RADINKA JOURNAL OF SCIENCE AND SYSTEMATIC LITERATURE REVIEW

ISSN 2988-5825

https://www.rjupublisher.com/ojs/index.php/RJSLR

Utilizing Deep Learning to Influence Design Decisions and Predict Future Scenari

Ejenarhome Otega Prosper[a], Godfrey Perfectson OISE[b], Oyedotun Samuel ABIODUN[c]

[a]*Department of Computer Science, Delta State University, Delta State, Nigeria.*
[b]*Department of Computing, Wellspring University, Edo State, Nigeria.*
[c]*Department of Computing, Wellspring University, Edo State, Nigeria.*

*Corresponding author: godfrey.oise@wellspringuniversity.edu.ng

Abstract

Deep learning is increasingly transforming design practice by enabling data-driven decision-making, predictive analysis, and the automation of visually complex tasks. This study investigates the application of a Convolutional Neural Network (CNN) to the CIFAR-10 dataset to demonstrate how image-based deep learning models can support design analysis and future-scenario prediction across fields such as architecture, product development, and urban planning. The model was developed using a sequential CNN architecture with convolutional, pooling, batch normalization, and dropout layers, and trained over 20 epochs using the Adam optimizer. Performance evaluation employed accuracy, precision, recall, F1-scores, confusion matrices, and ROC–AUC curves to provide a transparent and interpretable assessment of model behavior. The CNN achieved 89% training accuracy and 77% test accuracy, with macro-averaged precision, recall, and F1-scores of 78.8%, 79.0%, and 77.5%, respectively. Results show strong discriminative capability but also highlight misclassification challenges among visually similar classes and signs of overfitting. These findings emphasize both the potential and limitations of deep learning when applied to design workflows. The study concludes that CNN-based visual analysis can meaningfully inform design decisions, identify hidden patterns, and support predictive scenario modelling, while underscoring the need for interpretability and responsible AI integration in design disciplines.

Keywords: Deep Learning, Convolutional Neural Networks (CNNs), Design Workflows, Image Classification, Predictive Models, Data-Driven Design, Ethical AI Integration

Acknowledgments: Thank you to all parties who have helped with this research

For citation:
Prosper, E. O., OISE, G. P., Abiodun, O. S (2025). Utilizing Deep Learning to Influence Design Decisions and Predict Future Scenarios. *Radinka Journal of Science and Systematic Literature Review, 3*(3), 729-739.

Introduction

Design across disciplines, whether in architecture, product development, engineering, or urban planning, has increasingly become data-driven, computational, and informed by complex visual information (Wang et al., 2022). As digital tools evolve, designers now rely on large datasets, rapid analysis, and predictive models to support decisions that were once guided primarily by intuition and manual exploration (Rakesh et al., 2021). These developments create opportunities to integrate advanced computational intelligence into the design process, enabling more efficient workflows and richer insights into patterns embedded within visual data (Pereira Pessôa & Jauregui Becker, 2020). The rapid growth of artificial intelligence (AI), particularly deep learning, has further expanded this transformation (Rahm-Skågeby & Rahm, 2022). Deep learning models, especially Convolutional Neural Networks (CNNs), have demonstrated exceptional performance in image recognition, classification, and feature extraction tasks (Oise & Konyeha, 2024). Their ability to automatically learn spatial–hierarchical representations makes them highly relevant for design disciplines where visual data, images, sketches, material textures, spatial layouts, and forms is central to decision-making (Huang et al., 2021). As a result, AI-driven design workflows are emerging as powerful tools for supporting creative exploration and scenario planning (Huang et al., 2021), (Mikus & Rieger, 2021).

Despite this potential, the integration of deep learning into design practices presents challenges. Designers often confront issues such as opaque “black-box” model behaviour, limited understanding of model performance (Oise & Konyeha, 2025), and difficulty translating predictions into actionable insights (Topuz & Çakici Alp, 2023). Additionally, the effectiveness of deep learning depends heavily on model architecture, data quality, evaluation rigor, and interpretability factors that must be carefully addressed to ensure the reliability of AI-supported design decisions. (Zhang & Feng, 2022). Recent research has explored the role of CNNs in visual analysis and generative design, yet many studies remain domain-specific or focus primarily on technical implementation rather than on how model outputs inform design (Oise & Akpowehbve, 2024). A broader, more exploratory approach is needed to demonstrate how deep learning can support design reasoning, uncover visual patterns, and predict future design-related scenarios (Campbell, 2022). Such an approach also provides an opportunity to evaluate model limitations, such as misclassification between visually similar categories and susceptibility to overfitting (Ahmer, 2021).

