International Journal of GEOMATE, Aug., 2025 Vol.29, Issue 132, pp.110-123
ISSN: 2186-2982 (P), 2186-2990 (O), Japan, DOI: https://doi.org/10.21660/2025.132.4988
Geotechnique, Construction Materials and Environment

# DCNN-BASED MODEL TO PREDICT CONCRETE STRENGTH FROM MOBILE PHONE IMAGES USING MACHINE LEARNING

*Ronnel M. Quinto[1] and Gilford B. Estores[2]

[1]School of Graduate Studies, Mapua University, Philippines

[2]School of Civil, Environmental, and Geological Engineering, Mapua University, Philippines

*Corresponding Author, Received: 03 April 2025, Revised: 21 April 2025, Accepted: 28 April 2025

**ABSTRACT:** Concrete's compressive strength is critical for ensuring the safety, durability, and efficiency of construction projects. Traditional strength testing methods, though reliable, are labor-intensive, costly, and impractical for rapid, large-scale evaluations. This research explores a novel approach using Deep Convolutional Neural Networks (DCNN), specifically a ResNet50V2 architecture combined with transfer learning, to predict the compressive strength of concrete from images captured using mobile phones. A dataset of 49,000 dry concrete specimen images was prepared and enhanced through perceptual hashing for duplicate elimination, image preprocessing (resizing, normalization, standardization), and twenty-fold data augmentation, including random rotations, brightness/contrast adjustments, and horizontal flipping, to improve data diversity and model robustness. The DCNN model underwent a two-phase training process: initial feature extraction with frozen base layers, followed by fine-tuning of the final layers. The model accurately predicted compressive strengths of normal-strength concrete ranging from 12.96 MPa to 28.67 MPa. Evaluation metrics, including the coefficient of determination ($R^2$ = 0.9691), root mean square error (RMSE = 1.2349 MPa), and mean absolute percentage error (MAPE = 1.8651%), confirmed the model's high prediction accuracy. A paired t-test indicated no statistically significant difference (p = 0.9968) between true and predicted values, validating model reliability. An Android-based mobile application was developed for real-time, on-site predictions. A second paired t-test comparing outputs from the Python-based model and the mobile app yielded a p-value of 0.9148, confirming cross-platform consistency. This study presents a scalable, efficient, and highly accurate nondestructive method for concrete strength evaluation.

*Keywords: Deep Learning, Concrete Strength, Residual Network, Transfer Learning, Mobile Application*

## 1. INTRODUCTION

Concrete remains the most extensively utilized construction material globally, essential in developing infrastructure such as buildings, bridges, and roads [1]. Its widespread use is attributed to its durability, versatility, and cost-effectiveness. A critical property governing concrete's structural integrity and longevity is its compressive strength, typically assessed through traditional laboratory-based testing methods. While accurate, these physical tests require significant time and resources, posing challenges for rapid assessments or large-scale construction projects [2].

The compressive strength of concrete varies based on multiple parameters. These include the water-to-cement (w/c) ratio, aggregate gradation, cement content, pore structure, and material uniformity. Researchers such as Lange et al. [3] and Başyiğit et al. [4] demonstrated that surface texture characteristics—such as pore distribution, aggregate orientation, and cement paste composition—can visually reflect internal structural integrity. Image-based analysis has emerged as a potential nondestructive method to capture such features and estimate strength.

However, early image processing (IP) methods required manual segmentation and thresholding, introducing subjectivity and limited adaptability to surface heterogeneity and varying lighting conditions. Dogan et al. [5] introduced a hybrid approach combining Artificial Neural Networks (ANN) with IP to predict compressive strength using surface images. Their method yielded prediction accuracies of up to 99.87%. Similarly, Waris et al. [6] used ANN and DSLR images to estimate the strength of concrete with fly ash and silica fumes, achieving a high correlation (R = 0.9865) between predicted and actual values. These studies highlighted the potential of AI for strength prediction but also underscored the challenges of handling high-dimensional image data.

Researchers have adopted Deep Convolutional Neural Networks (DCNNs) to address these limitations, which automate feature extraction and excel in learning spatial hierarchies from images. Shin et al. [7] evaluated DCNN architectures such as AlexNet, GoogLeNet, and ResNet in predicting compressive strength, reporting an RMSE of 3.56 MPa with the ResNet-based ConcNet_R. Lee et al. [8] found that high-resolution DSLR images produced better predictions (RMSE = 3.779 MPa)

than microscope images. Cuasay et al. [9] demonstrated the feasibility of a region-based DCNN for structural damage detection in reinforced concrete, reinforcing the broader applicability of deep learning in construction-related image analysis.

Hybrid ensemble models have been explored to improve generalization and mitigate overfitting. Mayya et al. [10] combined multiple CNN architectures and transfer learning models through ensemble methods such as stacking and boosting, resulting in improved prediction accuracy for crack detection.

Despite these advancements, most existing studies rely on high-end cameras, microscopes, or detailed mix design data, which are impractical in typical field environments. Addressing these limitations, this research proposes a DCNN model employing Residual Networks (ResNet) combined with transfer learning techniques to predict the compressive strength of normal-strength concrete from images captured using mobile phone cameras. This research aims to develop a practical, cost-effective tool that delivers rapid and reliable strength assessments on construction sites.

## 2. RESEARCH SIGNIFICANCE

The significance of this research aligns directly with the United Nations Sustainable Development Goals (SDGs), specifically Goal 9 (Industry, Innovation, and Infrastructure), Goal 11 (Sustainable Cities and Communities), and Goal 12 (Responsible Consumption and Production). By utilizing smartphone imaging and advanced deep learning techniques, the study promotes innovation in construction practices, enhances the sustainability and safety of urban infrastructures, and promotes responsible resource management. Through quicker, more accurate concrete strength predictions, this approach minimizes material wastage, optimizes resource efficiency, and supports environmentally conscious construction, contributing to sustainable urban development and resilient infrastructure systems.

## 3. METHODS

### 3.1 Data Collection

*3.1.1 Material Preparation*

The material preparation significantly influenced the quality of concrete and the accuracy of collected data for machine learning [11]. Portland Cement Type 1 was selected for its consistency and versatility, directly affecting the strength and durability of the concrete. Coarse aggregates of 19mm (3/4 inch) provided structural integrity, with a rodded density of 1568 kg/m³, an absorption rate of 0.36%, and a moisture content of 0.31%. Vibro sand, used as the fine aggregate, enhanced workability and had a fineness modulus of 2.69, an absorption rate of 2.88%, and a moisture content of 1.58%.

*3.1.2 Mix Design*

Twenty concrete specimens were prepared for each of four distinct mix designs, yielding 80 cylindrical samples. Each mix was designed to achieve a target compressive strength ranging from 11 MPa to 42 MPa. The water-to-cement (w/c) ratio was systematically varied to observe its effect on strength development. Mix proportions were determined using the absolute volume method, following established mix design guidelines and material characterization results.

All materials—including water, cement, fine aggregate, and coarse aggregate were weighed per cubic meter (kg/m³) of concrete. The slump range was maintained at 70–100 mm across all mixes to ensure consistent workability during specimen casting. The fine and coarse aggregates were adjusted based on their respective moisture content and absorption rates to ensure accurate, effective water content and mix consistency. The detailed proportions for each mix are presented in Table 1, which reflects the batch composition per cubic meter of concrete.

