Manila Journal of Science 18 (1) 2026, pp. 22-34

Impact of Data Splitting Techniques on the Performance of a Convolutional Neural Network Facial Recognition Model

Qudus Lekan Salaudeen[1,*] and Christopher Godwin Udomboso[1,2]
[1]Department of Statistics, University of Lagos, Nigeria
[2]Department of Statistics, University of Ibadan, Nigeria

*Corresponding author: quduslekan24@gmail.com

ABSTRACT

The performance of facial recognition models is influenced by the choice of data partitioning strategy. Appropriate data splitting techniques enable more reliable estimation of model generalization and help assess overfitting. In this study, we examine the performance of a convolutional neural network (CNN)-based facial recognition model under four commonly used data splitting approaches: random splitting, stratified splitting, bootstrap validation, and k-fold cross-validation. Experiments are conducted on two benchmark data sets, Labeled Faces in the Wild and the Olivetti Research Laboratory. Model performance is evaluated using accuracy as well as empirical estimates of model bias and variance. The results indicate that k-fold cross-validation provides a more stable performance estimate under the experimental conditions considered, suggesting its suitability as a validation procedure for CNN-based facial recognition models.

Keywords: bootstrap validation, convolutional neural network, cross-validation, data splitting, evaluation metrics, facial recognition

INTRODUCTION

In recent decades, there has been a significant improvement in the interpretation and categorization of objects from digital images and videos. This advancement has led to the resurgence of various modern technologies such as intelligent surveillance systems, medical imaging, and vehicle recognition (Nurhopipah & Harjoko, 2018). For example, Udomboso et al. (2024) suggested a statistical image processing system for tracking vehicle movements within a confined territory using the Tesseract optical character recognition engine and You Only Look Once version 5 (YOLOv5). The system contributed to a reliable solution for improving security, traffic monitoring, and decision-making. These capabilities have especially been extended to recognizing human faces through algorithms that extract and compare features that can be used to identify and authenticate faces, known as facial recognition.

The wide adoption of biometric systems in recent years led to the increase in the usage of image processing systems, particularly facial recognition. Facial recognition has developed as a replacement to the traditional approach of individual recognition, which includes the use of passwords and personal identification numbers (Wang & Li, 2018). Other biometric-based identification systems, such as fingerprint, retina, and

voice authentication, are gaining traction. While these are also good methods of recognizing individuals, the issue that arises from their usage is that they require users to perform an action before using them. Additionally, voice recognition systems can be unreliable in noisy environments. All these made facial recognition the most popular, as it is cheaper, is easier to use, and does not require actions from users each time it is being used. This made it a reliable recognition system across commercial, forensic, and security sectors (Dantcheva et al., 2016).

The leading algorithm used in facial recognition is convolutional neural networks (CNNs). CNN is a deep learning algorithm that is commonly employed in image and voice recognition systems due to its better performance (Karkuzhali et al., 2024). It is widely used in facial recognition because it is capable of learning and analyzing patterns in images using three main layers, which are the convolution layer, pooling layer, and fully connected layer. Each neuron in neural networks functions similarly to a logistic regression unit and can be expressed as

$$h_{W,b}(x) = f(W^T x) = f\left(\Sigma_{i-1}^{3} W_i x_i + b\right) \quad (1)$$

where x = input vector, b = bias term for the neuron, and W = weight vector (Wang & Li, 2018).

The performance of machine learning models often depends on different factors, which include but are not limited to the sample size, data characteristics, algorithm, and the data splitting strategy employed in evaluating performance. The performance estimates of even a good machine learning model can be misleading if an appropriate data partitioning strategy is not used (Nurhopipah & Uswatun, 2020; Tantithamthavorn et al., 2017). Data splitting, which is often carried out after data preprocessing, involves dividing the data set into at least two subsets, which include, namely, a training set and a testing set to enable evaluation of model performance (Camilo et al., 2019; Genç & Tunç, 2019).

In building a facial recognition model using CNNs, the traditional data splitting technique commonly known as random data sampling might not be the best option, as a data partitioning strategy that will produce an unbiased performance estimate and help detect overfitting is required. Overfitting is a condition where the model produces good performance when it is applied to training data but results poorly when it is applied to unknown samples in the testing process (Lopez et al., 2025; Ying, 2019). Generally, a good data splitting technique should ensure that the samples are good representatives of the population data and are less affected by noise. Vabalas et al. (2019) showed that cross-validation does better than the training-testing set split in this regard by providing estimates of model performance that are more stable and reliable.