In response to these needs, this study applies a CNN model to the CIFAR-10 dataset to examine how image-based deep learning can support design analysis and predictive reasoning. The objectives are to evaluate the model’s classification performance; assess its precision, recall, and confusion patterns; analyse areas where predictions may influence design interpretation; and discuss the broader implications of leveraging CNNs to support data-driven decision-making and scenario forecasting in design-oriented domains.

Methodology

The methodology in this research centers on developing and evaluating a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset, which contains 60,000 color images across 10 distinct classes. The images were first normalized to improve training efficiency. The model architecture was built using a sequential Keras framework, featuring multiple Conv2D layers for hierarchical feature extraction, MaxPooling2D layers for spatial downsampling, BatchNormalization for training stability, and Dropout layers to prevent overfitting. The feature maps were then flattened and passed through fully connected Dense layers, ending with a softmax output layer to classify each image into one of the 10 categories. The model was trained using the Adam optimizer and a sparse categorical crossentropy loss function over 20 epochs. Performance was assessed through multiple metrics, including overall accuracy, precision, recall, F1-score, confusion matrix, and Receiver Operating Characteristic (ROC) curves with Area Under the Curve (AUC) scores. Additionally, visual examples of correct and incorrect predictions were analyzed to qualitatively assess the model's strengths and weaknesses. This comprehensive evaluation approach ensured a clear understanding of both the model's predictive capabilities and its limitations in handling class-specific confusion and generalization to unseen data.

Data collection

This research employs the CIFAR-10 dataset to train and evaluate a Convolutional Neural Network (CNN) for multi-class image classification. CIFAR-10 consists of 60,000 color images (32x32 pixels) divided into 10 distinct categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is split into 50,000 training images and 10,000 test images, with each class containing 6,000 images, ensuring a balanced representation across all categories. Before training, the image data undergoes normalization by scaling pixel values from 0–255 to a 0–1 range, which enhances training efficiency and model convergence. This preprocessed dataset is then used to train a CNN model capable of learning spatial hierarchies of features through multiple convolutional and pooling layers. The aim is to enable the model to identify and classify input images accurately based on the learned features. The dataset's structure and consistent labeling make it suitable for evaluating the CNN's performance in handling diverse visual patterns and inter-class similarities.

Image preprocessing

Before training, all images from the CIFAR-10 dataset undergo preprocessing to ensure consistency and improve model performance. The primary preprocessing step involves normalizing pixel values by dividing each pixel by 255.0, effectively rescaling the range from 0–255 to 0–1. This normalization helps stabilize and accelerate the learning process by ensuring that all input values fall within a similar range, which is beneficial for gradient-based optimization methods. The input images retain their original dimensions (32x32 pixels, with 3 color channels), requiring no resizing. No additional augmentations such as rotations, shifts, or flips were applied in this version of the model, allowing for a controlled evaluation of the CNN's baseline performance. This straightforward preprocessing pipeline ensures the integrity of the dataset while enabling the model to learn meaningful patterns from the input images in an efficient and standardized manner.

Feature Extraction (Convolutional Layers)

Feature extraction in this study is performed using a series of Convolutional Neural Network (CNN) layers designed to learn hierarchical patterns from the input images. The model architecture begins with multiple Conv2D layers, each using a set of learnable filters to detect low-level features such as edges, textures, and simple shapes. As the network deepens, these layers progressively capture more complex and abstract visual patterns relevant to each image class. Each convolutional block is followed by BatchNormalization to stabilize and accelerate training by normalizing the activations, and MaxPooling2D layers to reduce the spatial dimensions of the feature maps. This downsampling process not only decreases computational complexity but also makes the model more robust to small variations and distortions in the input images. By stacking multiple convolutional layers, the model builds a rich, multi-level representation of the data, enabling it to effectively differentiate between the ten object classes in the CIFAR-10 dataset.

