Received: 9 September 2024 | Revised: 16 April 2025 | Accepted: 22 May 2025 | Published online: 7 June 2025

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 4(2) 215–223
DOI: 10.47852/bonviewAIA52024309

Automated Evaluation of Smartphone Screen Damage: A CNN-Based Image Recognition System

Ogechi Chiama[1,*] and Swathi Ganesan[1,*]

[1]*Computer Science and Data Science Department, York St John University, UK*

Abstract: The automation of defect detection in smartphone screens is important in ensuring fairness in the refurbished smartphone market. Conventional methods for evaluating screen damage rely on manual assessments, which are subjective, unreliable, and susceptible to human error. This paper presents a deep-learning-based approach using a convolutional neural network (CNN) to classify smartphone screens as cracked or uncracked. The CNN model was trained using a custom dataset of smartphone images, with an accuracy rate of 92.0% in classifying screen damage. CNN outperformed conventional machine learning methods in terms of feature extraction, resulting in higher precision in defect detection. However, the model encountered difficulty in identifying subtle crack patterns, fluctuations in light conditions, and overfitting resulting from the dataset's limited diversity. Future work will focus on expanding the dataset, refining methods for data augmentation, and exploring alternative algorithms such as a transformer-based or hybrid model to improve model quality. This study aims to facilitate the standardization of defect assessment in the refurbished smartphone industry by automating screen damage detection, thereby increasing consumer confidence and boosting resale market efficiency.

Keywords: convolutional neural network (CNN), screen damage classification, smartphone screens, machine learning, deep learning, mobile application, defect detection

1. Introduction

In the digital age, computer vision has evolved into a transformative technology that enables machines to interpret and comprehend the visual world with human-level intelligence [1]. One significant application is defect detection, which is crucial for quality control in many industries, including manufacturing and electronics resale. In the refurbished smartphone market, screen damage analysis is very important for determining the value of the product and maintaining a fair price. However, current techniques for assessment heavily rely on manual inspection, which can be time-consuming, inconsistent, and susceptible to human error. The core objective of computer vision is to enable machines to analyze and interpret images or video content, allowing them to recognize patterns, detect anomalies, and understand scenes in a manner that mimics human visual perception [2].

According to the study reported in Reference [3], image recognition is a key area of study that focuses on analyzing key features of images rather than relying solely on raw pixel values. Convolutional neural networks (CNNs) have greatly improved image recognition accuracy by learning structured patterns from image data. When applied to a refurbished market, this recognition ability can be integrated with pricing algorithms to enable more accurate product valuation, increasing transparency and trust between buyers and sellers.

Although current research on image recognition spans from feature extraction methods to advanced deep learning models, its real-world application remains a challenge. Specifically, product variation, degree of damage, and user preference all interfere with image assessment and automated pricing. Further work is required to create an effective system that can address these challenges and offer a scalable and fair pricing solution.

Prior research on defect detection applied traditional machine learning techniques such as support vector machines and decision trees. Although these techniques perform well in detecting basic defects, they often struggle to identify more complex and fine-grained patterns of screen damage such as microcracks, surface abrasions, and impact-induced distortions. To address this limitation, this study applies a deep learning approach using CNN, which can learn hierarchical visual representations directly from image data. A custom CNN model was developed and trained on a specific dataset of images, which maximizes feature extraction for fine-grained defect classification. This study aims to improve pricing transparency for consumers and sellers by helping in automating quality assessment in the refurbished smartphone market, thereby lowering variations in damage evaluation.

2. Literature Review

In recent years, significant progress has been observed in image recognition technology, rooted in computer vision and deep learning. Businesses are increasingly utilizing photo recognition technology as big data and AI continue to advance. Digital images are essential for object recognition and information retrieval, known as image retrieval. Image recognition aims to search and retrieve similar images based on a query and user-specified information. This process relies on pixel values, with images composed of pixels arranged in rows and columns. Image recognition technology closely resembles the manner the human brain processes images. Key steps involve preprocessing, feature

*Corresponding authors: Ogechi Chiama, Computer Science and Data Science Department, York St John University, UK. Email: ogechi.chiama@yorksj.ac.uk and Swathi Ganesan, Computer Science and Data Science Department, York St John University, UK. Email: s.ganesan@yorksj.ac.uk

extraction, and classification, which aim to improve image processing accuracy by improving pertinent information [4].

Li [5] assessed different deep learning methods used in image recognition, which were centered on CNNs, recurrent neural networks, and generative adversarial networks. The study highlighted the widespread use of CNNs in image recognition due to their potential to facilitate preprocessing and enhance feature extraction efficiency. The study also addressed the development of deep learning methods and their impact on prioritizing inputs in image recognition tasks. Similarly, Esteva et al. [2] highlighted how CNNs have revolutionized medical computer vision by improving the accuracy of diagnosis and the interpretation of imaging data. This progress results in better patient care outcomes and empowers healthcare professionals to make more informed decisions.

Artificial neural networks (ANNs) were used in the study reported in Reference [6] to simulate the brain's structure for computing purposes. Despite their remarkable capabilities, they are still constrained by modern technology. These networks provide a streamlined representation of the brain's complexity using interconnected layers. CNNs adopt a different approach by integrating convolutional layers, which helps in hierarchical information processing. These layers enable effective data management, which produces more precisely targeted results. This study paves the way for understanding and creativity by proving the interaction between biology and technology in neural networks.