However, there is no best data partitioning strategy for all domains (Xu & Goodacre, 2018). In a study of the comparison of machine learning valuation for nonstationary time-series data, Schnaubelt (2019) concluded that using cross-validation for time-series applications comes at a great risk, as it often produces a model with larger bias and variance compared to other data validation methods, and recommended that blocked variants of cross-validation should be employed if it is to be used.

The most commonly used variant of cross-validation is k-fold cross-validation.

Nurhopipah and Uswatun (2020) concluded that k-fold cross-validation produces more stable performance than the other data splitting techniques studied including random splitting. According to the study, k-fold cross-validation produced the highest accuracy and least bias-variance tradeoffs, with the Moralis–Lima–Martin Validation splitting technique having the lowest performance. Although cross-validation often produces reliable estimates of model performance, it has been shown that stratified cross-validation tends to produce better performance estimates (Kohavi, 1995).

While data partitioning techniques, such as random splitting and k-fold cross-validation, are often used in evaluating machine learning models, it is necessary to consider other data splitting techniques, as the two validation techniques are not without limitations (Bischl et al., 2012; Ramezan et al., 2019). Udomboso et al. (2025) pointed out that although random splitting often enhances evaluation of model performance by separating the data set into samples that are good representatives, it tends to perform poorly on imbalanced data sets, with the model having high bias and variance. In contrast,

k-fold cross-validation provides more reliable performance estimates, but it requires more computational power, especially in large data sets, and is generally time-consuming. Bootstrap validation, on the other hand, is often associated with low bias and high variance, which might make its performance estimates less stable (Kohavi, 1995; Moss et al., 2018).

Although prior extensive studies have shown that data splitting strategies influence the performance evaluation of machine learning models, empirical comparisons of these data splitting procedures specifically for CNN-based facial recognition models using accuracy, bias, and variance remain limited. Furthermore, many existing studies focus on accuracy without analyzing the bias-variance tradeoff to explain the stability of the validation technique. This study will contribute by examining the performance of four different data splitting procedures on CNN-based facial recognition, evaluated using accuracy, bias, and variance, to determine the data partitioning strategy with the most stable performance for facial recognition.

MATERIALS AND METHODS

Research Design

This study focused on how data splitting techniques impact generalization of facial recognition models using CNNs. After a thorough preprocessing, the data sets were divided into two parts: training sets and test sets using each data validation technique. The CNN algorithm was applied to the training set of each data set with no hyperparameter tuning, and the test sets were used to measure the performance of the trained model. The classification performance metrics employed are accuracy, bias, and variance. Figure 1 outlines the research design for the investigation of impact of data splitting techniques on the CNN-based facial recognition model.