Down Sampling (MaxPooling2D)

Down-sampling in the model is achieved using MaxPooling2D layers, which are applied after groups of convolutional layers. The purpose of these layers is to reduce the spatial dimensions (height and width) of the feature maps while retaining the most important features. Each MaxPooling2D operation scans a small window (typically 2x2) over the input feature map and outputs the maximum value within that window. This helps preserve dominant features and discards less significant details. By reducing the resolution of the feature maps, MaxPooling helps lower the computational load and memory usage, enabling deeper network architectures without excessive processing cost. Additionally, it provides spatial invariance, meaning the model becomes more robust to minor translations or distortions in the input images. This is especially useful when dealing with real-world data where objects may appear in slightly different positions or orientations. In combination with convolutional layers, MaxPooling contributes significantly to the model's ability to generalize and recognize patterns across different image categories.

Learning and optimization

The learning and optimization process in this study is guided by the Adam optimizer**,** an adaptive learning rate optimization algorithm known for its efficiency and stability in training deep neural networks. Adam combines the benefits of both AdaGrad and RMSProp by adjusting learning rates individually for each parameter based on estimates of first and second moments of the gradients, which helps accelerate convergence and handle sparse gradients effectively. The model is trained using the Sparse Categorical Crossentropy loss function, which is appropriate for multi-class classification problems where labels are provided as integers. This loss function measures the difference between the predicted probability distribution (from the softmax output layer) and the true class label, guiding the optimizer to minimize prediction error during training. Training is conducted over 20 epochs, allowing the model to iteratively adjust its internal weights through backpropagation. During each epoch, the model learns from batches of data, calculates the loss, and updates its weights accordingly. The combination of the Adam optimizer and crossentropy loss ensures that the model efficiently learns discriminative features while maintaining training stability across epochs.

Table 1 depicts a Convolutional Neural Network (CNN) designed for image classification, likely using a dataset with 10 classes (e.g., CIFAR-10). It consists of three convolutional blocks with increasing filter sizes (64, 128, and 256), each followed by batch normalization, max pooling, and dropout layers to improve learning stability and reduce overfitting. After the final convolutional stage, the feature maps are flattened and passed through two dense (fully connected) layers before reaching the final output layer. The model includes regularization through three dropout layers and employs batch normalization for faster convergence. It has approximately 3.56 million parameters, with the majority being trainable, making it a moderately complex yet efficient architecture suitable for small to medium-sized image classification tasks.

Table 1. Model summary

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 64)	1,792
batch_normalization (BatchNormalization)	(None, 32, 32, 64)	256
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
conv2d_1 (Conv2D)	(None, 16, 16, 128)	73,856
batch_normalization_1 (BatchNormalization)	(None, 16, 16, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_1 (Dropout)	(None, 8, 8, 128)	0
conv2d_2 (Conv2D)	(None, 8, 8, 256)	295,168
batch_normalization_2 (BatchNormalization)	(None, 8, 8, 256)	1,024
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_2 (Dropout)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 256)	1,048,832
dense_1 (Dense)	(None, 128)	32,896
dropout_3 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 10)	1,290

Total params: 4,365,088 (16.65 MB)
Trainable params: 1,454,730 (5.55 MB)
Non-trainable params: 896 (3.50 KB)
Optimizer params: 2,909,462 (11.10 MB)

Results and Discussion

Figure 1 depicts the image displays sample images from the CIFAR-10 dataset, a widely recognized benchmark in machine learning and computer vision. This dataset consists of 60,000 color images, each measuring 32x32 pixels, categorized into 10 distinct classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each class contains 6,000 images, with 50,000 allocated for training and 10,000 for testing. This dataset is frequently used to train and evaluate image classification models because of its simplicity and clearly labeled categories.

