Mura) and micro (pixel). Mura defects are larger, visible inconsistencies in brightness or color. Micro defects, like dead pixels (black dots from all transistors failing) and stuck pixels (single-color dots from one or two transistors failing), are too small to see easily (Celik et al., 2022).

Even micro defects in a TFT-LCD panel can have a significant impact. Dead pixels, where a pixel is permanently off, or bright pixels, where a pixel is constantly lit, create visible blemishes that detract from the overall aesthetics of the display. More concerning are defects that affect color accuracy or uniformity, potentially leading to distorted or washed-out visuals. In critical applications like automotive instrument clusters, where clear and concise information display is paramount for driver safety, even slight variations in pixel performance can be disastrous. Imagine a scenario where a warning light appears, but due to a color defect, its hue is ambiguous, potentially delaying driver reaction time and jeopardizing safety (Axmann et al., 2020).

The diverse range of potential defects in TFT-LCD panels highlights the importance of robust and efficient detection methods during the manufacturing process. Early identification and elimination of defective panels not only ensure high-quality displays but also contribute to significant cost savings by preventing faulty products from reaching the final assembly stage. Traditional defect detection methods often rely on rule-based algorithms or manual visual inspection. However, these approaches have limitations. Rule-based algorithms (Nordin et al., 2025) often face challenges in keeping up with changing defect patterns, while manual inspection is laborious, error-prone, and produces inconsistent results among inspectors.

Recent advancements in deep learning, particularly in Convolutional Neural Networks (CNNs), have enabled the development of efficient and accurate methods for automated defect detection in TFT-LCD panels. CNNs excel at capturing spatial hierarchies and extracting discriminative features from image data, making them well-suited for identifying subtle pixel-level anomalies. When applied to industrial inspection systems, however, the successful deployment of CNN-based models demands careful attention to network architecture design, availability of annotated training data, and computational efficiency. These considerations become especially critical in high-throughput manufacturing environments, where real-time performance (Raiaan et al., 2023) and low-latency inference are essential.

Review of Literature

In earlier pixel defect detection methods, a notch multi-band Fourier filter (Ferreira et al., 2017) was used to locate and quantify the bright and dark dots. Qian et al. (2018) proposed a method using a quadrant gradient operator to accurately locate the sub-pixel defects. Guo et al. (2018) proposed another method using a notch filter and image registration. Moving beyond classical image processing, CNNs have become central to defect detection. This deep learning paradigm has revolutionized industrial quality control by enabling the automated, robust, and highly accurate identification of pixel-level anomalies in diverse manufacturing environments.

Deep learning has transformed computer vision, providing robust tools for automating defect detection across various industries. Among these tools, CNNs have emerged as the dominant architecture for image recognition and classification tasks. A key advantage of CNNs is their ability to automatically learn features from image data, eliminating the need for manual feature extraction. This capability is rooted in their distinctive architecture, which includes convolutional, pooling (or sub-sampling), normalization, and fully connected layers. Convolutional layers extract features from the input image, pooling layers reduce data dimensionality and introduce translation invariance, and fully connected layers perform classification or regression tasks using the learned features. By identifying latent features without human supervision, CNNs have become the most widely used algorithm in deep learning (Alzubaidi et al., 2021).

Deep neural network algorithms perform computer vision tasks such as classification, object detection, and segmentation. Unsupervised or semi-supervised learning of surface defects often utilizes object detection algorithms like Faster-RCNN, YOLO, and SSD. Çelik et al. (2021) proposed a Dead Pixel Network (DPNet) for LCD pixel detection. Celik et al. (2022) presented an image acquisition system and deep learning-based object detectors for LCD display pixel defect recognition. This method can test displays ranging from 32 to 65 inches. The image acquisition system uses Otsu thresholding to remove the LCD panel border from raw images. The system utilizes three different deep learning object detectors for defect identification: RetinaNet with a ResNet-101 backbone, YOLOv3 with a DarkNet-53 backbone, and M2Det with a VGG-16 backbone. These detectors are evaluated using F1-Scores, Precision, and recall metrics to determine their effectiveness in defect detection.