Table 1. The Mix Proportion of Experimental Concrete Specimens

| W/C (%) | Water (kg/m$^3$) | Cement (kg/m$^3$) | Fine Aggregate (kg/m$^3$) | Coarse Aggregate (kg/m$^3$) | Slump (mm) |
|---|---|---|---|---|---|
| 0.66 | 216.2 | 327.48 | 836.12 | 992.48 | 70-100 |
| 0.60 | 215.76 | 360.92 | 802.15 | 992.48 | 70-100 |
| 0.44 | 214.11 | 488.1 | 672.96 | 992.48 | 70-100 |
| 0.36 | 212.84 | 585.71 | 573.81 | 992.48 | 70-100 |

*3.1.3 Specimen Images and Videos*

High-resolution images and videos of concrete specimens were captured using the POCO X6 5G smartphone with a 64 MP OmniVision OV64B sensor (1/2" size, 0.7µm pixels) with a 25mm f/1.8 aperture lens and Optical Image Stabilization (OIS). The specimens were photographed within a Pxel LB80LED Studio Soft Box LED Light Tent (800 × 800 × 800 mm), as shown in Fig. 1, ensuring consistent illumination with 12000 LUX brightness, 5500K color temperature, and a color rendering index of 85. This controlled lighting environment ensured accurate color representation and uniform exposure, essential for reliable image-based analysis and compressive strength prediction [12].

Fig.1 Image Capture Setup Using Pxel LB80LED Studio Soft Box LED Light Tent

The specimens were placed on a flat base, as shown in Fig. 1, with a plain, uniform white background to avoid distractions or reflections. For each specimen, images were captured from the top and bottom surfaces, as shown in Fig. 2, ensuring a comprehensive view of the critical areas likely to provide significant information about the concrete's compressive strength.

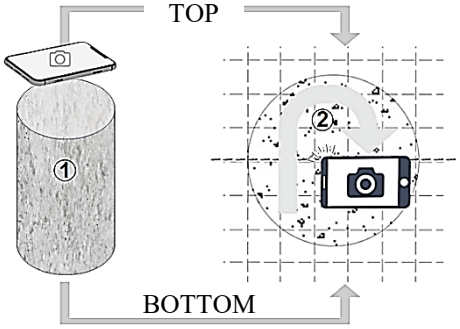


Fig.2 Specimen Images and Videos Capturing Process

The POCO X6 5G mobile camera was positioned at a fixed distance of 7 cm from each specimen's surface to ensure detailed, consistently scaled, and focused images, as shown in Fig. 3. Default camera settings were employed to maintain uniform image conditions and eliminate variability resulting from manual adjustments. In addition to still images, videos of dry concrete specimens were recorded to comprehensively capture surface details from multiple angles. Individual frames were extracted from these videos, complementing the still images and capturing subtle variations or features potentially missed during still photography. Multiple captures from each angle ensured the selection of the highest quality images, minimizing potential blurriness or focus issues. This method generated a robust dataset of thousands of high-quality images, which is critical for accurately predicting compressive strength using the DCNN model.

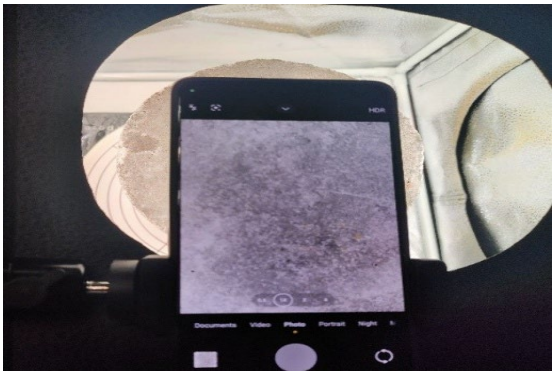

Fig.3 Image and Video Capturing Setup

### 3.2 Data Preprocessing

Images were initially extracted from video recordings at one frame per second, ensuring dataset diversity without redundancy. Duplicate images were identified and eliminated using perceptual hashing (pHash) with an 8-bit hash and zero thresholds for exact matches. To balance the dataset, 700 unique images per compressive strength value were selected, totaling 49,000 images across 70 distinct strength levels, thus preventing bias due to overrepresentation. All images were resized to standardized dimensions of 224 × 224 pixels to meet the input requirements of the DCNN model, maintaining consistency and computational efficiency. Pixel values were normalized to a 0 to 1 range and subsequently standardized to achieve a mean of 0 and standard deviation of 1, ensuring stable and efficient model training.

Furthermore, each image underwent twenty augmentations, as shown in Fig. 4, including horizontal flipping, random brightness adjustments (±10%), contrast changes (range 0.9 to 1.1), and random rotations of 0°, 90°, 180°, or 270°. These augmentations significantly expanded the dataset, improving the model's robustness and generalization.

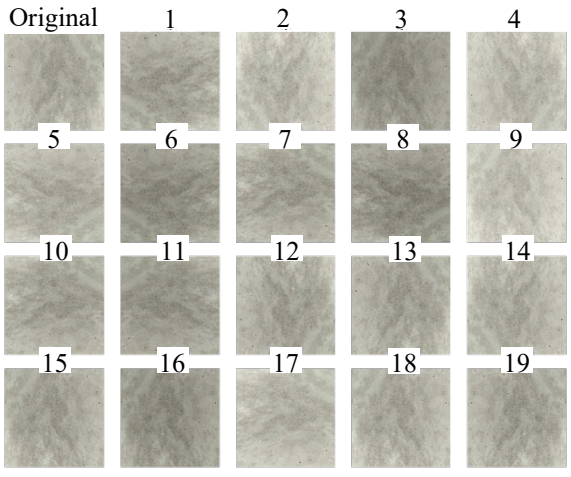


Fig.4 Sample Image Augmentation

### 3.3 Model Development

The DCNN model was developed using Jupyter Notebook. Python was the primary programming language used due to its extensive deep-learning libraries, notably TensorFlow and Keras [13]. TensorFlow provides a flexible framework for computational graph execution, while Keras simplifies model construction, training, and fine-tuning [14]. The ResNet50V2 architecture, pre-trained on ImageNet, was employed as the foundational model, leveraging transfer learning to enhance training efficiency and model performance [15]. Initially, base layers of ResNet50V2 were frozen to retain general features. Subsequently, specific layers were unfrozen for fine-tuning to enhance the network's ability to identify concrete-specific features.

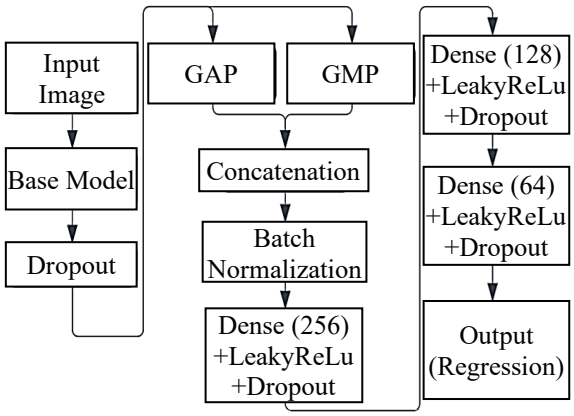


Fig.5 Schematic Diagram of the Deep Learning Model using Resnet50V2

As shown in Fig. 5, the custom model architecture included standardized inputs of 224 × 224 pixels in RGB channels. Global Average Pooling (GAP) and Global Max Pooling (GMP) layers condensed feature maps from ResNet50V2, capturing spatial details and reducing overfitting. GAP computed the mean, and GMP captured the maximum values of the feature maps, resulting in a combined rich feature representation for the model. Batch normalization, dropout layers (30% and 50%), and dense layers (256, 128, 64 neurons with LeakyReLU activation and L2 regularization) ensured stable training and robust feature extraction. The final dense layer utilized linear activation for regression, predicting compressive strength values.