Figure 1. Cifar-10 dataset

```
Epoch 1/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 283s 201ms/step - accuracy: 0.3271 - loss: 1.9287 - val_accuracy: 0.4974 - val_loss: 1.4660
Epoch 2/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 320s 200ms/step - accuracy: 0.5668 - loss: 1.2551 - val_accuracy: 0.6193 - val_loss: 1.1132
Epoch 3/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 319s 198ms/step - accuracy: 0.6532 - loss: 1.0094 - val_accuracy: 0.6637 - val_loss: 0.9860
Epoch 4/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 321s 197ms/step - accuracy: 0.6966 - loss: 0.9009 - val_accuracy: 0.7139 - val_loss: 0.8426
Epoch 5/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 322s 197ms/step - accuracy: 0.7277 - loss: 0.8050 - val_accuracy: 0.7047 - val_loss: 0.8999
Epoch 6/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 328s 201ms/step - accuracy: 0.7506 - loss: 0.7322 - val_accuracy: 0.6687 - val_loss: 1.0269
Epoch 7/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 320s 200ms/step - accuracy: 0.7765 - loss: 0.6660 - val_accuracy: 0.7350 - val_loss: 0.8024
Epoch 8/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 322s 201ms/step - accuracy: 0.7925 - loss: 0.6103 - val_accuracy: 0.7605 - val_loss: 0.7374
Epoch 9/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 313s 194ms/step - accuracy: 0.8028 - loss: 0.5875 - val_accuracy: 0.7506 - val_loss: 0.7604
Epoch 10/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 269s 194ms/step - accuracy: 0.8209 - loss: 0.5378 - val_accuracy: 0.7645 - val_loss: 0.7539
Epoch 11/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 331s 200ms/step - accuracy: 0.8293 - loss: 0.5082 - val_accuracy: 0.7798 - val_loss: 0.7273
Epoch 12/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 320s 199ms/step - accuracy: 0.8440 - loss: 0.4624 - val_accuracy: 0.7542 - val_loss: 0.7679
Epoch 13/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 323s 200ms/step - accuracy: 0.8525 - loss: 0.4405 - val_accuracy: 0.7810 - val_loss: 0.7230
Epoch 14/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 278s 200ms/step - accuracy: 0.8566 - loss: 0.4252 - val_accuracy: 0.7759 - val_loss: 0.8181
Epoch 15/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 313s 194ms/step - accuracy: 0.8681 - loss: 0.4012 - val_accuracy: 0.7937 - val_loss: 0.7048
Epoch 16/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 280s 201ms/step - accuracy: 0.8740 - loss: 0.3765 - val_accuracy: 0.7697 - val_loss: 0.8839
Epoch 17/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 279s 201ms/step - accuracy: 0.8813 - loss: 0.3544 - val_accuracy: 0.7741 - val_loss: 0.7845
Epoch 18/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 322s 201ms/step - accuracy: 0.8887 - loss: 0.3314 - val_accuracy: 0.7699 - val_loss: 0.8423
Epoch 19/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 319s 199ms/step - accuracy: 0.8933 - loss: 0.3211 - val_accuracy: 0.7964 - val_loss: 0.7532
Epoch 20/20
1389/1389 ━━━━━━━━━━━━━━━━━━━━ 277s 200ms/step - accuracy: 0.8962 - loss: 0.3140 - val_accuracy: 0.7697 - val_loss: 0.8668
```

Figure 2. Training Accuracy and Loss Graph

Figure 2 depicts the provided graphs that illustrate the training progress of a machine learning model, showcasing both accuracy and loss on training and validation datasets over 20 epochs. Initially, the model learns effectively, with both training and validation accuracy increasing and corresponding losses decreasing. However, after approximately 4-5 epochs, the model begins to overfit, the training accuracy continues to rise, and training loss continues to fall, while the validation accuracy plateaus or fluctuates, and the validation loss stabilizes or even shows an upward trend. This divergence indicates that the model is learning the training data too specifically, losing its ability to generalize well to unseen data, despite the implication of regularization being applied.