RetinaNet achieves superior recall performance and can detect all pixel-level defective images. Lin et al. (2022) introduced DCANet, a deep channel attention-based network for MURA defect detection. It utilizes antagonistic sample collection modelling based on frequency-domain Gaussian smoothing. Their ablation study shows that DConv + CA performs effectively without increasing network parameters. The proposed DCANet outperforms state-of-the-art methods like AlexNet, VGG-16, and ResNet-50 in experiments, while also being compared to detection approaches like SSD and YOLOv3. The antagonistic training approach achieves a significant average accuracy of 95.84% and reduces the time needed for sample annotation. Notably, this method is suitable for TFT-LCD automatic inspection even with a limited training dataset of 600 samples for detecting shallow MURA defects. Chien et al. (2022) propose UNISON-ADC, a CNN-based system for automatic defect classification. Unlike shallow-structure models, UNISON-ADC utilizes multiple processing layers for image recognition tasks, reducing the number of learnable weights.

The system classifies thirteen different TFT-LCD defect patterns, focusing on finding defects in the first patterned layer during glass substrate fabrication. While the trained model achieved a good average accuracy of 93.12%, it showed lower accuracy for specific defect patterns like dummy code and particle residue. Chang et al. (2022) propose a CNN model for multi-category classification of defective pixels on TFT-LCD panels. Their experimental setup used a 15.6-inch TFT-LCD with a specific resolution, brightness, and refresh rate. The classification process involves data processing, defect classification, and classification completion. Due to computational limitations, the original images were divided into smaller pieces. Following ISO/TR 9241 standards, a small number of defective pixels of various types is considered Grade-A. However, manufacturers rarely have images with such a high defect density.

To address this, data augmentation techniques like rotation and flipping were employed to create more training images. Since each pixel has three subpixels, a total of 12 defect image types were used. Initially, the model was trained and tested with 6,000 images (80% for training and 20% for testing) encompassing 15 categories (12 defective and 3 defect-free). Due to misclassifications caused by cutting some pixels during processing, the model initially achieved a sensitivity below 80% for specific blue defect categories. This issue was addressed by using a modified dataset with around 12,000 images. The final model achieved a minimum sensitivity of 96.97% for green half-bright defects and above 97% for all other categories, with six even reaching 100% sensitivity. Unfortunately, the classification process is time-consuming, with each panel requiring 467 seconds to screen one panel. Tomita et al. (2022) introduce a progressive hybrid model using machine learning. This model combines an 8-class CNN for multi-class classification, a 2-class ResNet for binary classification, and a separate 2-class CNN for another binary classification task. The model excels at classifying normal display images, which benefits the manufacturing process. However, its accuracy for classifying all eight defect categories requires improvement.

To address visual unevenness issues, Xie et al. (2021) proposed the UADD-GAN model for mura defect detection. In the display industry, collecting and annotating data is challenging due to the rarity of defects like mura. Song et al. (2021) thus introduced the Res-UNetGAN model to mitigate data imbalance. Liu et al. (2023) further advanced this by proposing a diffusion model for mura image generation. To ensure screen quality, Zheng et al. (2025) developed methods for high-quality defect dataset synthesis.

Experimental Setup for Dataset Preparation

An experimental platform was established for pixel-level fault emulation, comprising an Arduino Uno controlling a 2.4-inch TFT-LCD. This platform facilitates the injection of 1 to 15 random pixel defects per image via display control commands, referencing the dead pixel count ranges outlined in ISO 9241-302 Class 1, Class 2, and Class 3 panels. This experimental setup was developed to facilitate the systematic acquisition of images from a TFT-LCD module for the purpose of constructing a dataset. A REES52 Raspberry Pi camera module, interfaced with a Raspberry Pi Single-Board Computer (SBC), is mounted securely on the top surface (Pecolt et al., 2025) of a custom-built enclosure designed to isolate the display from external ambient light interference. The TFT-LCD screen is positioned at the base of the enclosure, connected via USB to a dedicated Arduino UNO for controlled image display. The enclosure ensures a consistent imaging environment, minimizing variations in lighting and reflections that could affect data quality. The camera, oriented directly above the display and connected through a flat ribbon cable to the Raspberry Pi, captures images in a top-down configuration as shown in Fig. (1). This setup was employed exclusively for dataset preparation, enabling the collection of both defective and defect-free dead pixel LCD images.