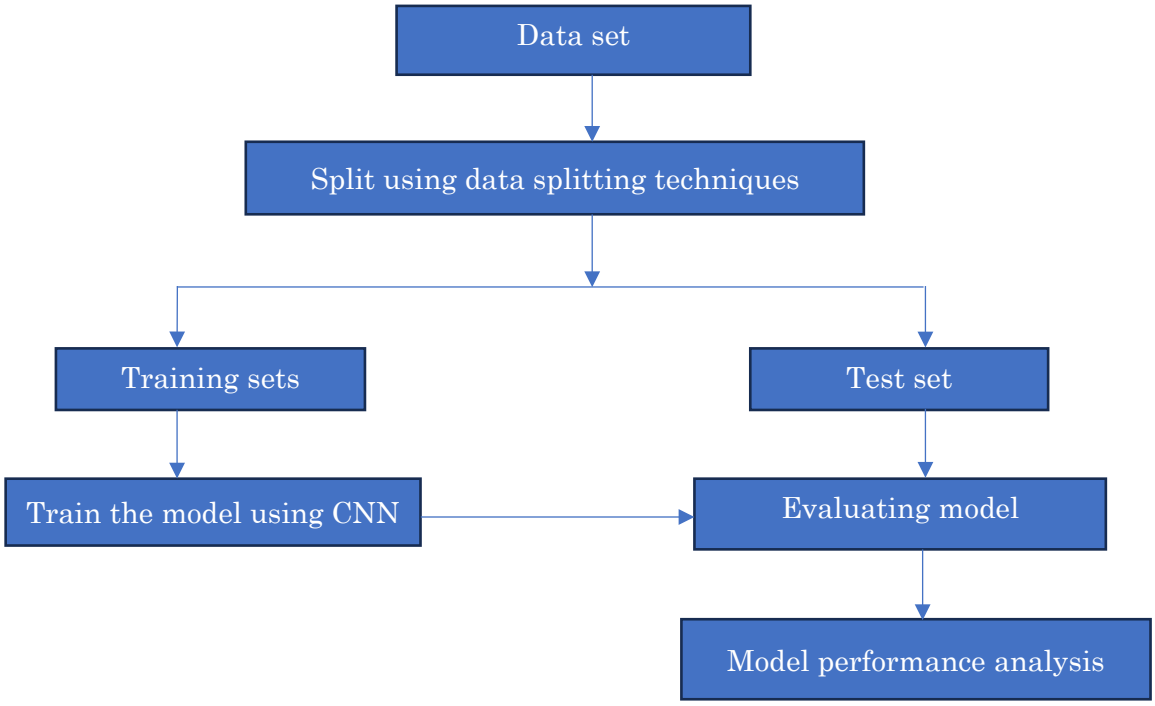


Figure 1. Comparative study of data validation techniques on CNN model. CNN = convolutional neural network.

Data Set Description and Preprocessing

In this study, two data sets, D1 and D2, were used during the analysis. D1 is the LFW (Labeled Faces in the Wild; http://vis-www.cs.umass.edu/lfw/) data set, a database available publicly for facial recognition study. D1 contains over 13,000 facial images each of size 250 × 250 with varying lighting, background, and facial expressions collected from the web. Among the 5,749 individuals represented in the LFW data set, 1,680 have at least two facial images, while only five individuals have at least 100 images in the data set (Huang et al., 2007).

D2 refers to the Olivetti Research Laboratory (http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.tar.Z), a data set of 400 grayscale facial images from 40 unique subjects that were captured at different times, under varying lighting conditions and facial expressions with each subject having 10 images. The resolution of each image in the data set, from the data set source, is 92 × 112 pixels, and the gray level per pixel is 256 (Samaria & Harter, 1994).

During the import of the D1 data set, the data were filtered to include only subjects with at least 100 images resulting in a new data set of five (5) unique subjects. The images were loaded as 128 × 128 color images since CNNs perform better when the input images have equal widths and heights. To make the new data set balanced, a further step was taken to ensure that there are only 100 images per subject in D1. The pixel values were normalized, scaling them to the range [0.1], and the class labels were converted to categorical variables by one-hot encoding. D2 on the other hand, is a balanced data set. Pixel values were normalized, and one-

hot encoding was employed on the class labels. The images were also reshaped to a single-channel dimension yielding input tensors of shape (64, 64, 1).