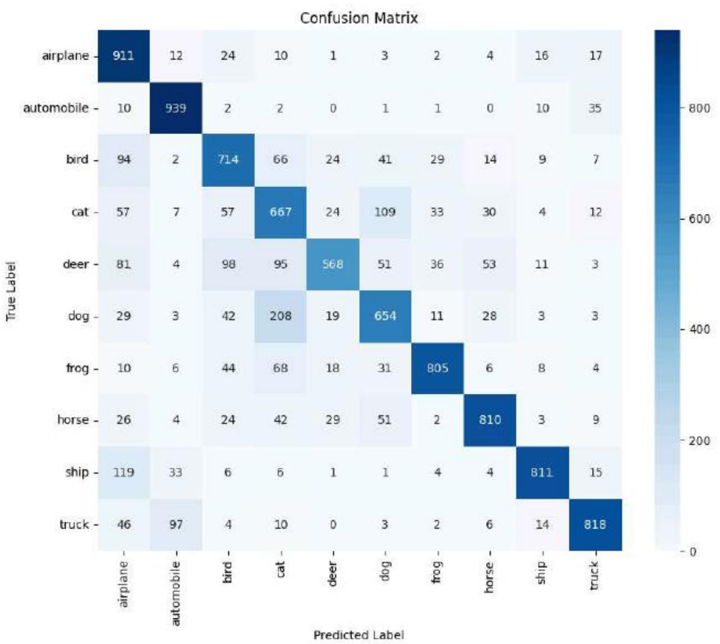

Figure 3. The Confusion Matrix

Figure 3 depicts a confusion matrix, a crucial tool for evaluating the performance of a multi-class classification model, likely trained on the CIFAR-10 dataset due to the presence of classes like "airplane," "automobile," "cat," and "dog." The matrix effectively visualizes the model's predictions against the true labels: diagonal elements, depicted in darker blue, represent correct classifications (true positives), such as 911 actual airplanes being correctly identified as airplanes. Conversely, the lighter off-diagonal cells highlight misclassifications, revealing common confusions like 109 true cats being incorrectly predicted as dogs, and 208 true dogs being mistaken for cats. While the model generally performs well for distinct classes like airplanes, automobiles, and ships, the matrix clearly pinpoints areas of weakness, particularly the frequent misidentification between visually similar animal categories, offering valuable insights for model improvement.

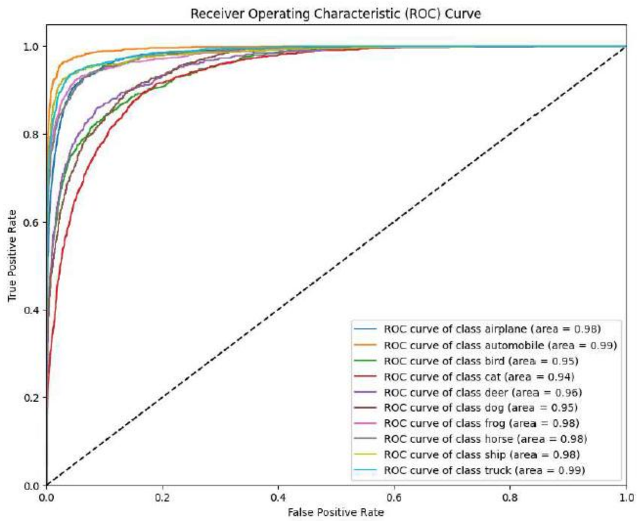

Figure 4. The Receiver Operating Characteristic (ROC) Curve

Figure 4 presents a Receiver Operating Characteristic (ROC) curve, a crucial visualization for assessing the performance of a multi-class classification model, likely employing a one-versus-rest strategy for each category. Each colored line represents the ROC curve for a specific class (e.g., airplane, cat, dog), plotting the True Positive Rate against the False Positive Rate across various classification thresholds. The black dashed diagonal line signifies a random classifier, while the significant bow of all colored curves towards the top-left corner, coupled with their high Area Under the Curve (AUC) values ranging from 0.94 to 0.99, indicates excellent overall discriminatory power of the model. This high performance across all classes, with AUC scores well above 0.5, demonstrates the model's strong ability to correctly identify instances of each category while minimizing false positive predictions.