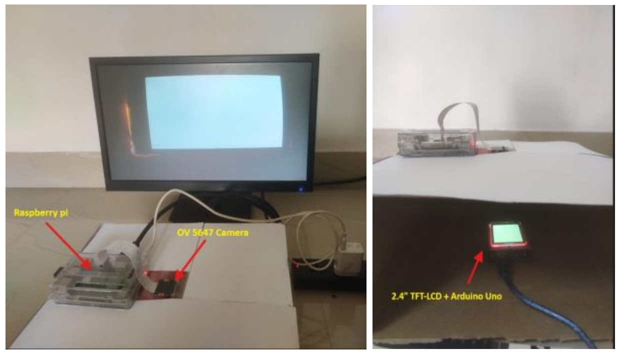

Fig. 1: Data acquisition system for TFT-LCD pixel defect

The height of the custom-built enclosure was systematically determined to ensure precise camera positioning for complete coverage of the TFT-LCD display, which was originally a vertically oriented module but was programmatically configured and physically mounted in a horizontal orientation for this setup. The display has a resolution of 240×320 pixels, and the reorientation was necessary to ensure proper alignment with the camera's field of view. In this configuration, the camera module used features a focal length of 3.6 mm and a sensor size of 3.6×2.7 mm. Based on these optical parameters, the Working Distance (WD) was calculated to be approximately 48.77 mm (Celik et al., 2022; Zheng et al., 2024). To accommodate practical considerations and ensure that the full display area, particularly the four corner edges where defects were intended to be displayed, was entirely within the frame, a small margin of approximately 2 cm was added to the calculated WD. This adjustment helped to guarantee uniform and consistent capture of the entire active screen area while preserving image quality and minimizing distortion. The enclosure effectively eliminated ambient light interference, ensuring optimal lighting conditions for quality image acquisition suitable for dataset preparation.

Due to the rarity of defective pixels in real-world scenarios and the impractical time frame spanning several months required to collect a sufficient sample, we programmatically simulated both defect-free and defective images to ensure dataset balance. Defect-free images across red, green, blue, and white backgrounds underwent augmentation via uniform RGB reduction (0.1–5%) with tone preservation, repeated 125 times each. For defective images captured against red, green, blue, and white backgrounds, pixel-level faults were emulated over 125 iterations per background, with each iteration randomly generating 1–15 dead pixels in both count and spatial location.

After collecting both defect-free and defective images using a Pi camera module interfaced with a Raspberry Pi SBC, we performed data augmentation (Kumar et al., 2024) to further balance the dataset. Specifically, flipping was applied to each category, increasing the number of images per class to 500. As a result, a total of 1000 images were prepared for model training.

To aid in the visual interpretation of single-pixel anomalies, representative sample images are selected and cropped to highlight pixel-level behaviour. As illustrated in Fig. 2, this visualization reinforces the spatial clarity of the simulated faults.

MobileNetV2 Architecture

MobileNetV2 is a lightweight and computationally efficient convolutional neural network architecture specifically designed for resource-constrained environments such as mobile and embedded devices (Sandler et al., 2018). Its core innovation lies in the use of inverted residuals and linear bottlenecks, allowing feature reuse across layers while preserving representational power. Each bottleneck block consists of a sequence: pointwise convolution, depth wise convolution, and another pointwise convolution without a non-linearity (i.e., a linear projection).

MobileNetV2 replaces standard convolutions with a combination of depthwise and pointwise convolutions, known collectively as depthwise separable convolutions. A depthwise convolution applies a single filter per input channel, drastically reducing the number of required operations. This is followed by a pointwise convolution (a 1×1 convolution), which performs channel-wise mixing and enables feature fusion across channels. This factorization greatly reduces computational complexity compared to traditional convolutions. Additionally, the architecture uses ReLU6 activation, which limits the output to a maximum value of 6, making it more robust for low-precision operations and further enhancing compatibility with quantization techniques. The MobileNetV2 architecture can be seen in Fig. (3).

In transfer learning, MobileNetV2 serves as a powerful feature extractor by leveraging pre-trained weights from large-scale datasets like ImageNet. The base model is typically frozen to retain learned representations, and a custom classification head (Gulzar, 2023) is used. This head commonly includes Global Average Pooling, followed by one or more Dense layers with ReLU activation, and a final output layer suited to the specific classification task (e.g., sigmoid or softmax). This modular design allows efficient adaptation to new tasks while retaining MobileNetV2's speed and compactness.