Bhatt et al. [1] emphasized CNN architectural advancement, which accelerated the creation of potent variants. The authors discussed how CNN design progress resulted in improved efficiency in image processing. CNNs have been extensively tested for image recognition in several fields.

Image recognition technologies can analyze visual cues to support pricing strategies, as shown by a food image identification system that uses CNN to estimate prices [7]. Wei et al. [8] addressed issues such as appearance variations, lighting, and cluttered backgrounds when using deep learning for retail product identification. These findings show the significance of image recognition systems for data-driven pricing decisions, improved inventory management, and heightened competitiveness in ever-changing market conditions. The two experiments underscored the capacity of CNN and deep learning to precisely evaluate aesthetic features, hence influencing pricing strategies across different markets.

2.1. Defect detection

Recent advancements in CNN-based defect identification have shown significant promise in several quality control applications, particularly in identifying smartphone screen defects. The field has benefited from the robustness of CNN architectures, which excel in object detection, image segmentation, and classification tasks.

CNNs have been successfully detecting cracks, scratches, and damages on phone screens, making it well suited for quality control in mobile devices. Chen et al. [9] developed a method that combines R-CNN with an efficient channel attention mechanism to detect defects in liquid crystal displays. With an accuracy of 91%, the model revealed the efficacy of CNN-based methodologies in processing multifaceted visual inputs. This is essential to the identification of defects in smartphone screens, where high-resolution images must be processed swiftly and precisely.

The study conducted by Ma et al. [10] on GoogLeNet-based CNN for defect detection on smartphone surfaces reported an accuracy of 99.5% in identifying surface defects such as scratches. Yu and Yang [11] presented a Faster R-CNN model improved by a multihead attention mechanism for detecting defects in cell phone screens, with an average accuracy of 95.71%. In addition, the smartphone screen glass dataset, which consists of over 2,500 high-quality images classified into seven defect categories, was introduced [12].

Dung and Anh [13] developed a deep FCN-based technique for identifying cracks in concrete buildings, with an average accuracy of almost 90%. This FCN-based technique was also effective in identifying various types of damage (such as cracks, spalling, efflorescence, and holes) in concrete structures. Although ANNs have been used to detect concrete cracks [14], their limited processing capacity makes it impossible for them to precisely identify local features in images. Conventional machine learning methods depend on manual extraction of low-dimensional features, but because of background noise, this can be time-consuming and inefficient.

Current defect detection methods have several limitations despite their promising results. Many deep-learning techniques, such as Faster R-CNN, ANNs, and Mask R-CNN, are not suitable for real-time detection on devices with limited hardware capabilities because they need substantial computational resources. This presents a challenge for practical deployment in mobile applications, where rapid processing and minimal computational requirements are essential. Although CNNs elevate the detection rate, they can still result in false positives, especially in cluttered or noisy backgrounds. In some cases, defects are small or subtle, making them difficult to detect accurately without a high-resolution image.

The developments of automated quality control systems in sectors such as smartphone manufacturing are pertinent in CNN-based defect identification. The efficiency of CNN-based methodologies in detecting defects on smartphone screens has been proven by recent studies. These models offer promising solutions to issues in automated quality control in the smartphone sector using advanced techniques such as attention mechanisms and curated datasets. This literature review highlights how ongoing advancements in CNN architectures continue to drive innovation in the field, making them a suitable choice for further investigation in smartphone screen defect detection.

This study aimed to develop a lightweight and efficient CNN architecture to alleviate the computational load of traditional CNN models. Our model has a small size and computational demands yet retains a high accuracy by employing techniques such as model pruning and quantization. This makes real-time detection possible on smartphones and allows deployment on devices with limited hardware.

In summary, although existing CNN-based approaches have greatly improved defect detection, issues with computational cost, accuracy, and scalability remain. Our proposed method address these drawbacks by providing a more precise and accurate approach that is scalable to smartphone screens. This method offers a promising advancement in automated defect detection in various materials by integrating advanced techniques such as attention mechanisms and optimizing the model for real-time detection on mobile devices.

3. Research Methodology

A technical overview of the proposed CNN that detects cracked screens in mobile phones is provided. This section outlines the research design, along with the CNN model selection, hyperparameters, data augmentation strategies, and dataset biases. The model was trained on publicly available phone images, which can correctly identify cracked phone screens by analyzing the visual pattern of the input image.

3.1. Research design

The research design proposed the development of a CNN model for the identification of cracked and uncracked phone screens. Raw image data of Samsung phones were collected and processed to develop a deep-learning model for image classification. A CNN model was trained and tuned with hyperparameters using evaluation criteria such as accuracy, precision, and recall gauge performance.