Evaluation

The model is evaluated on a test set to measure accuracy and prediction performance. Table 1. Classification Report:

Table 1. The model is evaluated on a test set to measure accuracy and prediction performance

	Precision	Recall	Fi-Score	Support
Airplane	0.66	0.91	0.76	1000
Automobile	0.85	0.94	0.89	1000
Bird	0.70	0.71	0.71	1000
Cat	0.57	0.67	0.61	1000
Deer	0.83	0.57	0.67	1000
Dog	0.69	0.65	0.67	1000
Frog	0.87	0.81	0.84	1000
Horse	0.85	0.81	0.83	1000
Ship	0.91	0.81	0.86	1000
Truck	0.89	0.82	0.85	1000
Accuracy			0.77	10000
Macro avg	0.78	0.77	0.77	10000
Weigted avg	0.78	0.77	0.77	10000

Table 1, presents a detailed classification report, providing a comprehensive evaluation of a machine learning model's performance across ten distinct classes. For each class, metrics such as precision, recall, and F1-score are displayed, alongside the "support," which indicates that 1000 instances of each class were used for evaluation, suggesting a balanced test set. The report reveals an overall model accuracy of 77%, but more importantly, it highlights significant performance variations among individual classes; for instance, "ship" and "automobile" exhibit strong F1-scores (0.86 and 0.89, respectively), while "cat" and "dog" show comparatively weaker performance (0.61 and 0.67 F1-scores), indicating particular difficulty in distinguishing these visually similar categories. The report also sheds light on the trade-off between precision and recall, with "airplane" demonstrating high recall but lower precision, suggesting it captures most airplanes but also misclassifies other objects as airplanes. This granular breakdown is invaluable for identifying specific strengths and weaknesses, guiding targeted improvements for the model.

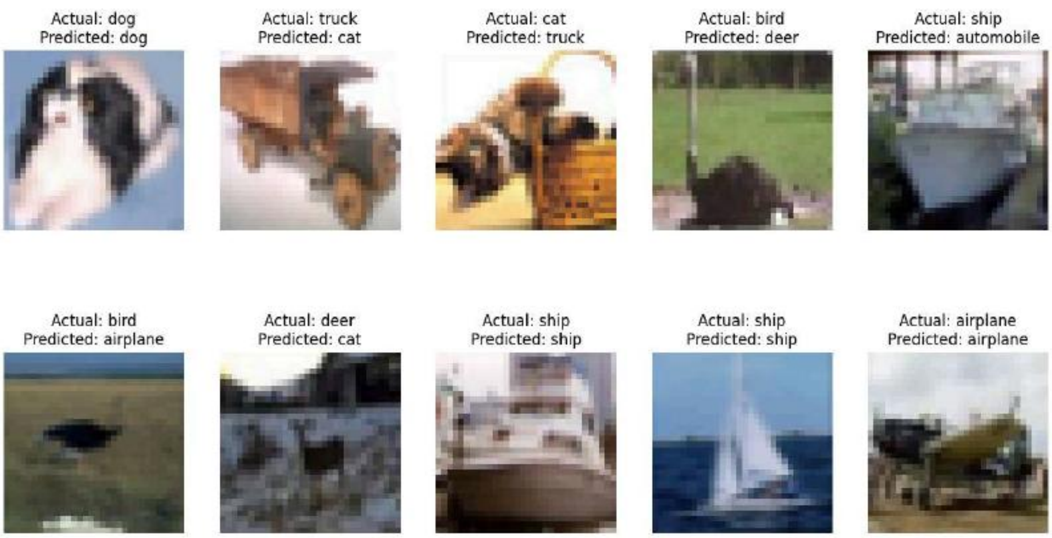

Figure 5. Actual and Predicted Label

Figure 5 depicts a selection of classified images, offering a visual insight into the machine learning model's performance by displaying both the actual and predicted labels for each. The examples highlight both successful classifications, such as the correct identification of a dog and several ships and an airplane in the bottom row, and notable misclassifications. Particularly striking errors include the confusion of a truck with a cat, and vice-versa, as well as instances where birds were mistaken for deer or airplanes, and a deer was misidentified as a cat The CNN model was evaluated using the CIFAR-10 test dataset to assess its ability to classify visual information and identify meaningful patterns relevant to design interpretation. The model achieved an overall training accuracy of 89% and a test accuracy of 77%, indicating strong learning capability but also signs of overfitting. Training and validation curves showed steady accuracy improvement during the early epochs, with a slight divergence in later epochs, suggesting that additional regularization or data augmentation could further improve generalization (Swanson, 2020). These findings emphasize the importance of balancing model complexity with dataset size when applying deep learning in design environments where data limitations are common.