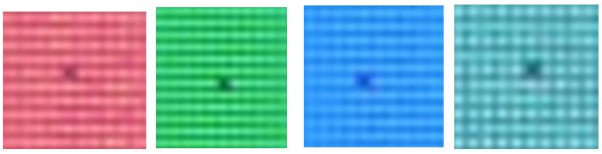

Fig. 2: Cropped Visualization of a Single Dead Pixel Across Backgrounds

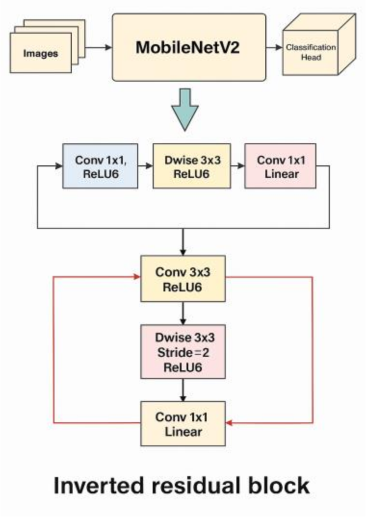

Fig. 3: Transfer Learning MobileNetV2 Architecture

Proposed Methodology

This study evaluated five traditional transfer learning models (VGG16, ResNet50, MobileNetV2, Xception, and InceptionV3) for comparison in pixel defect detection. For each base architecture, all layers were initially frozen to utilize pre-trained ImageNet feature representations. A tailored classification head was subsequently attached to the frozen backbone, comprising a global average pooling layer followed by dense-dropout-dense layers, with the output dense layer employing sigmoid activation for binary classification.

Comprehensive hyperparameter tuning was performed, systematically varying batch sizes (16, 32, 64, 128), activation functions (Tanh, ReLU, PReLU, LeakyReLU), and optimizers (Adagrad, SGD, Adamax, RMSprop, Adam, Nadam). Overfitting was addressed through evaluation of dropout rates (0.6, 0.5, 0.4, 0.3) alongside learning rates (0.01, 0.05, 0.001, 0.005) (Raiaan et al., 2024; Tuba et al., 2021). Initial comparative training across all architectures utilized the Adam optimizer over 40 epochs. Hyperparameters were set with a batch size of 32, ReLU activation, a 0.5 dropout rate, and a learning rate of 0.0001 optimized through prior tuning to balance stable performance and convergence. Early stopping with patience = 3 halted training after three consecutive validation epochs without improvement.

To assess the impact of dataset partitioning on model performance, various split ratios were tested using the lightweight MobileNetV2 architecture, and the LCD background color is red. The outcomes of these experiments are summarized in Table (1). Among the evaluated configurations, the 70/15/15 split produced the highest test accuracy (0.9733) and the lowest test loss (0.0499), indicating superior generalization. Allocating 70% of the data for training provides sufficient samples for stable feature learning, while the balanced 15% validation and 15% test sets ensure reliable hyperparameter tuning and unbiased performance estimation. This split therefore offers the optimal trade-off between learning capacity and evaluation robustness.

This study developed the model using Python 3.9 and the TensorFlow framework, with Jupyter Notebook employed for code execution and model training. The experiments were conducted on a computational platform equipped with an Intel Core i7-14700 K processor (20 cores, 3.40 GHz), 32 GB of DDR5-5200 RAM, an NVIDIA GeForce RTX 4060 Ti with 8 GB of VRAM, and running Windows 10.

Table 1: Impact of Dataset Partitioning on MobileNetV2 Performance Metrics

Split ratio (Train/ Validation/Test)	Test Accuracy	Test Loss
60/20/20	0.9850	0.07204
70/10/20	0.9850	0.0519
70/20/10	0.9900	0.0542
70/15/15	0.9933	0.0499

Results and Discussion

This section presents a comprehensive performance comparison between the other models and our proposed model. The evaluation emphasizes the strengths, limitations, and intended applications of each model, highlighting how architectural choices and task definitions impact their suitability for defect detection tasks. Table (2) shows a comparative performance analysis between five transfer learning models.

Although the VGG16 and ResNet50 baseline models demonstrated exceptionally high performance (achieving 1.0 test accuracy across all background screen datasets), our proposed MobileNetV2 model offers a superior and optimal balance between accuracy and computational efficiency, making it the preferred choice over these heavier architectures for real-world deployment. It achieved an average testing accuracy of 0.9916. Table (3) shows the performance metrics of our proposed model.

Based on the investigation of the model's accuracy and loss curves for the proposed architecture, the rapid initial climb is characteristic of transfer learning, particularly when employing a frozen backbone. The pre-trained weights provide a highly effective initialization, enabling the newly added classification head to quickly map these high-level features to the target classes. In all accuracy and loss curve plots, no widening gap is observed between the training and validation curves for any background-screen dataset. The accuracy and loss curves of the proposed model across all background-screen datasets are shown in Fig. (4-7).

The model described in Celik et al. (2021) is optimized for real-time industrial applications, particularly in production lines where rapid identification and classification of defective pixels are critical. Its focus lies in both defect classification and pixel localization. While its primary objective differs from our proposed method especially given our emphasis on classification rather than localization—we compare key evaluation metrics and model complexity, such as the number of parameters, to provide useful context regarding their respective efficiencies. In their proposed RetinaNet, which uses ResNet101 as the backbone, they reported a precision of 0.950, a recall of 0.997, and an F1-score of 0.973.