### 3.4 Training of DCNN Model

The dataset was systematically split into training (70%), validation (10%), and testing (20%) subsets, ensuring balanced representation across all compressive strength categories. Training images underwent extensive augmentation, generating 686,000 images, significantly enhancing model robustness and generalization. The validation and testing subsets underwent minimal augmentation to maintain evaluation integrity, as shown in Table 2.

Table 2. Datasets Composition

| Data Sets | Preprocess Images | Augmented Images (20 Augmentations) | Total |
|---|---|---|---|
| Training (70%) | 34,300 | 686000 | |
| Validation (10%) | 4,900 | - | |
| Testing (20%) | 9800 | - | |
| Total | 49000 | 686000 | 735000 |

The training phase comprised two stages: initial training and fine-tuning. In the initial training stage, the ResNet50V2 base layers were frozen to preserve pre-trained feature extraction capabilities, training only the newly added dense layers for 20 epochs. During the fine-tuning stage, the last 50 layers of ResNet50V2 were unfrozen and trained for 40 epochs with a reduced learning rate, allowing deeper feature adaptation specific to concrete compressive strength prediction.

Hyperparameters were systematically optimized, including learning rates, batch sizes, dropout rates, and epochs. The Adam optimizer and the Huber loss function facilitated efficient training and minimized sensitivity to outliers. The final model employed batch normalization, dropout layers (30% initially and 50% subsequently), and callbacks such as ReduceLROnPlateau, EarlyStopping, and ModelCheckpoint to dynamically manage training and enhance model performance.

### 3.5 Model Validation and Evaluation

The proposed model's performance was assessed using established statistical metrics to validate its reliability and effectiveness for predicting concrete compressive strength. The evaluation involved calculating the coefficient of determination ($R^2$), Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

The Huber loss function was implemented throughout model training and validation to ensure robustness against outliers. For minor errors, it operated similarly to MSE to provide sensitivity, transitioning to MAE behavior for more significant errors to minimize the impact of extreme values.

Continuous monitoring of Huber loss and MAE was conducted during validation to detect potential overfitting or underfitting. Consistently low validation metrics suggested strong generalization capability, whereas increasing validation loss despite declining training loss indicated overfitting. Corrective measures, such as incorporating dropout

and batch normalization techniques, were employed to enhance generalization.

### 3.6 Testing of DCNN Model

*3.6.1 Initial Testing*

In the initial testing phase, the developed Deep Convolutional Neural Network (DCNN) model, utilizing ResNet50V2 and transfer learning, was evaluated using a test dataset constituting 20% of the total dataset. This subset remained completely independent of the training and validation datasets to objectively assess the model's generalization capabilities. All test images underwent standardized preprocessing, including resizing, normalization, and standardization, ensuring consistency in evaluation. The model's predictive performance was evaluated using Mean Absolute Error (MAE), Coefficient of Determination ($R^2$), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), collectively providing a thorough assessment of accuracy and predictive reliability.

*3.6.2 Final Testing*

A final testing phase was conducted using a distinct dataset composed of ten entirely new compressive strength categories unseen during training, validation, and initial testing to validate the model's robustness and real-world applicability. Images were preprocessed following identical steps used during initial testing.

The DCNN model generated predictions for each image, which were compared against their compressive strength values. The accuracy was evaluated by averaging predictions within each category. Performance trends were critically analyzed to ensure the model's accuracy remained consistent across varying compressive strength levels. Results from this comprehensive evaluation informed potential refinements in model architecture, hyperparameters, and data augmentation strategies, aiming to maximize model reliability and predictive accuracy in practical, real-world applications.

### 3.7 Android-based Prototype Application Development

A prototype Android-based mobile application was developed to integrate the trained Deep Convolutional Neural Network (DCNN) model. This application lets construction professionals obtain real-time predictions of concrete compressive strength directly from images captured on mobile devices.

The trained DCNN model was adapted for deployment on Android devices using TensorFlow Lite (TFLite), a framework optimized for mobile and embedded systems. This conversion involved exporting the model trained with TensorFlow/Keras, applying quantization techniques to optimize its size and computational efficiency, and converting it into the TFLite format suitable for mobile devices [16].

*3.7.1 Development Environment*

Android Studio was chosen as the development environment due to its extensive capabilities, compatibility with TensorFlow Lite, and robust features, enabling seamless integration of the DCNN model within the mobile application [17].

*3.7.2 User Interface (UI)*

The application's user interface (UI) was carefully designed for clarity and ease of use. As shown in Fig. 6, users were presented with intuitive options to capture or upload images of concrete specimens directly from the application. A built-in image preview feature allowed users to confirm image quality before initiating prediction.

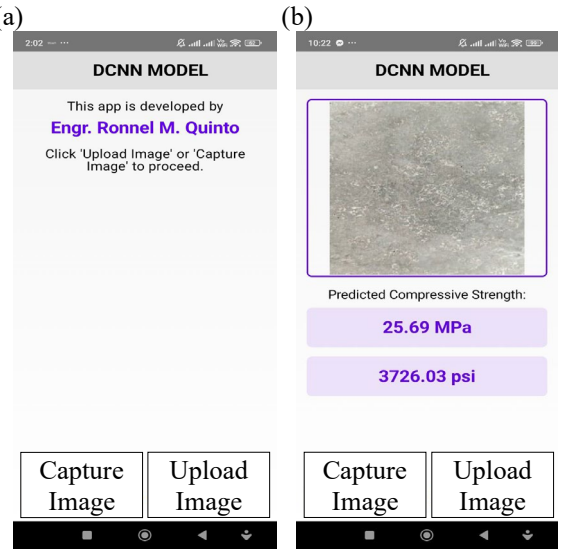


Fig.6 User Interface of Android-based Application (a) Image Input Interface (b) Prediction Interface

Upon selection, the images underwent preprocessing identical to the training procedures. The TFLite model processed these images to generate predictions, displaying results in megapascals (MPa) and pounds per square inch (psi).

### 3.8 Statistical Analysis of True Values vs Predicted Values

A statistical evaluation using a Paired T-test was conducted to assess the accuracy and reliability of the developed Deep Convolutional Neural Network (DCNN) model and its integration within the Android-based application in predicting concrete compressive strength. Two primary comparisons were performed. The first compared actual experimental compressive strength values with

DCNN model-predicted values. The second compared DCNN model predictions obtained via Python against predictions from the Android-based application using TensorFlow Lite. For each comparison, a significance level (α) of 0.05 was established. The paired t-test assumes that the differences between paired observations are approximately normally distributed and that the variances between groups are equal. These assumptions were evaluated using descriptive statistics, distribution symmetry, and sample size considerations to ensure the appropriateness and validity of the test, thereby strengthening the reliability of the comparative analysis across different prediction platforms and conditions.

## 4. RESULTS AND DISCUSSION

### 4.1 Datasets and Preprocessing

The dataset consisted of images of cylindrical concrete specimens, with a total of 49,000 preprocessed images. The training set initially contained 34,300 images, which, after augmentation, expanded to 686,000 images, significantly enhancing the model's generalization ability. The validation set comprised 4,900 images, each undergoing a single augmentation to maintain dataset consistency. The testing set, containing 9,800 images, remained unchanged to provide an unbiased evaluation of the model's real-world applicability.