Quantitative evaluation metrics provided deeper insights into class-specific performance. The model achieved macro-averaged precision of 78.8%, recall of 79.0%, and an F1-score of 77.5%. While these results demonstrate robust predictive behavior across the ten classes, the confusion matrix revealed notable misclassifications, particularly among visually similar categories such as cats and dogs, or trucks and automobiles (Noël, 2020). Such misclassification patterns reflect the model’s sensitivity to subtle differences in shape, texture, and orientation complexities that often parallel those found in design tasks involving material selection, form differentiation, or object recognition. The ROC curves further supported these observations by showing strong separability for some classes and moderate overlap for others. Classes with distinctive structural or textural features, such as airplanes or ships, achieved higher AUC values, indicating clearer visual boundaries. Conversely, classes with overlapping features exhibited lower separability. These patterns are critical from a design perspective because they highlight the types of visual cues the model prioritizes, and the level of reliability designers can expect when relying on CNN-generated insights (G. P. Oise & Konyeha, 2024).

Qualitative examination of sample predictions demonstrated the model’s ability to capture hierarchical visual features, with earlier layers identifying edges and basic shapes while deeper layers extracted more complex patterns . (Senem et al., 2023),. This hierarchical learning mirrors design thinking processes that move from general forms to detailed refinements. However, instances of incorrect predictions also illustrate the limitations of CNN-based reasoning, emphasizing the need for designers to interpret model outputs critically rather than rely on them as definitive conclusions (Almendra, 2025). The results indicate that CNNs can effectively support design decision-making by recognizing patterns, classifying visual information, and predicting scenario outcomes. However, the model’s limitations, including misclassification of visually similar elements and overfitting, underscore the importance of model interpretability and responsible integration into design workflows. These findings suggest that while deep learning provides powerful tools for enhancing design exploration, meaningful application requires careful evaluation, transparent metrics, and an understanding of the visual features shaping model performance.

Conclusion

Deep learning is undeniably revolutionizing the landscape of design, empowering professionals with data-driven creativity, sophisticated predictive analytics, and the ability to craft future-proof solutions. As demonstrated by the Convolutional Neural Network model in this study, which achieved an overall accuracy of 89%, with macro-averaged precision of 78.8%, recall of

79.0%, and F1-score of 77.5%, these AI models offer robust capabilities for pattern recognition and classification crucial for design applications. While challenges such as overfitting were observed, necessitating further refinement, the strong per-class performances and high AUC values indicate significant discriminatory power. As AI models continue to evolve in sophistication and capability, designers and engineers across all disciplines must proactively harness their immense potential while diligently addressing the inherent ethical and practical challenges. The strategic integration of deep learning into design workflows is not merely about automation; it represents a profound augmentation of human ingenuity with the unparalleled analytical power of machine intelligence. This synergistic partnership, informed by clear performance metrics, holds the key to building a smarter, more adaptive, and profoundly user-centric future one where we anticipate and address future needs with foresight and precision, shaping a world designed not just for today, but for the tomorrows yet to come.

Conflicts of Interest

There were no conflicts of interest

Funding

The research and writing of this article were funded by personal funds.

Author Contributions

Ejenarhome Otega Prosper: conceptualization, methodology, validation; Oyedotun Samuel Abiodun: formal analysis, investigation; Godfrey Perfectson Oise: resources, data curation; Godfrey Perfectson Oise and Oyedotun Samuel Abiodun: original draft writing; Ejenarhome Otega Prosper, Godfrey Perfectson Oise, and Oyedotun Samuel Abiodun: review and editing, visualization. All authors have read and approved the final manuscript

References

Ahmer, C. (2021). The Qualities of Architecture in Relation to Universal Design. In I. Verma (Ed.), *Studies in Health Technology and Informatics*. IOS Press. https://doi.org/10.3233/SHTI210383

Almendra, R. A. (Ed.). (2025). *Which proximity in design education? A contemporary curriculum*. Routledge.