Table 2: Comparative performance analysis between five transfer learning models

Models	Screen Color	Test Accuracy	Test Loss	Confusion Matrix	No. Of parameters
VGG 16	Red	1.0	0.0158	[[75 00 75]]	
	Green	1.0	0.0296	[[75 00 75]]	
	Blue	1.0	0.0107	[[75 00 75]]	
	White	1.0	0.0098	[[75 00 75]]	138 M
Res Net 50	Red	1.0	0.0147	[[75 00 75]]	
	Green	1.0	0.0064	[[75 00 75]]	25.6 M
	Blue	1.0	0.0042	[[75 00 75]]	
	White	1.0	0.0057	[[75 00 75]]	
Mobile NetV2	Red	0.9933	0.0499	[[74 1075]]	
	Green	0.98	0.0875	[[72 30 75]]	
	Blue	1.0	0.0534	[[75 00 75]]	
	White	1.0	0.0618	[[75 00 75]]	3.5 M
Xception	Red	1.0	0.0373	[[75 00 75]]	
	Green	0.9533	0.1539	[[69 61 74]]	
	Blue	0.9733	0.0608	[[71 40 75]]	
	White	0.867	0.3012	[[63 128 67]]	22.9 M
InceptionV3	Red	0.9933	0.0439	[[74 10 75]]	
	Green	0.9533	0.1216	[[63 70 75]]	
	Blue	0.9867	0.0705	[[73 20 75]]	
	White	0.967	0.0998	[[73 23 72]]	23.9 M

Table 3: Performance metrics of MobileNetV2 trained on images with varying background conditions

Screen Color	Test Accuracy	Test Loss	Precision	F1 Score	Sensitivity
Red Defective Free	0.986	0.0571	0.986	0.986	0.986
Red Defective			0.986	0.986	0.986
Green Defective Free	0.98	0.0875	1.0	0.96	0.979
Green Defective			0.961	1.0	0.980
Blue Defective Free	1.0	0.0530	1.0	1.0	1.0
Blue Defective			1.0	1.0	1.0
White Defective Free	1.0	0.0618	1.0	1.0	1.0
Red Defective			1.0	1.0	1.0

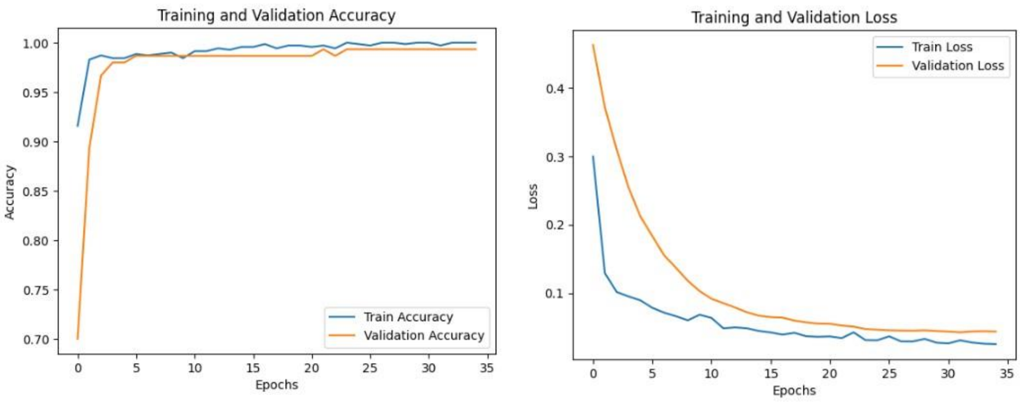

Fig. 4: Accuracy and loss curves for MobileNetV2 trained on red background images