### 4.2 Model Optimization

The model was built using the ResNet50V2 architecture, pre-trained on the ImageNet dataset, and adapted for regression-based compressive strength prediction. The training process was refined through multiple training runs, allowing for iterative improvements in hyperparameter selection and model structure. The best-performing approach involved a two-phase training process:

Initial Training Phase: The base layers of ResNet50V2 were frozen, meaning only the newly added fully connected layers were trained. This approach leveraged pre-trained feature extraction, allowing the model to adapt to the concrete compressive strength prediction task.

Fine-Tuning Phase: Through extensive experimentation, it was determined that unfreezing the last 50 layers was optimal for retaining learned features and adapting to the dataset's unique characteristics. Unfreezing too few layers limited the model's ability to capture task-specific details, while unfreezing too many layers led to overfitting. The learning rate was reduced to $1e^{-5}$ to ensure that the fine-tuning process adjusted weights gradually without disrupting the previously learned representations.

Table 3. Optimal Hyperparameter Settings for DCNN Model

| Hyperparameter | Optimal Value |
|---|---|
| Model Architecture | ResNet50V2 (pre-trained) |
| Input Image Size | 224 × 224 pixels |
| Batch Size | 32 |
| Initial Training Epochs | 20 |
| Fine-Tuning Epochs | 40 |
| Initial Learning Rate | $1e^{-4}$ |
| Fine-Tuning Learning Rate | $1e^{-5}$ |
| Dropout Rate (Initial Layers) | 30% |
| Dropout Rate (Dense Layers) | 50% |
| Regularization (L2) | 0.01 |
| Optimizer | Adam |
| Loss Function | Huber Loss |
| Activation Function (Dense) | LeakyReLU (α=0.1) |
| Train/Val/Test Split | 70% / 10% / 20% |
| Augmentations per Image | 20 |
| Total Training Images | 686,000 |

Hyperparameter tuning was a crucial aspect of model optimization. The Adam optimizer was used with an initial learning rate of $1e^{-4}$, which was dynamically reduced using ReduceLROnPlateau to prevent overfitting during convergence. Dropout layers (0.3 after the base model and 0.5 after fully connected layers) were introduced to improve model stability and prevent overfitting. Training performance was monitored using EarlyStopping, which halted training if validation loss did not improve for five consecutive epochs, and ModelCheckpoint, which ensured that the best-performing model was saved. The selection of learning rate and dropout values was determined based on extensive experimentation, with multiple training iterations guiding the identification of the most effective configuration.

### 4.3 Training and Validation

The performance of the developed Deep Convolutional Neural Network (DCNN) model was systematically assessed by analyzing the training and validation loss values recorded over successive training iterations. At the beginning of training, both the training and validation losses exhibited elevated values, as shown in Fig. 7, signifying the model's initial lack of proficiency in capturing relevant data patterns. As training continued, both loss values demonstrated a stable decline, indicating effective learning and improved recognition of critical features necessary for accurately predicting compressive strength from concrete specimen images. The sharpest decrease in loss values occurred in the early stages, highlighting rapid model adaptation and feature extraction.

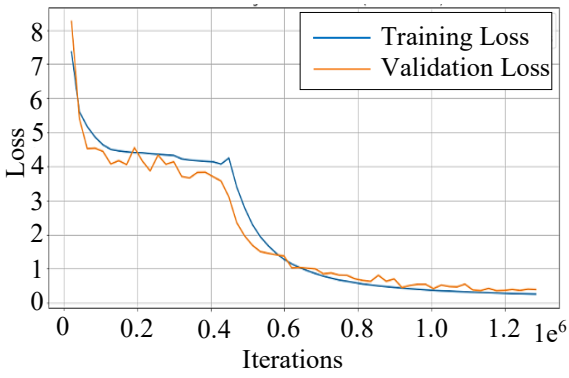


Fig.7 Loss Curve Graph

As training continued towards convergence, the reduction rate of loss values diminished, suggesting incremental refinements and optimization of model predictions. Although minor fluctuations in validation loss were observed, likely attributable to inherent variations within the validation dataset, the overall trend consistently moved toward stability. The close alignment and parallel decrease of validation loss relative to training loss further suggested the successful avoidance of overfitting.

### 4.4 Performance Metrics

The deep convolutional neural network (DCNN) model was evaluated using key performance metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coefficient of Determination ($R^2$), and Mean Absolute Percentage Error (MAPE), as summarized in Table 4.

Table 4. Performance Metrics Result

| Metric | Value |
|---|---|
| Best Training Loss | 0.2539 |
| Best Training MAE | 0.4228 |
| Best Validation Loss | 0.3521 |
| Best Validation MAE | 0.4708 |
| Validation RMSE | 1.1753 |
| Test RMSE | 1.2349 |
| Coefficient of Determination ($R^2$) | 0.9691 |
| Mean Absolute Percentage Error (MAPE) | 1.8651 |

*4.4.1 Root Mean Square Error*

The validation RMSE was recorded at 1.1753, while the test RMSE was computed at 1.2349, indicating that the model maintained a low prediction error across different datasets.

The slight difference between the validation RMSE and the test RMSE (1.2349) indicated that the model performed consistently on unseen data without significant overfitting.

*4.4.2 Coefficient of Determination ($R^2$)*

The $R^2$ value was recorded at 0.9691, indicating that the model accurately predicted 96.91% of the variation in compressive strength. This high $R^2$ value confirmed that the model successfully captured the underlying patterns in the dataset and exhibited substantial predictive accuracy.

*4.4.3 Mean Absolute Percentage Error*

The Mean Absolute Percentage Error (MAPE) for the validation dataset was recorded at 1.8651%, indicating that, on average, the model's predictions deviated by approximately 1.87% from the actual compressive strength values. The low MAPE value suggested that the deep convolutional neural network (DCNN) model consistently produced accurate predictions with minimal deviation from actual values.

### 4.5 Predicted vs True Values

As shown in Fig. 8, most data points were closely clustered along the perfect prediction line, suggesting that the model's predictions aligned well with the actual compressive strength values. This indicated that the model effectively learned the underlying patterns from the dataset, allowing it to produce accurate predictions of compressive strength based on concrete images.

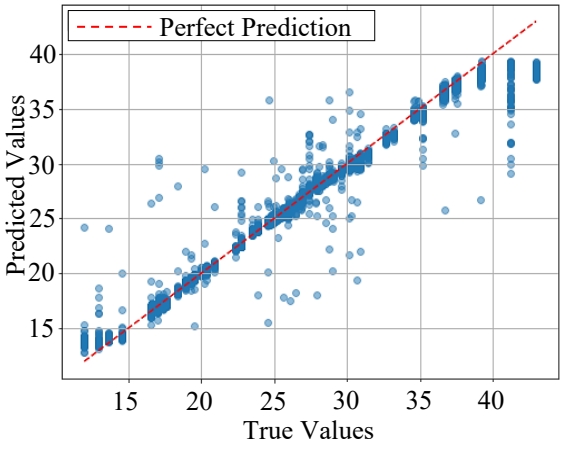


Fig.8 Predicted vs True Values Scatter Plot

However, minor deviations were observed, particularly in the mid-range compressive strength values (20 MPa to 35 MPa), where some points were scattered away from the ideal line. These deviations suggested the presence of minor errors, possibly due to variations in image quality, dataset inconsistencies, or limitations in the model's generalization ability. At lower strength values predictions were well-distributed around the perfect prediction line, showing that the model performed reliably in this range. Similarly, predictions aligned

with actual values at higher compressive strength values (above 35 MPa). However, some clustering was detected, indicating a potential saturation effect where the model's predictions were slightly constrained within a specific range. A few outliers were observed, particularly in the mid-to-high strength range, indicating instances where the model's predictions deviated more significantly from the actual values. However, these errors remained within an acceptable margin, suggesting the model retained strong predictive capabilities.