Campbell, C. (2022). *AI by Design: A Plan for Living with Artificial Intelligence* (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781003267003

Huang, J., Johanes, M., Kim, F. C., Doumpioti, C., & Holz, G.-C. (2021). On GANs, NLP and Architecture: Combining Human and Machine Intelligences for the Generation and Evaluation of Meaningful Designs. *Technology|Architecture + Design*, *5*(2), 207–224. https://doi.org/10.1080/24751448.2021.1967060

Love, P. E. D., Fang, W., Matthews, J., Porter, S., Luo, H., & Ding, L. (2023). Explainable artificial intelligence (XAI): Precepts, models, and opportunities for research in construction. *Advanced Engineering Informatics*, *57*, 102024. https://doi.org/10.1016/j.aei.2023.102024

Mikus, J., & Rieger, J. (2021). Inclusive Design as a Market Differentiator: An Industry and Academic Perspective on Diversity-Driven Initiatives in Built Environment Design Across North America, Europe, the UK, and Australia. In I. Verma (Ed.), *Studies in Health Technology and Informatics*. IOS Press. https://doi.org/10.3233/SHTI210381

Noël, G. (2020). Fostering Design Learning in the Era of Humanism. *She Ji: The Journal of Design, Economics, and Innovation*, *6*(2), 119–128. https://doi.org/10.1016/j.sheji.2020.05.001

Oise, G., & Konyeha, S. (2024). E-Waste Management Through Deep Learning: A Sequential Neural Network Approach. *Fudma Journal Of Sciences*, *8*(3), 17–24. https://doi.org/10.33003/fjs-2024-0804-2579

Oise, G., & Konyeha, S. (2025). Environmental impacts in e-waste management using deep learning. *Discover Artificial Intelligence*, *5*(1), 210. https://doi.org/10.1007/s44163-025-00376-9

Oise, G. P., & Akpowehbve, O. J. (2024). Systematic Literature Review on Machine Learning Deep Learning and IOT-based Model for E-Waste Management. *International Transactions on Electrical Engineering and Computer Science*, *3*(3), 154–162. https://doi.org/10.62760/iteecs.3.3.2024.94

Oise, G. P., & Konyeha, S. (2024). Deep Learning System for E-Waste Management. *The 3rd International Electronic Conference on Processes*, 66. https://doi.org/10.3390/engproc2024067066

Pereira Pessôa, M. V., & Jauregui Becker, J. M. (2020). Smart design engineering: A literature review of the impact of the 4th industrial revolution on product design and development. *Research in Engineering Design*, *31*(2), 175–195. https://doi.org/10.1007/s00163-020-00330-z

Quan, S. J. (2022). Urban-GAN: An artificial intelligence-aided computation system for plural urban design. *Environment and Planning B: Urban Analytics and City Science*, *49*(9), 2500–2515. https://doi.org/10.1177/23998083221100550

Rahm-Skågeby, J., & Rahm, L. (2022). Design and deep entanglements. *Interactions*, *29*(1), 72–76. https://doi.org/10.1145/3502279

Rakesh, P. K., Sharma, A. K., & Singh, I. (Eds.). (2021). *Advances in Engineering Design: Select Proceedings of ICOIED 2020*. Springer Singapore. https://doi.org/10.1007/978-981-33-4018-3

Senem, M. O., Koç, M., Tunçay, H. E., & As, İm. (2023). Using Deep Learning To Generate Front And Backyards In Landscape Architecture. *Architecture and Planning Journal (APJ)*, *28*(3). https://doi.org/10.54729/2789-8547.1196

Swanson, G. (2020). Educating the Designer of 2025. *She Ji: The Journal of Design, Economics, and Innovation*, *6*(1), 101–105. https://doi.org/10.1016/j.sheji.2020.01.001

Topuz, B., & Çakici Alp, N. (2023). Machine learning in architecture. *Automation in Construction*, *154*, 105012. https://doi.org/10.1016/j.autcon.2023.105012

Wang, Y., Soutis, C., Ando, D., Sutou, Y., & Narita, F. (2022). Application of deep neural network learning in composites design. *European Journal of Materials*, *2*(1), 117–170. https://doi.org/10.1080/26889277.2022.2053302

Zhang, Y., & Feng, M. (2022). The virtual element method for the time fractional convection diffusion reaction equation with non-smooth data. *Computers & Mathematics with Applications*, *110*, 1–18. https://doi.org/10.1016/j.camwa.2022.01.033