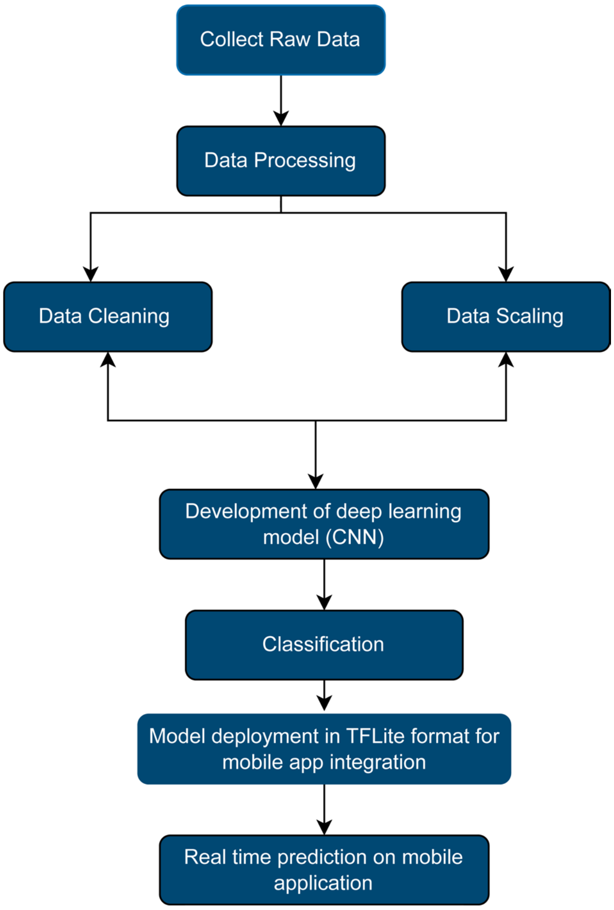
The dataset was randomly split into training, validation, and testing sets to reduce bias and ensure adequate representation of both cracked and uncracked phone images. The completed model was converted into a mobile-compatible format and integrated into the application for prediction.

To uncover patterns and correlations, image data were collected and analyzed using quantitative methodology. The CNN model was tested using images of both cracked and unblemished Samsung phone screens. The overall design structure for real-time prediction on mobile applications, from model training to deployment, is shown in Figure 1.

3.2. Data collection

Acquiring labeled data for deep learning was more challenging than building and training the machine learning model. The quantity

Figure 1
Project design structure



and quality of data used to train the model greatly affect its performance. Deep learning models depend on large amounts of data to make accurate predictions, and poor performance can result from a lack of sufficient data. Collecting and processing large datasets can be time-consuming and tedious.

Using a Google Chrome plugin, web scraping was utilized in this study to gather 635 publicly accessible images of both cracked and uncracked Samsung phones from online sources, of which 322 were cracked and 313 were uncracked. This dataset was hosted on Google Drive for easy access and processing, and the images were categorized and stored in standard file formats.

3.3. CNN architecture

CNNs are feedforward neural networks designed for analyzing image data and have been successful in image classification tasks by learning and extracting features automatically. Neurons are the basic building blocks of CNNs, which consist of multiple layers for analyzing different parts of an image. The CNN architecture draws inspiration from visual perception. CNNs are widely used in deep learning and can uncover complex data structures by consolidating properties from previous layers [15, 16].

The flowchart of the CNN model shown in Figure 2 was created to classify images of cracked and uncracked phone screens. To extract features and make predictions, the architecture consists of convolutional layers, max pooling, a flattening layer, and a dense layer.

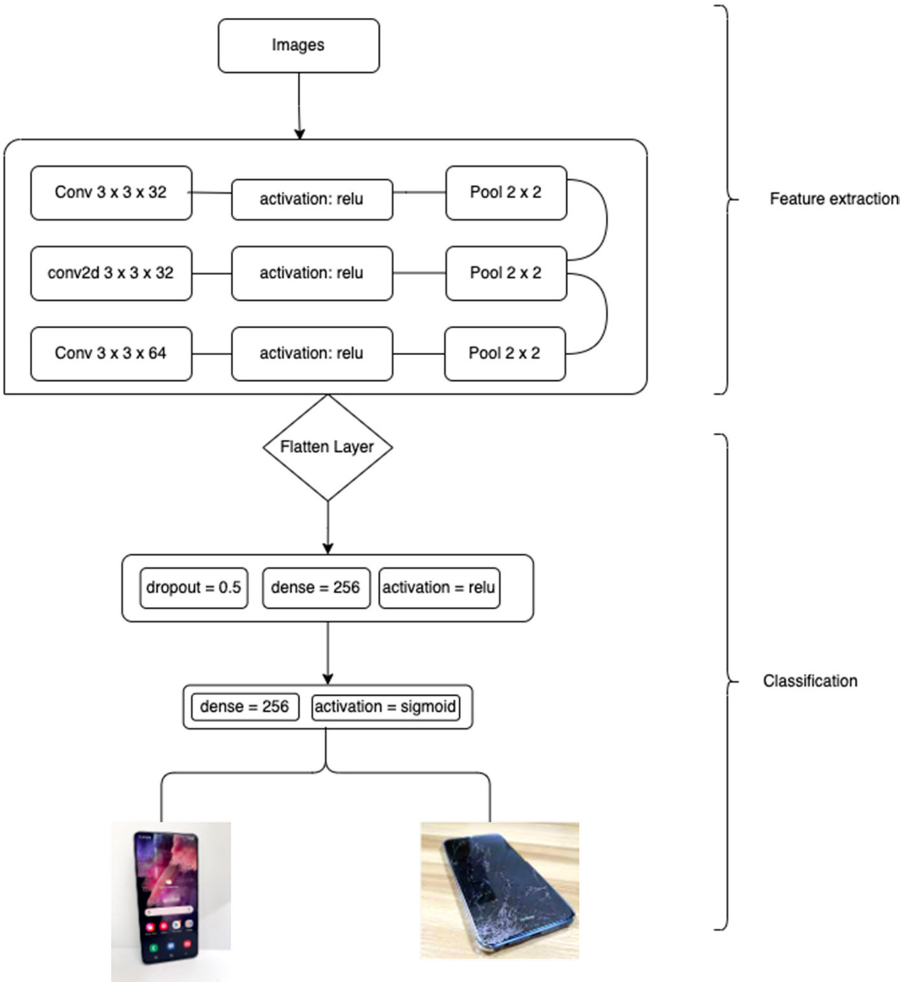
The first layer, known as the input layer, accepts raw input data of an image with dimensions of 150 × 150 × 3, corresponding to the image's height, width, and RGB color channels, respectively. This layer offers a structured format for the data, ensuring that it is appropriately scaled and normalized, making it easier for subsequent layers to process and extract useful features, which helps the CNN learn from the data and perform well in classification.