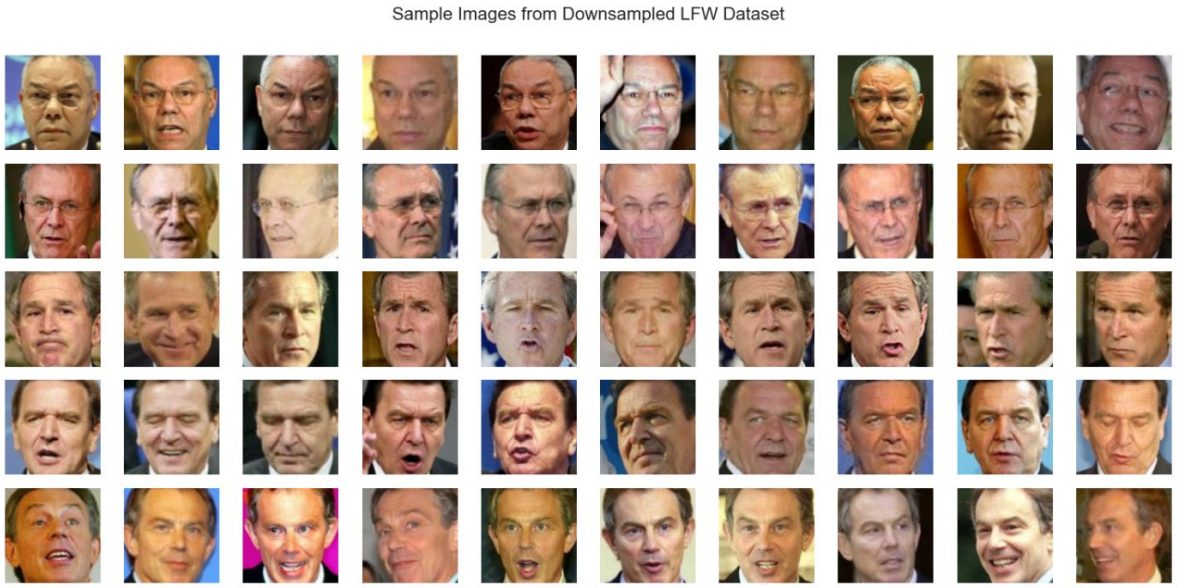


Figure 2a. Sample images in D1 (Labeled Faces in the Wild data set).

Figure 2b. Sample images in D2 (Olivetti Research Laboratory data set).

CNN Model Architecture

The model used is a CNN algorithm containing three pairs of convolution and pooling layers. The first convolutional layer uses 32 filters with a 3 × 3 kernel size and rectified linear unit (ReLU) activation. The subsequent convolutional layers employ 64 filters with the same kernel size and activation function as the first

convolutional layer. Each convolutional layer is followed by a 2×2 max pooling layer.

The resulting 2D feature maps are flattened to a one-dimensional feature vector and passed to a fully connected dense layer with 128 neurons and a ReLU activation function. The output layer is a dense layer with a softmax activation function and a number of neurons corresponding to the number of class labels.

The model was compiled using the Adaptive Moment Estimation (Adam) optimizer with the default learning rate (0.001) and sparse categorical cross-entropy as the loss function. Training was performed with a batch size of 25 for 15 epochs, and model performance was assessed all through the process, using classification accuracy on both the training set and the validation set.

Data Splitting Techniques

Four data splitting techniques were employed in splitting the data sets into training and test sets. Subject-dependent splitting is not enforced so images from the same individuals may appear in both training and testing sets.

Random Subsampling Validation

Definition: Given the data set $D = \{(x_i, y_i)\}_{i=1}^{N}$ to be divided ∋

$$D_{train} \cup D_{test} = D, \quad D_{train} \cap D_{test} = \emptyset \quad (2)$$

where $\cup$ denotes union and $\cap$ denotes intersection with

$$|D\_train| = \lfloor \alpha N \rfloor, \quad |D_{test}| = N - \lfloor \alpha N \rfloor, (3)$$

where $0 < \alpha < 1, \alpha = 0.8$.

The data sets were randomly divided into 80% for training and 20% for testing for model evaluation. This split was chosen to ensure there is a sufficient number of training samples for the model to capture the right features for facial recognition and testing samples for reliable estimation of model performance, which is consistent with common practices in machine learning literature (Hastie et al., 2009). This procedure was done five times to obtain the average model accuracy.

k-Fold Cross-Validation

Definition: Partition D into k disjoint folds ∋

$D = \bigcup_{i=1}^{k} F_i, F_i \cap F_j = \emptyset$ for $i \neq j$ with each fold size $\frac{N}{k}$.

For the jth fold,

$$D_{train}^{(j)} = D \setminus F_j, \quad D_{test}^{(j)} = F_j \quad (4)$$

where k is the number of folds and F denotes each fold.

In k-fold cross-validation, the data set is divided into k equal parts or folds. Each fold is used once for validation while the remaining $k - 1$ folds are used for training (Wang et al., 2021). In this study, k was set to 5, which implies an 80% training and 20% validation split.

Stratified Sampling

Definition: If the data set D has C classes ∋ $D = \bigcup_{i=1}^{k} D_c$, (5)

With $D_c = \{(x_i, y_i) : y_i = c\}$ (6)

Preserve class proportions:

$$\frac{|D_{train,c}|}{|D_{train}|} \approx \frac{|D_c|}{|D|}, \quad \frac{|D_{test,c}|}{|D_{test}|} \approx \frac{|D_c|}{|D|} \quad (7)$$

Stratified data splitting was applied to both D1 and D2 to maintain the same class distribution across the training and testing subsets as in the original data set.