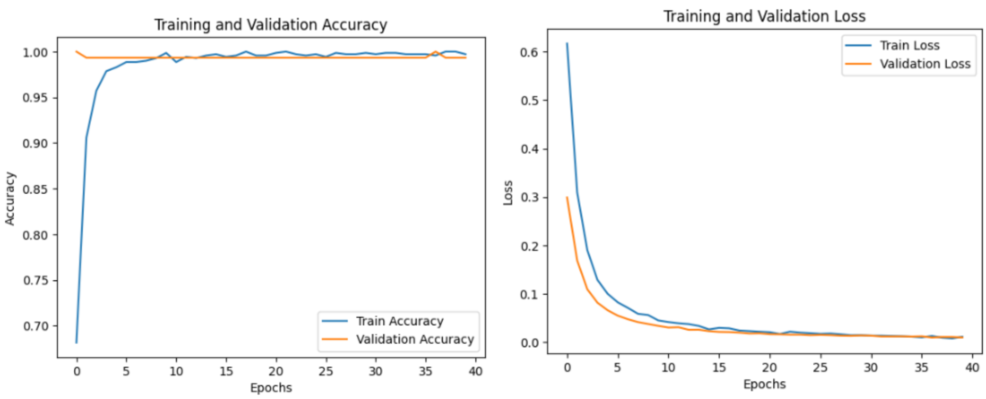

Fig. 5: Accuracy and loss curves for MobileNetV2 trained on green background images

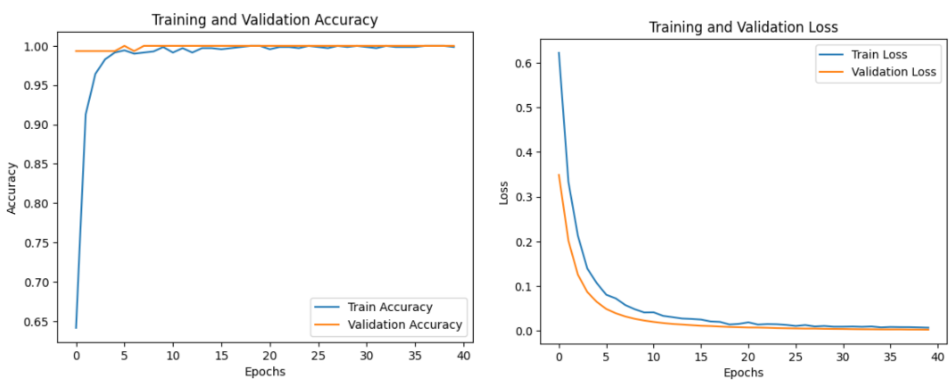

Fig. 6: Accuracy and loss curves for MobileNetV2 trained on blue background images

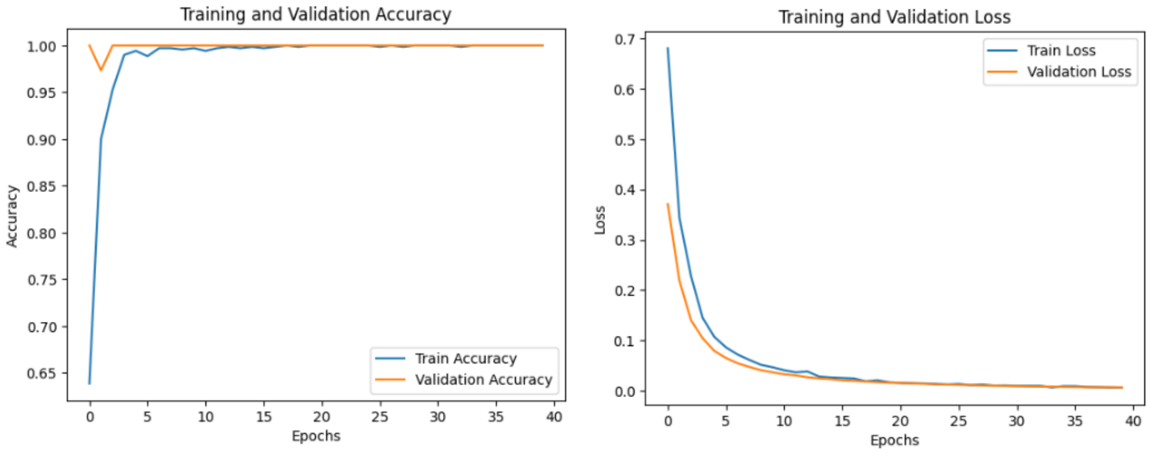

Fig. 7: Accuracy and loss curves for MobileNetV2 trained on white background images

In contrast, the method in Chang et al. (2022) bears a greater resemblance to our proposed binary classification model. However, a direct comparison of parameter count was not feasible due to insufficient details provided in Chang et al. (2022). They reported defect categories as full-dark defect, half-dark defect, full-bright defect, half-bright defect, and defect-free across red, green, and blue screen colors. However, for our analysis, we specifically investigated their model's sensitivity to two categories: full-dark defect and defect-free, evaluated across red, green, and blue screens. Their model achieved the following sensitivities: 0.884 for the blue full-dark defect, 0.947 for the green full-dark defect, and 0.8657 for the red full-dark defect.