CNN's fundamental building blocks are convolutional layers, also known as Conv2D layers. They use an array of filters to perform convolution operations on input images. These layers are responsible for identifying features such as edges, textures, and patterns. A convolutional layer's depth is determined by the number of filters that it has. Convolution is an iterative method used to discover significant patterns that involves sliding a filter across the layer and computing the dot product between the filter and the layer values, which develops feature maps [17]. These feature maps are the output of the convolutional layer, which detects features and provides input for subsequent layers in the neural network architecture. These layers can include additional convolutional layers or a final layer that makes a prediction based on the patterns discovered from the data [18].

This study used 32, 32, and 64 filters for the first, second, and third convolutional layers, respectively, using a 3 × 3 kernel size and a rectified linear unit (ReLU) activation function. After each convolution operation, a ReLU activation function was applied to introduce a nonlinear behavior into the model, which converted all negative values in the feature map to zero, allowing the network to identify and understand complex patterns in the data. The activation function decides if a neuron is activated by computing the weighted sum and then adding bias to introduce nonlinearity to the output. These functions enhance the network's capacity to understand and represent complex relationships between the input and output.

The MaxPooling2D layer, also known as downsampling, is an intermediary layer in many neural networks that connects the fully connected (FC) layer and the convolutional layer. It was used to minimize the spatial dimensions of the feature map while keeping the most essential characteristics. The technique reduces computational

Figure 2
Flowchart of the CNN pipeline for cracked/uncracked classification



complexity and helps in generalizing the model by providing translation invariance. In accordance with the study reported in Reference [18], max pooling is the most common type of pooling strategy used in neural networks, with strong characteristics such as edges or lines being well preserved by max pooling, in contrast to average pooling, which tends to blend activations.

The flattening layer converts the feature maps, which are a multidimensional array (3D) from the Conv2D and MaxPooling2D layers into a one-dimensional (1D) array, making it suitable for the dense layer to understand. The FC (dense) layers can identify global patterns in the data, given that every neuron in one layer is linked to all neurons in the next layer. The first dense layer in our model consists of 256 units with ReLU activation, which helps in incorporating nonlinearity into the model and allows the model to learn complex patterns. Dense layers gather data from all regions of the image to determine the class by linking all neurons in the image. The second dense layer serves as the output layer, containing a single unit and using a sigmoid activation function. This setup is utilized for the binary classification of phone images. The sigmoid function returns a probability ranging from 0 to 1. The probable outcome suggests that the input image matches a specific class. For an output closer to 1, the model predicts an intact image, and when the output is closer to 0, the model predicts a cracked image. Table 1 presents the parameters of the CNN used to optimize the model for improved performance and detection.

Table 1
CNN hyperparameters

Hyperparameter	Value
Learning rate	0.001
Optimizer	Adam
Batch size	32
Epochs	20
Activation function	ReLU (hidden layers), sigmoid (output layer)
Kernel size	3 × 3
Pooling	MaxPooling2D (2 × 2)

3.4. Data augmentation

To improve model generalization and reduce overfitting, data augmentation methods were adopted. Classification accuracy can be improved by 5% using data augmentation [19]. Table 2 shows the parameters used for augmenting the input image

ImageDataGenerator was used to rescale the pixel values to the [0,1] range. The efficient operation of any deep learning algorithm is

Table 2
Data augmentation parameters

Rotation	To replicate different views, images were arbitrarily rotated by up to 15°.
Flipping	To increase the diversity of the input image.
Noise addition	To simulate real-world faults.
Scaling	Images were randomly zoomed by up to 10%.

greatly affected by the normalization strategy. Normalization aims to generate high-quality data that can be fed into the deep learning model, which involves converting the input into a standardized scale. The normalized data helps in retaining a more consistent gradient throughout the CNN layers, preventing them from increasing or shrinking. In addition, this speeds up model training and reduces the probability of poor convergence with the optimizer conducting enhanced weight updates.