### 4.6 Residuals

*4.6.1 Histogram*

As shown in Fig. 9, the residual histogram demonstrated that most prediction errors were tightly clustered around zero, indicating that the DCNN model generated compressive strength predictions with minimal deviation from actual values. The sharp central peak reflected high model accuracy, while the symmetrical, narrow spread of residuals suggested an absence of systemic bias. Most residuals fell within ±5 MPa, with only a few outliers exceeding ±10 MPa. Dataset inconsistencies, image quality issues, or anomalous concrete samples likely caused these rare errors. The residual distribution confirmed the model's robust predictive performance across the dataset.

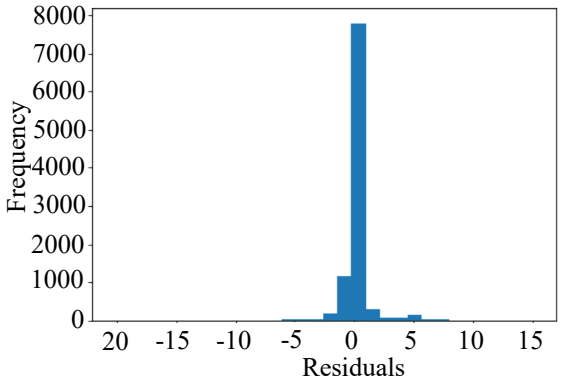


Fig.9 Residual Distribution

*4.6.2 Plot*

As shown in Fig. 10, the residual plot illustrated the prediction errors by plotting residual differences between actual and predicted compressive strengths against predicted values. Residuals were predominantly clustered around the zero line, suggesting high prediction accuracy and minimal systematic bias. The symmetrical spread confirmed the absence of overestimation or underestimation tendencies. However, a widening dispersion was observed for predictions above 30 MPa, indicating increased variance and reduced accuracy at higher strength levels. Notably, residuals below 20 MPa remained small, while higher strength values tended to be underestimated, as reflected by more significant negative residuals. A few outliers exceeded ±15 MPa. The presence of heteroscedasticity highlighted slight performance inconsistencies of the DCNN model at elevated strength ranges.

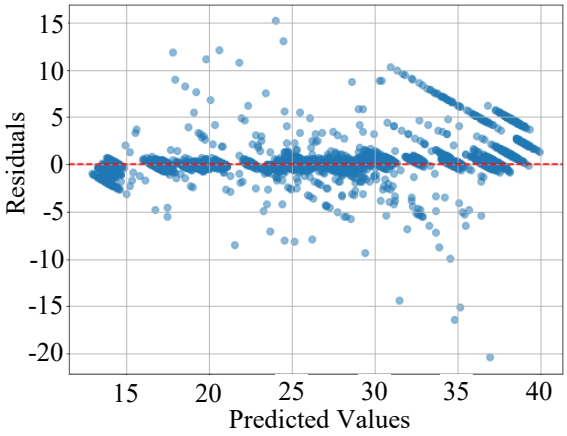


Fig.10 Residual Plot

### 4.7 Final Testing on Unseen Datasets

The model was tested on ten (10) unique datasets containing concrete specimens with varying compressive strengths that had never been included in training, validation, or initial testing.