Collectively, our findings validate the practical advantages of our proposed method, which outperforms the existing models, particularly in scenarios demanding real-time, efficient, and high-performing pixel defect classification.

Conclusion

This study introduces a lightweight and efficient model for detecting pixel-level defects in TFT-LCD panels, optimized for real-time industrial inspection scenarios. The proposed architecture was benchmarked against the complex VGG16 and ResNet50 models and other existing solutions using a range of evaluation metrics. Despite its reduced complexity, our model demonstrates competitive classification accuracy, while significantly improving computational efficiency and reducing inference latency. These characteristics make it particularly well-suited for deployment in practical manufacturing environments where speed and resource constraints are critical. Future research will focus on extending the current framework beyond classification to jointly address feature extraction, classification, and bounding-box regression on low-power platforms such as embedded and edge devices. In addition, this work will be further continued to detect a broader range of defects in TFT-LCD panels, with the goal of scientifically strengthening quality inspection processes and improving industrial product reliability.

Acknowledgment

The author gratefully acknowledges Dr. Manish Chowdhury for invaluable encouragement and helpful comments on the initial version of this manuscript, and sincerely thanks the journal's editors and anonymous reviewers for their constructive feedback.

Funding Information

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Author’s Contributions

J. Sheikshabjan: Contributed to conceptualization, Methodology, Software design, validation, writing of the original draft, preparation, review, and editing.

K. Dhanakodi: Contributed to supervision, review, and finalizing the manuscript.

A.T. Rajamanickam: Contributed to project administration and visualization

C. Josephine ArockiaMary and S. Athinarayanan: Contributed to data curation and investigation.

Ethics

The authors reiterate their commitment to promptly and transparently addressing any ethical issues that may be identified after publication, including taking appropriate corrective actions where necessary.

References

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *Journal of Big Data*, *8*(1), 1–74. https://doi.org/10.1186/s40537-021-00444-8

Axmann, B., Langner, F., Blankenbach, K., Vogelmann, M., von Gaisberg, A., Strobel, L., & Bauer, J. (2020). Advanced methods for safe visualization on automotive displays. *Journal of the Society for Information Display*, *28*(6), 483–498. https://doi.org/10.1002/jsid.909

Celik, A., Kucukmanisa, A., Sumer, A., Celebi, A. T., & Urhan, O. (2021). LCD Pixel Defect Detection using Shallow CNN Based Approach. *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 1–6. https://doi.org/10.1109/inista52262.2021.9548540

Celik, A., Küçükmanisa, A., Sümer, A., Çelebi, A. T., & Urhan, O. (2022). A real-time defective pixel detection system for LCDs using deep learning based object detectors. *Journal of Intelligent Manufacturing*, *33*(4), 985–994. https://doi.org/10.1007/s10845-020-01704-9

Chang, Y.-C., Chang, K.-H., Meng, H.-M., & Chiu, H.-C. (2022). A Novel Multicategory Defect Detection Method Based on the Convolutional Neural Network Method for TFT-LCD Panels. *Mathematical Problems in Engineering*, *2022*, 1–14. https://doi.org/10.1155/2022/6505372

Chien, C.-F., Ling, Y.-M., Kao, S.-X., & Lin, C.-H. (2022). Image-Based Defect Classification for TFT-LCD Array via Convolutional Neural Network. *IEEE Transactions on Semiconductor Manufacturing*, *35*(4), 650–657. https://doi.org/10.1109/tsm.2022.3199856

Ferreira, F. P., Forte, P. M. F., Felgueiras, P. E. R., Bret, B. P. J., Belsley, M. S., & Nunes-Pereira, E. J. (2017). Evaluating sub-pixel functional defects of a display using an arbitrary resolution camera. *Displays*, *49*, 40–50. https://doi.org/10.1016/j.displa.2017.06.001

Gulzar, Y. (2023). Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability*, *15*(3), 1906. https://doi.org/10.3390/su15031906

Guo, L., Li, S., Hu, W., Wu, J., Tu, B., He, W., Ou, X., & Zhang, G. (2018). Sub-Pixel Level Defect Detection Based on Notch Filter and Image Registration. *International Journal of Pattern Recognition and Artificial Intelligence*, *32*(06), 1854016. https://doi.org/10.1142/s0218001418540162

Kumar, T., Brennan, R., Mileo, A., & Bendechache, M. (2024). Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *IEEE Access*, *12*, 187536–187571. https://doi.org/10.1109/access.2024.3470122