A total of 635 images were gathered for this study. Out of these 635 images, 80.48% (511 images) were labeled for training; 15.71% (100 images), for validation; and the remaining 3.81%, for testing. This sectioning allows for thorough model training, tuning, and evaluation, ensuring a robust process with sufficient data for validation and testing.

3.5. Dataset bias and handling

Potential biases in the dataset were considered and mitigated as follows:

1) Imbalanced representation of cracked versus uncracked screens: The dataset was roughly split (322 cracked and 313 uncracked images) to provide balanced training data. Augmentation was also added to expand the training dataset.
2) Lighting conditions: Images from various lighting environments were included to improve robustness.
3) Device variations: Although only Samsung phone screens were used, future work aims to include other brands of phone screens to improve generalization.

3.6. Evaluation metrics

Numerous performance indicators were used to assess the effectiveness of the trained CNN model, with accuracy being the key measure. The reliability of the model is demonstrated by high accuracy on new data. Poor accuracy indicates that improvement through training or tuning is required.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

where "TP" denotes true positives; "TN," true negatives; "FP," false positives; and "FN," false negatives in this context.

The confusion matrix, F1-score, precision, and recall are additional crucial measures used in this study. The key to precision is how well a model distinguishes between real positives and false positives among all occurrences that it flags as positive. It is computed as the ratio of true positives to the sum of true positives and false positives. A precision score that is close to 1 indicates that the model performs well in recognizing real positives, whereas a score that is lower than 0.5 indicates a high number of false positives, which may be the result of class imbalances or incorrect parameter tuning.

$$Precision = \frac{TP}{TP+FP}$$

Recall assesses the model's accuracy in identifying the true positive cases in the data. It is often referred to as sensitivity or positive rate. A high recall rate means that the model reduces false negatives and captures most positive cases.

$$Recall = \frac{TP}{TP+FN}$$

The F1-score is a balanced measure of a model's accuracy, considering both precision and recall by taking their harmonic mean. A high F1-score indicates balanced performance with high precision and recall, suitable for imbalanced classification problems. Conversely, a low F1-score indicates limited insight, with low recall indicating inaccurate positive case identification and low precision implying many incorrect positive predictions.

$$f1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

As illustrated in Figure 3, the confusion matrix provides a comprehensive visual overview of classification performance in a tabular style by comparing expected and actual outcomes to assess the accuracy of a classification model.

3.7. Flutter application

This study introduced a mobile application that was developed using Flutter, an open-source UI framework for cross-platform application development, to assess the image recognition capabilities of the model. The application functions as an easy-to-use interface for image classification and real-time model performance tracking. Flutter offers a vast array of user interface (UI) components for creating engaging UIs, with a focus on real-time image recognition. Its integration of deep learning model plugins, multiplatform compatibility, and a rapid development process led to its selection. The trained model was converted into TensorFlow Lite format, which enabled efficient mobile device deployment. Post-training quantization techniques were implemented to adjust model size and accuracy. The TensorFlow Lite Flutter plugin ensures compatibility with both iOS and Android. The model was loaded dynamically during runtime to perform optimal mobile performance.

This study establishes an extensive framework for integrating deep learning in the assessment of mobile device damage and the potential for notable advancement in automated image classification.

4. Results

In this section, the outcome of the trained CNN model shows how well it performed in categorizing images of cracked phone screens and

Figure 3
Confusion matrix representation

those that are intact. In addition, its performance is compared with that of other deep learning architectures. This analysis examines the trade-offs between precision, computational efficiency, and the practicality of real-time implementation.

4.1. Comparative analysis

The performances of various deep learning models are summarized in Table 3.

With the addition of extensive preprocessing and an optimization strategy, the model could identify cracked and uncracked phone screens, achieving 92.0% accuracy. The performance metrics shown in Table 4 indicates model precision of 1.00 for cracked screens and 0.86 for uncracked screens, confirming reliable cracked screen detection. Recall is 0.83 for cracked screens and 1.00 for uncracked screens, with high F1-scores of 0.91 and 0.92, implying a balanced performance.

Figure 4 shows the training and validation performances of the CNN model over 20 epochs. The left graph demonstrates accuracy, where both the training and validation accuracy values steadily improve, with a small fluctuation in the validation accuracy due to dataset variability. The right graph presents the loss curves, which demonstrate a steady decrease over epochs in both training and validation losses, indicating effective learning and reduced errors over time. This suggests that the model is learning effectively and improving prediction precision. Fluctuations in validation loss are normal, reflecting dataset variations. The overall decline in loss without a significant gap between losses indicates that the model has not overfitted to the training data.

The confusion matrix shown in Figure 5 displays high true positives, true negatives, low false positives, and false negatives when categorizing smartphone screens as cracked or uncracked. By averaging the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions, the matrix shows the model's performance.

TP: the number of correctly detected cracked screens.

TN: the number of correctly detected uncracked screens.

FP: the number of screens incorrectly classified as cracked when they are not.

FN: the number of cracked screens misclassified as being in good condition.