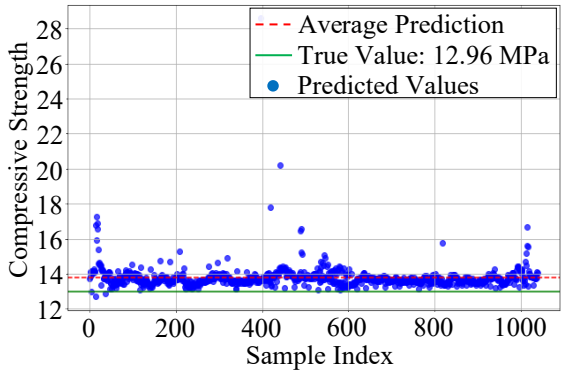


Fig.11 Predicted Values per Image in 12.96 MPa

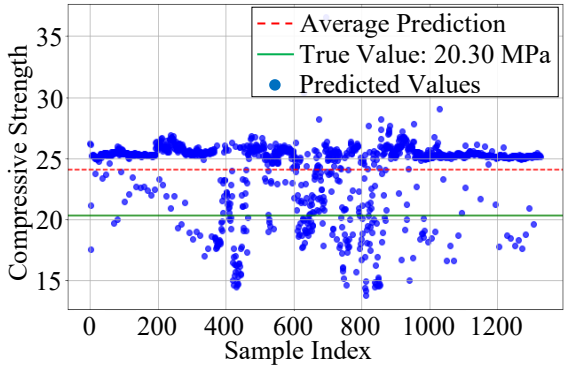


Fig.12 Predicted Values per Image in 20.30 MPa

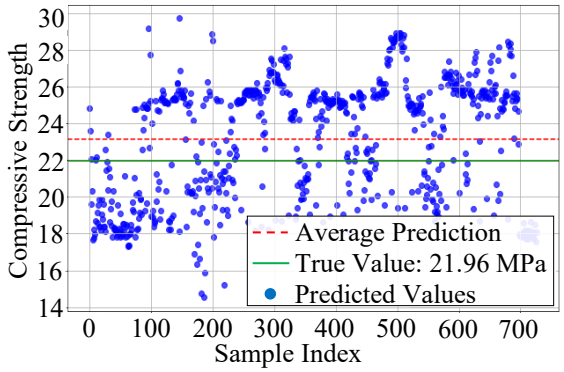


Fig.13 Predicted Values per Image in 21.96 MPa

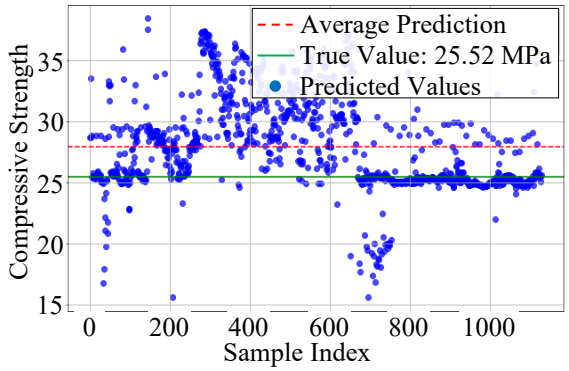


Fig.14 Predicted Values per Image in 25.52 MPa

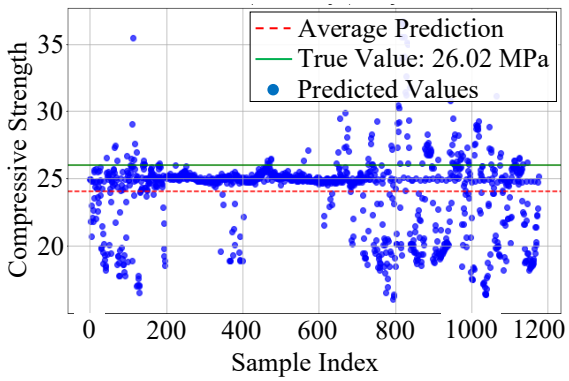


Fig.15 Predicted Values per Image in 26.02 MPa

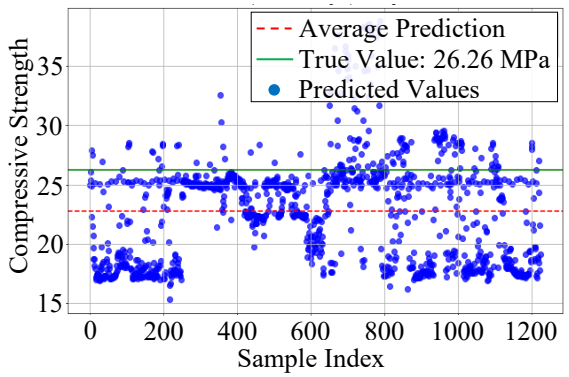


Fig.16 Predicted Values per Image in 26.26 MPa

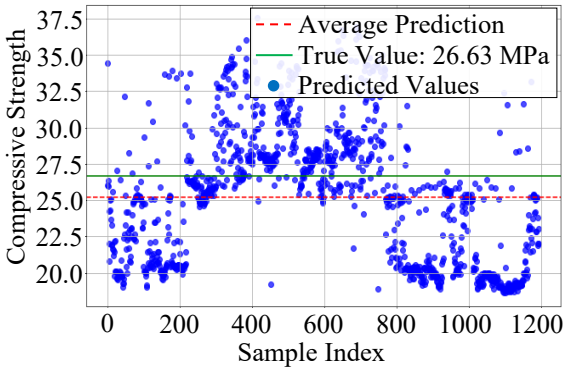


Fig.17 Predicted Values per Image in 26.63 MPa

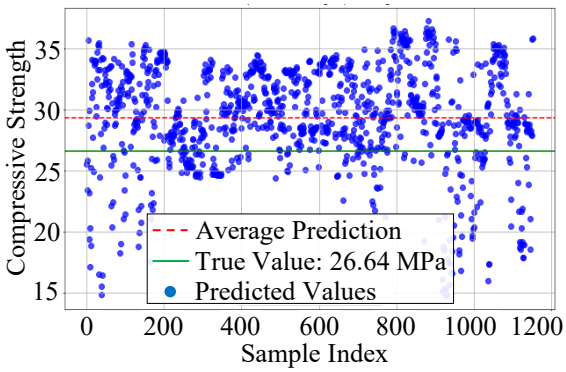


Fig.18 Predicted Values per Image in 26.64 MPa

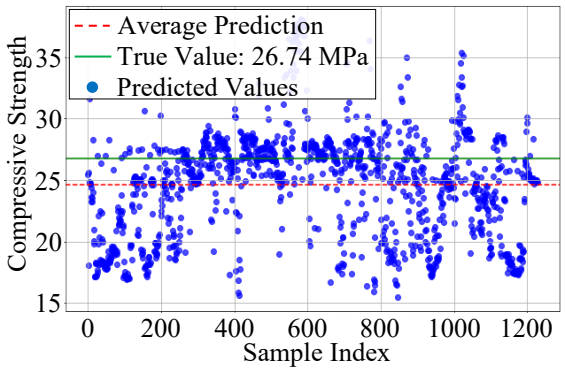


Fig.19 Predicted Values per Image in 26.74 MPa

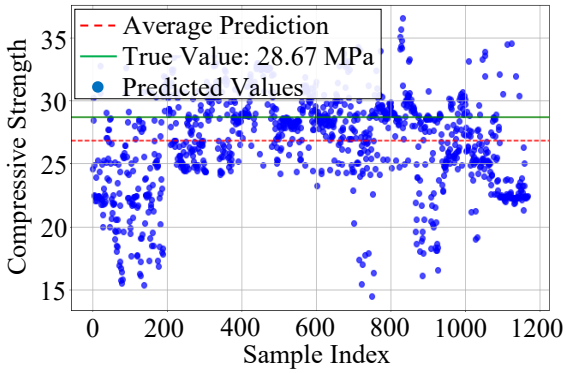


Fig.20 Predicted Values per Image in 28.67 MPa

The model exhibited both overestimations and underestimations across different compressive strength levels. The model consistently overestimated lower compressive strengths, such as 12.96 MPa predicted as 13.78 MPa (Fig. 11) and 20.30 MPa predicted as 24.08 MPa (Fig. 12). Predictions tended to cluster slightly above the actual values, indicating a bias toward higher strength estimates in this range.

At moderate compressive strengths, the model showed a tendency to underestimate values, such as 26.02 MPa predicted as 24.06 MPa (Fig. 15) and 26.26 MPa predicted as 22.76 MPa (Fig. 16). This pattern suggests that the model may have misinterpreted certain structural features in mid-range strength concrete specimens.

Predictions fluctuated between overestimation and underestimation in datasets with strengths above 26 MPa. For example, 26.64 MPa were overestimated at 29.36 MPa (Fig. 18). In comparison, 28.67 MPa were underestimated at 26.82 MPa (Fig. 20). This suggests that the model handled higher-strength specimens less consistently, possibly due to variations in texture, aggregate distribution, or lighting conditions.

These observations indicate that while the model maintained strong predictive accuracy, specific strength ranges introduced subtle biases in estimation. To quantify the model's accuracy, the Root Mean Square Error (RMSE) was calculated at 2.35, reflecting the average deviation between predicted and actual values. The relatively low RMSE suggests minimal significant errors, reinforcing the model's generalization ability to unseen data.

### 4.8 Statistical Analysis of True Values and Predicted Values

*4.8.1 True Values vs DCNN Predicted Values*

To evaluate the accuracy of the DCNN model, a paired t-test was conducted comparing ten actual compressive strength values to their corresponding predictions from the Python-based model. The test resulted in a mean difference of -0.008 MPa, a t-statistic of 0.004, and a p-value of 0.9968, far above the significance level of 0.05. These results indicate no statistically significant difference between the true values and the model's predictions, as shown in Table 5.

The assumption of normality of differences was evaluated using descriptive statistics and distribution symmetry. Although formal normality testing was not applied due to the small sample size (n = 10), visual inspection of the data revealed no strong deviation from normality. The test also assumed equal variances, which was considered reasonable given the similarity in standard deviations between true and predicted values.

Table 5. Paired T-Test Result (True Values and Python Model Predicted Values)

| Metric | Value |
|---|---|
| Mean Difference | -0.008 |
| Sum of Squared Differences | 19.7004 |
| T Statistic | 0.004 |
| Degrees of Freedom | 18 |
| Critical Value | 2.1009 |
| p-value | 0.9968 |
| Remark | $H_0$ |

*$H_o$ = There is no significant difference between the true compressive strength values and the predicted values.*
*$H_a$ = There is a significant difference between the true and predicted compressive strength values.*

These findings confirm that the Python-based DCNN model provides highly accurate and unbiased predictions, supporting its reliability as a nondestructive tool for estimating concrete compressive strength.

*4.8.2 Python Model Predicted Values vs Android-based Application Predicted Values*

A second paired t-test was performed to compare the predicted compressive strength values generated by the Python-based model and the Android-based mobile application. The dataset consisted of 50 paired observations, and the test produced a mean difference of -0.109 MPa, a t-statistic of 0.1072, and a p-value of 0.9148, as presented in Table 6. This result indicates no statistically significant difference between the predictions from the two platforms.

Given the larger sample size (n = 50), the normality of differences was assumed based on the central limit theorem. The assumption of equal variances was supported by the similarity of standard deviations between the two sets (Python: 5.0994 MPa, Android: 5.0656 MPa).