Lin, G., Kong, L., Liu, T., Qiu, L., & Chen, X. (2022). An antagonistic training algorithm for TFT-LCD module mura defect detection. *Signal Processing: Image Communication*, *107*, 116791. https://doi.org/10.1016/j.image.2022.116791

Liu, W., Liu, C., Liu, Q., & Yu, D. (2023). Assigned MURA Defect Generation Based on Diffusion Model. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 4395–4402). https://doi.org/10.1109/cvprw59228.2023.00462

Nordin, H., Razak, B. A., Mokhtar, N., Jamaludin, M. F., & Mehmood, A. (2025). Automated mold defects classification in paintings: A comparison of machine learning and rule-based techniques. In *PLOS ONE* (Vol. 20, Issue 1, p. e0316996). https://doi.org/10.1371/journal.pone.0316996

Pecolt, S., Błażejewski, A., Królikowski, T., Maciejewski, I., Gierula, K., & Glowinski, S. (2025). Personal Identification Using Embedded Raspberry Pi-Based Face Recognition Systems. In *Applied Sciences* (Vol. 15, Issue 2, p. 887). https://doi.org/10.3390/app15020887

Qian, J., Li, W., Chen, B., Qian, J., Liu, F., & Zhou, X. (2018). Sub-Pixel Defect Detection for Super Large LCD Images Using Quadrant Gradient Operator. In *2018 IEEE 18th International Conference on Communication Technology (ICCT)* (pp. 1183–1187). https://doi.org/10.1109/icct.2018.8600215

Raiaan, M. A. K., Fatema, K., Khan, I. U., Azam, S., Rashid, Md. R. U., Mukta, Md. S. H., Jonkman, M., & De Boer, F. (2023). A Lightweight Robust Deep Learning Model Gained High Accuracy in Classifying a Wide Range of Diabetic Retinopathy Images. In *IEEE Access* (Vol. 11, pp. 42361–42388). https://doi.org/10.1109/access.2023.3272228

Raiaan, M. A. K., Sakib, S., Fahad, N. M., Mamun, A. A., Rahman, Md. A., Shatabda, S., & Mukta, Md. S. H. (2024). A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks. In *Decision Analytics Journal* (Vol. 11, p. 100470). https://doi.org/10.1016/j.dajour.2024.100470

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference*, 4510–4520. https://doi.org/10.1109/cvpr.2018.00474

Song, S., Yang, K., Wang, A., Zhang, S., & Xia, M. (2021). A Mura Detection Model Based on Unsupervised Adversarial Learning. In *IEEE Access* (Vol. 9, pp. 49920–49928). https://doi.org/10.1109/access.2021.3069466

Tomita, S., Siritanawan, P., & Kotani, K. (2022). In-line mura detection using machine learning and subspace method in display manufacturing. In *Journal of the Society for Information Display* (Vol. 30, Issue 12, pp. 915–930). https://doi.org/10.1002/jsid.1180

DOI: 10.3844/jcssp.2026.2633.2641

Tuba, E., Bačanin, N., Strumberger, I., & Tuba, M. (2021). Convolutional Neural Networks Hyperparameters Tuning. In E. Pap (Ed.), *Artificial Intelligence: Theory and Applications* (Vol. 973, pp. 65–84). Springer International Publishing. https://doi.org/10.1007/978-3-030-72711-6_4

Wu, H., Liu, Y., & Xu, Y. (2023). An LCD Detection Method Based on the Simultaneous Automatic Generation of Samples and Masks Using Generative Adversarial Networks. In *Electronics* (Vol. 12, Issue 24, p. 5037). https://doi.org/10.3390/electronics12245037

Xie, C., Yang, K., Wang, A., Chen, C., & Li, W. (2021). A Mura Detection Method Based on an Improved Generative Adversarial Network. *IEEE Access*, *9*, 68826–68836. https://doi.org/10.1109/access.2021.3076792

Zheng, S., Zhao, Y., Chen, X., & Luo, S. (2025). An LCD Defect Image Generation Model Integrating Attention Mechanism and Perceptual Loss. *Symmetry*, *17*(6), 833. https://doi.org/10.3390/sym17060833

Zheng, Y., Wang, X., Jiang, Z., Xu, J., Yuan, R., Zhao, Y., Zhang, H., Liu, C., & Wang, Q. (2024). Adaptive multiscale microscope with fast zooming, extended working distance, and large field of view. *Light: Advanced Manufacturing*, *5*(1), 1–8. https://doi.org/10.37188/lam.2024.008