Table 4
Precision, recall, and F1-score of the implemented model

	Precision	Recall	F1-score
Cracked = Class 0	1.00	0.83	0.91
Uncracked = Class 1	0.86	1.00	0.92

The high values for TP and TN suggest that the model effectively detects screen conditions, whereas the low values for FP and FN reflect minimal misclassification. This matrix is an integral tool for evaluating the accuracy and validity of the model in practical applications.

4.2. Real-world application

Figure 6 [23] displays how the trained CNN model was successfully incorporated into a mobile application, which enables real-time screen damage predictions. Using this software, users can pick images of smartphone screens and determine swiftly if the screen is intact or damaged.

The CNN model's ability to learn and generalize was improved by effective data preprocessing involving feature scaling, data augmentation, and normalization. These steps lowered the risk of overfitting. Efficient training with a well-constructed CNN architecture and the Adam optimizer notably improved the convergence and learning process. The capacity of the model to generalize new data was ensured using a balanced dataset that included separate training, validation, and test datasets. The model's integration into a mobile application and its performance in real-world tests show its practical usability and the quality of the training process. It is easy to upload images, and the model assesses the visual input in a timely manner to obtain precise results. This helps consumers make better-informed decisions regarding their product and improves user experience.

4.3. Potential impact

Automated detection of cracked phone screens offers substantial benefits, which can be used across numerous industries, such as phone repair services, refurbished smartphone markets, and manufacturing quality control. Businesses can greatly benefit by improving operational

Table 3
Comparison of deep learning models for smartphone screen damage detection

Model	Strength	Limitation	Accuracy
Proposed CNN model	It is lightweight, performs well in generalizing new data, and is more optimized for real-time mobile deployment. Special hyperparameters such as ReLU activation are used during training. It offers a greater capability for feature extraction and removes the need for manual engineering.	The proposed model can be improved when exposed to larger datasets or several images. In addition, transfer learning integration can enhance performance.	92.0%
ResNet-50	This architecture optimizes feature extraction using residual blocks while achieving remarkable accuracy.	ResNet requires greater processing power and a larger number of parameters. It may not be ideal for real-time deployment.	97.5% [20]
VGG-19	This model is more efficient in transfer learning and has a strong feature extraction ability.	Because of its heavy architecture and computational cost, it poses a challenge for deployment in resource-constrained environments (e.g., mobile apps) due to its lengthy training time.	90% [21]
EfficientNetV2	This model delivers an optimal ratio between accuracy and computational speed.	An enormous amount of computing resources is needed during the first phase of training.	93.2% [22]

Figure 4
Performance accuracy and loss over 20 epochs

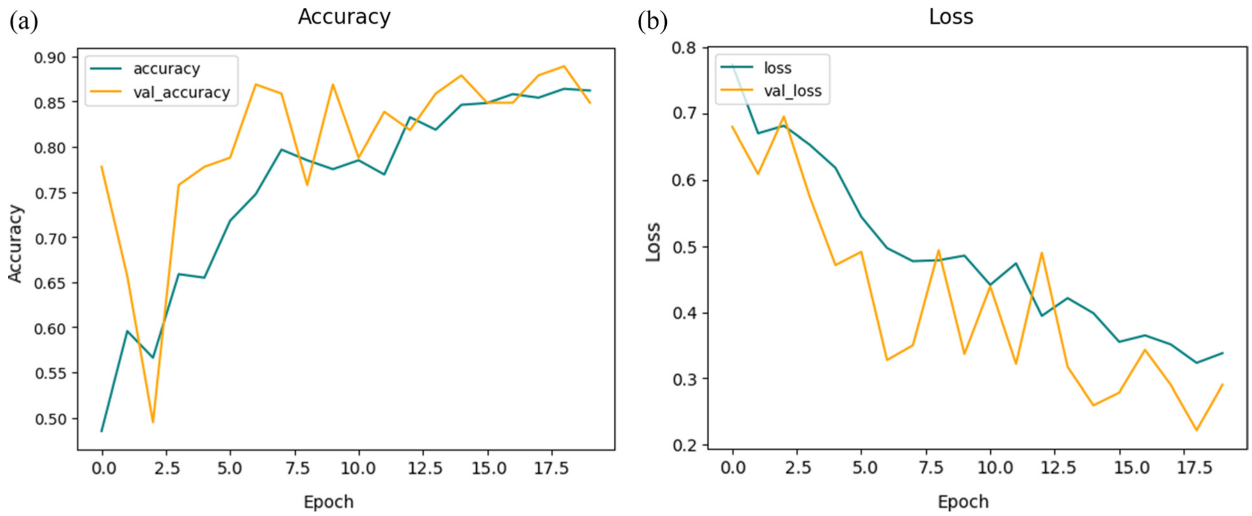
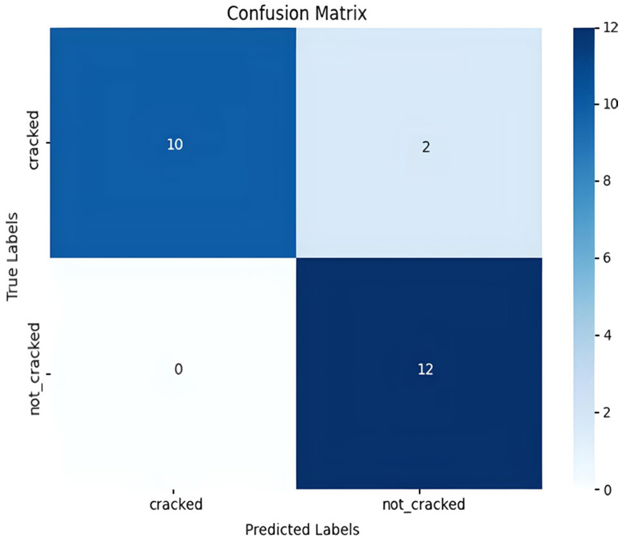