Table 6. Paired T-Test Result (True Values and Python Model Predicted Values)

| Metric | Value |
|---|---|
| Mean Difference | -0.109 |
| Sum of Squared Differences | 25.8319 |
| T Statistic | 0.1072 |
| Degrees of Freedom | 98 |
| Critical Value | 1.9845 |
| p-value | 0.9148 |
| Remark | $H_0$ |

*$H_o$ = There is no significant difference between the true compressive strength values and the predicted values.*
*$H_a$ = There is a significant difference between the true and predicted compressive strength values.*

These results validate that converting the trained model into TensorFlow Lite (TFLite) for mobile deployment did not introduce meaningful discrepancies, confirming the model's cross-platform consistency and field applicability.

### 4.9 Real-Time Prediction Performance

The performance evaluation of the optimized TensorFlow Lite (TFLite) model deployed within the Android application demonstrated highly efficient inference capabilities suitable for practical on-site usage. The final TFLite model was reduced to 93.8 MB, effectively balancing compactness and accuracy, allowing the application to operate smoothly on standard Android mobile devices. Inference speed was also optimized, with the model achieving an average prediction time of 34 milliseconds per image on a typical mid-range Android device. This ensures that prediction results are generated almost instantaneously, supporting real-time decision-making in field conditions.

### 4.10 Visual Reliability and Environmental Considerations

The Android-based model is optimized for rapid and lightweight performance; however, image-based misinterpretation can occur during prediction. Such issues may arise from non-uniform lighting, shadows, surface dust or debris, image blur, or improper camera angles, which can obscure or distort critical surface features such as aggregate texture, pore structure, or microcracks. These inconsistencies can affect the model's ability to interpret input images and predict compressive strength accurately.

To minimize these risks, the image acquisition process was standardized. A fixed working distance of 7 cm, a uniform white background, and controlled illumination using a softbox light (12,000 LUX, 5500K) were employed to ensure consistent and high-quality images across the dataset.

In addition to visual variability, environmental factors such as temperature, humidity, and concrete

Table 7. Comparison with existing NDT prediction results

| Author | Methods | Algorithm | Input Data | RMSE |
|---|---|---|---|---|
| Chen et al. [18] | NDT | Regression | Rebound hammer test value | 4.9 |
| Chen, Fu, Yao, et al., 2017 [19] | NDT | Regression | Ultrasonic pulse velocity test value | 4.9 |
| Rashid et al., 2017 [20] | NDT | Regression | Rebound hammer test value | 12.4 |
| | | | Ultrasonic pulse velocity test value | 37.9 |
| Yaseen et al., 2018 [21] | Machine learning | SVM | Concrete cement | 4.35 |
| | | | Concrete oven dry density | 3.57 |
| | | | Concrete water/binder | 4 |
| | | | Concrete foam | 3.33 |
| | Machine learning | M5 Tree | Concrete cement | 3.3 |
| | | | Concrete oven dry density | 2.17 |
| | | | Concrete water/binder | 6.84 |
| | | | Concrete foam | 2.29 |
| | Machine learning | MARS | Concrete cement | 3.68 |
| | | | Concrete oven dry density | 1.37 |
| | | | Concrete water/binder | 6.41 |
| | | | Concrete foam | 3.57 |
| | Machine learning | ELM | Concrete cement | 3.25 |
| | | | Concrete oven dry density | 1.06 |
| | | | Concrete water/binder | 3.8 |
| | | | Concrete foam | 1.48 |
| Jang et al., 2019 [22] | DCNN | ResNet | Concrete Microscope Image | 4.463 |
| Shin et al., 2019 [7] | DCNN | Concnet_A | Concrete Surface Digital Image | 3.82 |
| | | Concnet_G | Concrete Surface Digital Image | 3.64 |
| | | Concnet_R | Concrete Surface Digital Image | 3.56 |
| Lee et al., 2020 [8] | DCNN | DSLR | Concrete Surface DSLR Image | 3.779 |
| | | Microscope | Concrete Surface Microscope Image | 4.875 |
| Yu et al., 2024 [23] | DCNN | CNN | Pervious Concrete Surface Image | 2.72 |
| Proposed Model | DCNN/Transfer Learning | Resnet50V2 | Concrete Surface Mobile Image | 1.23 |

curing conditions can influence the surface appearance of the concrete and its true compressive strength. These factors may cause visible differences in surface texture or coloration and can impact the development of material properties over time. As such, they represent critical considerations when evaluating and interpreting image-based predictions in practical applications.

Given its demonstrated accuracy, speed, and portability, the proposed model offers strong potential for on-site concrete strength monitoring, particularly in remote or resource-limited environments. It is also well-suited for integration into mobile inspection tools, automated quality control workflows, and smart construction systems or IoT-based infrastructure platforms that require fast, nondestructive assessment of material performance.

### 4.11 Existing NDT Prediction Results

Table 7 shows the various nondestructive testing (NDT) methods and machine learning approaches used in previous research to predict concrete compressive strength. Shin et al. [7] introduced deep convolutional neural networks (DCNNs) as an alternative approach for predicting concrete compressive strength using digital images of concrete surfaces.

The models ConcNet_A, ConcNet_G, and ConcNet_R achieved RMSE values of 3.82, 3.64, and 3.56, respectively. These results demonstrated that DCNNs could reasonably approximate concrete strength from image data. However, their accuracy remained slightly lower than some best-performing machine learning models.

Lee et al. [8] proposed additional DCNN-based methods, using digital images captured via DSLR and microscope devices to predict compressive strength, achieving RMSE values of 3.779 and 4.875, respectively. Another recent model by Yu et al. [23] used a CNN to predict the compressive strength of pervious concrete and achieved an RMSE of 2.72. Similarly, Jang et al. [22] employed a ResNet-based model using digital microscope images and obtained an RMSE of 4.463.

Our new approach, the DCNN model developed using ResNet50V2 with Transfer Learning in the present study, yielded a significantly lower RMSE of 1.23. This indicated that the proposed model achieved a higher predictive accuracy than previous NDT-based and machine learning-based methods. The substantial reduction in RMSE suggested that deep learning techniques, particularly those leveraging pre-trained architectures such as ResNet50V2, could extract meaningful features from concrete images and accurately estimate compressive strength.

## 5. CONCLUSIONS

This study demonstrated the effectiveness of a ResNet50V2-based Deep Convolutional Neural Network (DCNN) for nondestructive prediction of concrete compressive strength using mobile phone imagery. Based on experimental results and statistical analysis, the following conclusions are drawn:

- The developed Deep Convolutional Neural Network (DCNN) using ResNet50V2 architecture achieved a coefficient of determination ($R^2$) of 0.9691, indicating that 96.91% of the variability in concrete compressive strength was successfully predicted from image data.
- The model obtained a Root Mean Square Error (RMSE) of 1.2349 MPa and a Mean Absolute Percentage Error (MAPE) of 1.8651%, demonstrating high prediction accuracy and minimal deviation from actual test results.
- A paired t-test between true compressive strength values and those predicted by the Python-based model showed a mean difference of -0.008 MPa and a p-value of 0.9968, confirming no statistically significant difference ($\alpha = 0.05$) and validating model accuracy.
- A second paired t-test comparing predictions from the Python-based model and the Android-based mobile application yielded a p-value of 0.9148, indicating no significant difference between platforms and confirming the mobile app's reliability.
- The TensorFlow Lite (TFLite) version of the model was optimized for real-time deployment, reducing the model size to 93.8 MB and achieving an average prediction speed of 34 milliseconds per image on standard Android devices.
- The model outperformed previous nondestructive testing (NDT) and machine learning approaches regarding RMSE, achieving a value of 1.23, compared to RMSEs of 3.56–12.4 from existing studies.
- Final testing on unseen data showed consistent predictive performance, with minor underestimations and overestimations within an acceptable range and a final testing RMSE of 2.35 MPa, supporting the model's generalizability.