Figure 5
Model performance based on the confusion matrix



efficiency, reducing human error, and improving customer trust. By streamlining diagnostic procedures, automated crack detection enables technicians to assess damage quickly and accurately. Standard assessments ensure that customers receive fair and consistent repair suggestions, reducing disputes over unnecessary repairs.

In the refurbished smartphone market, sellers can use automated defect detection tools to verify and certify the condition of used phones, reducing human bias in cases where damaged devices are marketed as intact. Buyers benefit from increased openness, allowing them to make more informed purchasing decisions based on confirmed screen conditions. Using this technology, marketplaces and resale platforms can offer automated condition reports, decreasing the need for arbitrary assessments.

In manufacturing and quality control, automated defect detection can be integrated into production lines to identify screen defects early, lowering manufacturing flaws and preventing faulty products from reaching consumers. This procedure lowers returns and warranty claims, saving costs for manufacturers and improving brand reputation.

4.4. Limitations and ethical considerations

This study developed a CNN deep learning model that classifies cracked and uncracked phone screens, with an accuracy rate of 92%. The model still faces drawbacks arising from the diverse shapes, sizes, and patterns of cracks, making it difficult for the model to accurately detect unusual or less common cracks that are not present in the training data. The model may also struggle to adjust to various phone models with differing screen materials or designs. Another issue is detecting minute or subtle cracks that blend into the phone's surface, which can hamper the model's ability to identify all damaged screens. The model's accuracy is also affected by factors such as poor lighting or distracting backgrounds in the input images. Addressing these challenges will require training the model on a more diverse dataset,

Figure 6
Application prediction of a cracked and intact phone screen

employing additional model techniques, and making sure that images are captured under standardized conditions. Although the model is lightweight, deploying it for real-time use on mobile applications still requires optimization to balance speed and accuracy without excessive computational load.

Ethical considerations include the potential risk of misclassification, which occurs when an intact screen is incorrectly classified as cracked while failing to detect genuine cracks. This may pose a risk of unnecessary repairs and customer dissatisfaction. Dataset bias can lead to unfair outcomes, particularly affecting users of lesser-known phone models. Enhancing the dataset can help in mitigating this risk. A hybrid strategy that combines AI with human verification can be employed in automated decision-making and can improve reliability and accountability while ensuring fairness in outcomes.

5. Conclusion

This study shows the effectiveness of the CNN model in classifying cracked and uncracked phone screens with an accuracy of 92%. The findings suggest that the CNN-based model is competitive with other deep learning models in detecting smartphone defects because of its lightweight nature and seamless integration with a mobile application. This discovery has practical applications for phone repair shops and refurbished smartphone market, offering a swift, dependable, and automated method for identifying screen damage. The findings could reduce the deceptive resale of damaged devices, which can improve quality control in production and optimize repair procedures while ensuring transparency.

Regardless of the promising result, several constraints remain: the model had difficulty in evaluating rare or subtle crack patterns that are present in the input image, potential biases in predictions due to lack of diversity in training data, and computational constraints for real-time mobile implementation. This study establishes a foundation for AI-driven quality evaluation in the smartphone industry and beyond by addressing these problems and enhancing automated defect detection.

Future work seeks to focus on expanding the dataset to include several screen conditions, diverse phones, and better lighting conditions. In addition, incorporating transfer learning from a domain-specific dataset may improve accuracy. A significant improvement would be developing a severity classification system to categorize cracks based on the level of crack damage, along with a pricing algorithm to evaluate the market value of damaged smartphones. This aims to offer a transparent and unbiased valuation tool for consumers, enterprises, and stakeholders in the refurbished market.

Recommendations

The performance of the CNN model in detecting phone screen damage can be significantly improved using several strategies. Enhancing the training dataset by including several phone models, screen types, and damage patterns will enable the model to generalize more effectively. High-quality images are needed to obtain more precise model predictions. Images with standard conditions must be considered. Consistent lighting, minimal reflections, and high resolution can reduce the outcome of misclassification and minimize noise. In addition, exploring advanced deep learning can improve the efficiency of the model.

Finally, real-world testing and user feedback can also help in confirming the model's reliability and performance. This feedback results in iterative refinement of the model, which improves the experience for both buyers and sellers in the refurbished smartphone market.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in RaYDaR at https://yorksj.figshare.com/.

Author Contribution Statement

Ogechi Chiama: Conceptualization, Methodology, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Swathi Ganesan:** Supervision.

Reference

[1] Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., …, & Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, *10*(20), 2470. https://doi.org/10.3390/electronics10202470

[2] Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., …, & Socher, R. (2021). Deep learning-enabled medical computer vision. *npj Digital Medicine*, *4*(1), 5. https://doi.org/10.1038/s41746-020-00376-2

[3] Wu, M., & Chen, L. (2015). Image recognition based on deep learning. In *2015 Chinese Automation Congress*, 542–546. https://doi.org/10.1109/CAC.2015.7382560

[4] Luo, Y. (2022). Research on computer intelligent image recognition technology. In *Proceedings of the 7th International Conference on Cyber Security and Information Engineering*, 735–737. https://doi.org/10.1145/3558819.3565181

[5] Li, Y. (2022). Research and application of deep learning in image recognition. In *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications*, 994–999. https://doi.org/10.1109/ICPECA53709.2022.9718847

[6] Zhao, X. (2023). Research and application of deep learning in image recognition. *Journal of Physics: Conference Series*, *2425*(1), 012047. https://doi.org/10.1088/1742-6596/2425/1/012047

[7] Nordin, M. J., Xin, O. W., & Aziz, N. (2019). Food image recognition for price calculation using convolutional neural network. In *Proceedings of the 2019 3rd International Conference on Digital Signal Processing*, 80–85. https://doi.org/10.1145/3316551.3316557

[8] Wei, Y., Tran, S., Xu, S., Kang, B., & Springer, M. (2020). Deep learning for retail product recognition: Challenges and techniques. *Computational Intelligence and Neuroscience*, *2020*(1), 8875910. https://doi.org/10.1155/2020/8875910

[9] Chen, M., Chen, S., Wang, S., Cui, Y., & Chen, P. (2023). Accurate segmentation of small targets for LCD defects using deep convolutional neural networks. *Journal of the Society for Information Display*, *31*(1), 13–25. https://doi.org/10.1002/jsid.1185

[10] Ma, L., Lu, Y., Nan, X., Liu, Y., & Jiang, H. Q. (2017). Defect detection of mobile phone surface based on convolution neural

network. *DEStech Transactions on Computer Science and Engineering*. http://dx.doi.org/10.12783/dtcse/icmsie2017/18645

[11] Yu, D., & Yang, S. (2024). Defect detection of cell phone screen using a faster regional convolutional neural network with multi-head attention mechanism. *Journal of Electrical Systems*, *20*(2), 1707–1716. https://doi.org/10.52783/jes.1618

[12] Han, H., Yang, R., Li, S., Hu, R., & Li, X. (2023). SSGD: A smartphone screen glass dataset for defect detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1–5. https://doi.org/10.1109/ICASSP49357.2023.10096682

[13] Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, *99*, 52–58. https://doi.org/10.1016/j.autcon.2018.11.028

[14] Yoo, H.-S., & Kim, Y.-S. (2016). Development of a crack recognition algorithm from non-routed pavement images using artificial neural network and binary logistic regression. *KSCE Journal of Civil Engineering*, *20*(4), 1151–1162. https://doi.org/10.1007/s12205-015-1645-9

[15] Al-Azzawi, A., Ouadou, A., Max, H., Duan, Y., Tanner, J. J., & Cheng, J. (2020). DeepCryoPicker: Fully automated deep neural network for single protein particle picking in cryo-EM. *BMC Bioinformatics*, *21*(1), 509. https://doi.org/10.1186/s12859-020-03809-7

[16] Li, G., Zhang, M., Li, J., Lv, F., & Tong, G. (2021). Efficient densely connected convolutional neural networks. *Pattern Recognition*, *109*, 107610. https://doi.org/10.1016/j.patcog.2020.107610

[17] IBM. (n.d.). *What are convolutional neural networks*?. https://www.ibm.com/topics/convolutional-neural-networks

[18] Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *173*, 24–49. https://doi.org/10.1016/j.isprsjprs.2020.12.010

[19] Dung, C. V., Sekiya, H., Hirano, S., Okatani, T., & Miki, C. (2019). A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks. *Automation in Construction*, *102*, 217–229. https://doi.org/10.1016/j.autcon.2019.02.013

[20] Zhang, W., Ma, H., Li, X., Liu, X., Jiao, J., Zhang, P., …, & Cao, S. (2021). Imperfect wheat grain recognition combined with an attention mechanism and residual network. *Applied Sciences*, *11*(11), 5139. https://doi.org/10.3390/app11115139

[21] Bhatele, K. R., & Bhadauria, S. S. (2020). Classification of neurodegenerative diseases based on VGG 19 deep transfer learning architecture: A deep learning approach. *Bioscience Biotechnology Research Communications*, *13*(4), 1972–1980. http://dx.doi.org/10.21786/bbrc/13.4/51

[22] Tan, M., & Le, Q. (2021). EfficientNetV2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning*, *139*, 10096–10106.

[23] Skinner, C.-A. (2017). *What to do if you smash your phone screen*. https://www.goodhousekeeping.com/uk/consumer-advice/technology/a554890/what-to-do-if-you-smash-your-smartphones-screen

How to Cite: Chiama, O., & Ganesan, S. (2026). Automated Evaluation of Smartphone Screen Damage: A CNN-Based Image Recognition System. *Artificial Intelligence and Applications*, *4*(2), 215–223. https://doi.org/10.47852/bonviewAIA52024309