This study highlights the potential of integrating deep learning with mobile technology for fast, reliable, and nondestructive concrete evaluation. This research sets a foundation for adopting AI-driven solutions in construction quality control by demonstrating high prediction accuracy, cross-platform reliability, and real-time performance.

Future studies can build on this framework by extending it to other materials, embedding it into smart infrastructure systems, and combining image-based AI with complementary sensing technologies to advance the field of intelligent structural assessment.

## 6. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Engineering Research and Development for Technology (ERDT) program for supporting this research. The authors also gratefully acknowledge Engr. Wyndell A. Almenor, Engr. Rolando J. Quitalig, Engr. Charity Hope A. Gayatin, and Engr. Ma. Paulina Zaizah S. Quinto, for their invaluable guidance, technical insights, and unwavering support throughout this study.

## 7. REFERENCES

[1] Moein M. M., Saradar A., Rahmati K., Ghasemzadeh H., Bristow J., Aramali V., and Karakouzian M., Predictive models for concrete properties using machine learning and deep learning approaches: A review, J. Build. Eng., Vol.63, 2022, pp.105444. https://doi.org/10.1016/j.jobe.2022.105444

[2] Zhang W., Guo J., Ning C., Cheng R., and Liu Z., Prediction of concrete compressive strength using a Deepforest-based model, Scientific Reports, Vol.14, 2024, Article 18918. https://doi.org/10.1038/s41598-024-69616-9

[3] Lange D. A., Jennings H. M., and Shah S. P., Image analysis techniques for characterization of pore structure of cement-based materials, Cement and Concrete Research, Vol.24, No.5, 1994, pp.841–853. https://doi.org/10.1016/0008-8846(94)90004-3

[4] Başyiğit C., Çomak B., Kılınçarslan Ş., and Üncü İ. S., Assessment of concrete compressive strength by image processing technique, Constr. Build. Mater., Vol.37, 2012, pp.526–532. https://doi.org/10.1016/j.conbuildmat.2012.07.055

[5] Dogan G., Arslan M. H., and Ceylan M., Concrete compressive strength detection using image processing based new test method, Measurement, Vol.109, 2017, pp.137–148. https://doi.org/10.1016/j.measurement.2017.05.051

[6] Waris M. I., Mir J., Plevris V., and Ahmad A., Predicting compressive strength of CRM samples using image processing and ANN, IOP Conf. Ser.: Mater. Sci. Eng., Vol.899, 2020, 012014. https://doi.org/10.1088/1757-899X/899/1/012014

[7] Shin H. K., Ahn Y. H., Lee S. H., and Kim H. Y., Digital vision based concrete compressive strength evaluating model using deep convolutional neural network, Comput. Mater. Continua, Vol.61, No.3, 2019, pp.911–928. https://doi.org/10.32604/cmc.2019.08269

[8] Lee S., Ahn Y. H., and Kim H., Predicting concrete compressive strength using deep convolutional neural network based on image characteristics, Comput. Mater. Continua, Vol.65, 2020, pp.1–17. https://doi.org/10.32604/cmc.2020.011104

[9] Cuasay R. M., Tan N. G., Torres J. N., and Estores G., A Faster Region-Based Convolutional Neural Network Approach to Automated Structural Damage Recognition and Detection of Reinforced Concrete Bridge Structures, AIP Conference Proceedings, Vol. 3034, Issue 1, March 2024, pp. 050008-1–050008-8. https://doi.org/10.1063/5.0194714

[10] Mayya A., Alkayem N. F., Shen L., Zhang X., Fu R., Wang Q., and Cao M., Efficient hybrid ensembles of CNNs and transfer learning models for bridge deck image-based crack detection, Structures, Vol.64, 2024, Article 106538. https://doi.org/10.1016/j.istruc.2024.106538

[11] Pham T. T., Nguyen T. T., Nguyen L. N., and Nguyen P. V., A neural network approach for predicting hardened property of geopolymer concrete, International Journal of GEOMATE, Vol.19, Issue 74, 2020, pp.176–184. https://geomatejournal.com/geomate/article/view/1873

[12] Doğan G., Özkış A., and Arslan M., A new methodology based on artificial intelligence for estimating the compressive strength of concrete from surface images, Ingeniería e Investigación, Vol.44, 2024, pp.e99526. https://doi.org/10.15446/ing.investig.99526

[13] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jozefowicz R., Jia Y., Kaiser L., Kudlur M., ... Zheng X., TensorFlow: Large-scale machine learning on heterogeneous systems, Software available from https://www.tensorflow.org/, 2015.

[14] Chollet F., and others, Keras, GitHub repository, 2015. Retrieved March 27, 2025, from https://github.com/keras-team/keras

[15] He K., Zhang X., Ren S., and Sun J., Identity mappings in deep residual networks, Proc. Eur. Conf. on Computer Vision (ECCV), 2016, pp.630–645. https://doi.org/10.1007/978-3-319-46493-0_38

[16] David R., Duke J., Jain A., Janapa Reddi V., Jeffries N., Li J., Kreeger N., Nappier I., Natraj M., Regev S., Rhodes R., Wang T., and Warden P., TensorFlow Lite Micro: Embedded machine learning on TinyML systems, arXiv preprint, arXiv:2010.08678, 2021.

https://doi.org/10.48550/arXiv.2010.08678
[17] Android Studio, Android IDE for application development, Google, Software available from https://developer.android.com/studio
[18] Chen J. H., Su M. C., Cao R., Hsu S. C., and Lu J. C., A self-organizing map optimization-based image recognition and processing model for bridge crack inspection, Autom. Constr., Vol.73, 2017, pp.58–66. https://doi.org/10.1016/j.autcon.2016.08.033
[19] Chen X., Fu J., Yao J., and Gan J., Prediction of shear strength for squat RC walls using a hybrid ANN–PSO model, Eng. Comput., Vol.34, 2018, pp.1–17. https://doi.org/10.1007/s00366-017-0547-5
[20] Rashid K. and Waqas R., Compressive strength evaluation by nondestructive techniques: An automated approach in construction industry, J. Build. Eng., Vol.12, 2017, pp.147–154. https://doi.org/10.1016/j.jobe.2017.05.010
[21] Yaseen Z., Deo R., Hilal A., Abd A., Cornejo Bueno L., Salcedo-Sanz S., and Nehdi M., Predicting compressive strength of lightweight foamed concrete using extreme learning machine model, Adv. Eng. Softw., Vol.115, 2018, pp.112–125. https://doi.org/10.1016/j.advengsoft.2017.09.004
[22] Jang Y., Ahn Y. H., and Kim H. Y., Estimating compressive strength of concrete using deep convolutional neural networks with digital microscope images, J. Comput. Civil Eng., Vol.33, No.3, 2019, Article 04019018. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000837
[23] Yu G., Zhu S., and Xiang Z., The prediction of pervious concrete compressive strength based on a convolutional neural network, Buildings, Vol.14, 2024, pp.907. https://doi.org/10.3390/buildings14040907