Bootstrap Validation

Definition: Sampling with replacement from D. Construct bootstrap sample:

$$D* = \{(x_1,\ y_1), (x_2,\ y_2), \ldots, (x_{iN}, y_{iN})\}, \quad (8)$$

where each $i_j \sim Uniform(\{1,2,\ldots,N\})$ (9)

Training is done on D*, and the test set is

$$D_{oob} = D \setminus D* \quad (10)$$

n samples were randomly selected with replacement, meaning that the same sample can be selected several times. This selected sample was used as a training set, and the unselected sample was used as a testing set.

Evaluation Metrics

The performance of the models was assessed using accuracy, bias, and variance. These are key metrics that help determine the impact of each data splitting technique on facial recognition model performance.

Accuracy

Accuracy refers to the proportion of all model classifications that were correct, whether positive or negative. It is a performance metric that measures how often a machine learning model makes the right classification (prediction) of the outcome and can be mathematically represented as follows:

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

where TP (true positive) is an instance for which both the classified and actual images are positive, TN (true negative) is an instance for which both the classified and actual images are negative, FP (false positive) is an instance for which the classified image is positive but the actual image is negative, and FN (false negative) is an instance where the model incorrectly classifies a positive instance as negative.

A model that produces no false positives or false negatives achieves a perfect accuracy of 100%.

The accuracy of the CNN model under each data splitting technique is computed over five runs with the mean accuracy recorded.

Bias

Bias represents the difference between the expected prediction of a model and the actual (true) value. A lower difference between these two means indicates lower prediction bias.
If Y denotes the true population parameter and $\hat{Y}$ is its estimate from a sample, the bias of the estimator $\hat{Y}$ can be expressed as follows:

$$bias(Y) = E(\widehat{Y}) - Y \quad (13)$$

where $E(\widehat{Y})$ is the expected value of the estimator.

A model with low bias has a better performance than a model with high bias.

Variance

Variance is a statistical measure of how much a model's predictions are affected when trained on different portions of the same data set. It denotes the model's sensitivity to small variations in the training data.

If $\hat{Y}$ denotes the predicted value and $E(\hat{Y})$ is its expected value, the model variance can be expressed as follows:

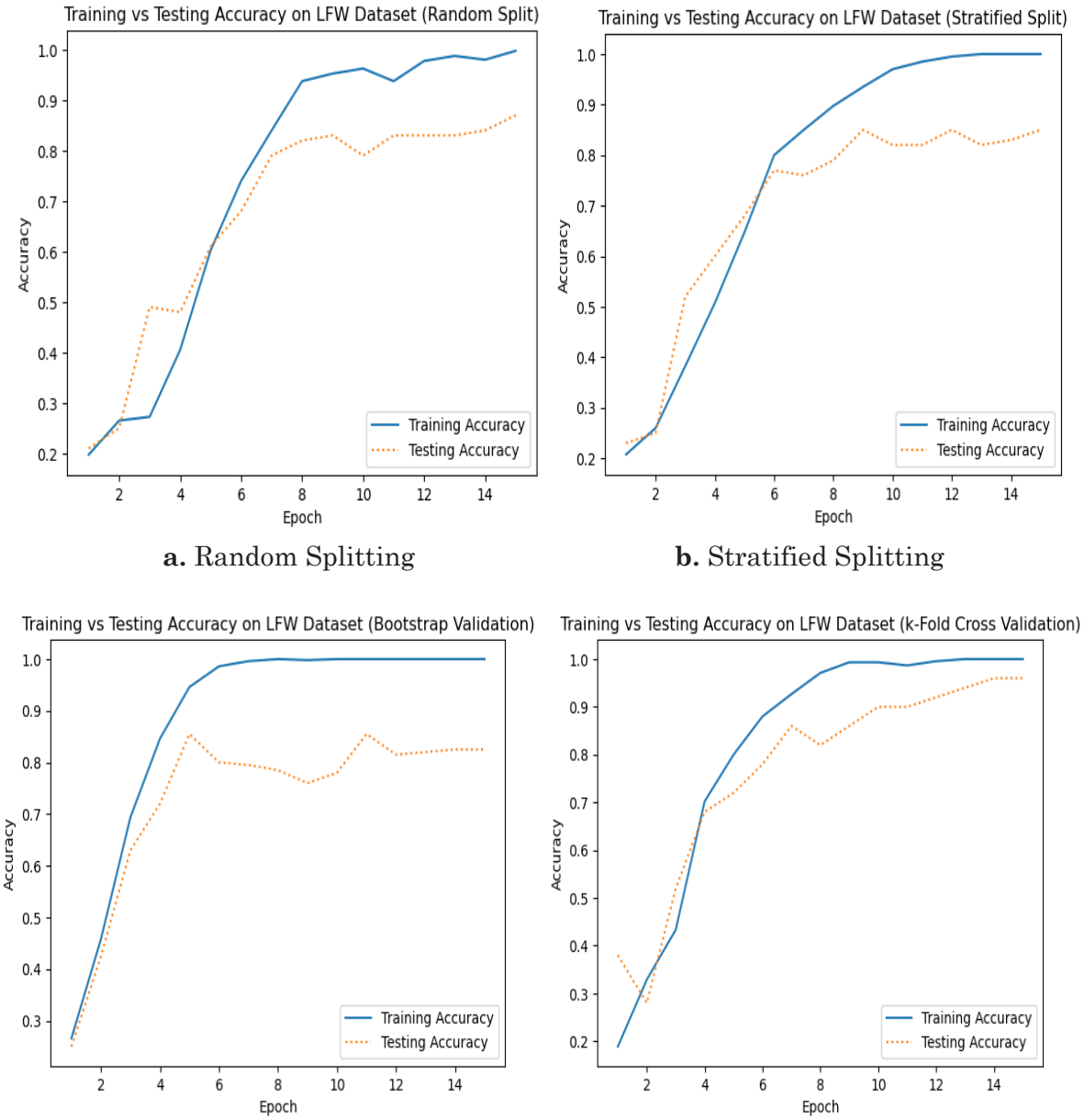
$$Var(Y) = E\left[\left(\hat{Y} - E(\hat{Y})\right)^2\right] \quad (14)$$

Models with lower variance are generally more stable and less affected by changes in the training data. In practice, strong model performance is characterized by high accuracy, low bias, and low variance.

The bias and variance of the model under each data partitioning strategy were computed using the bias-variance decomposition method with a 0-1 loss function proposed by Domingos (2000) and implemented using the bias_variance_decomp function from the MLxtend library (Raschka, 2018). The CNN model was trained five times on each training set, and predictions from the resulting five models were collected on the corresponding test instances and recorded. A fixed random seed (42) ensured reproducibility across all experiments. Bias was computed as the error incurred from the model's average prediction across repetitions, while variance was defined as the average deviation of the prediction of each test instance from the main prediction.

RESULTS AND DISCUSSION

The experiment was carried out using benchmark facial image data sets, and the results obtained from each data splitting technique were compared using accuracy, bias, and variance as evaluation metrics. Figures 3a to 3d show the visualization of the training and testing accuracy across each epoch during training of the CNN model on D1 under each data splitting technique.



a. Random Splitting **b.** Stratified Splitting

c. Bootstrap Validation d. k-Fold Cross-Validation

Figure 3. Training versus testing accuracy on D1.

Table 1 shows the summary of model accuracy metrics for each data splitting technique used in this study: random splitting, stratified splitting, k-fold cross-validation, and bootstrap validation.

Table 1. Model Accuracy Performance Comparison

Data Set	Sets	Accuracy			
		Random Splitting	Stratified Splitting	Bootstrap Validation	k-Fold Cross-Validation
D1	Training	0.9950	0.9925	0.9960	0.9980
	Testing	0.8700	0.8300	0.8250	**0.9000**
D2	Training	0.9900	0.9980	0.9900	0.9975
	Testing	0.8951	0.9125	0.8950	**0.9600**

Note. Bold values indicate the highest classification testing accuracy for each data set.

Table 2 shows that k-fold cross-validation has the highest mean test accuracy on D1 and D2. However, there is no sufficient evidence to suggest the difference between the accuracies under each data partitioning strategy is significant. To test for significant difference between the data partitioning strategies, a Friedman test and post hoc Conover test (Conover, 1999) corrected using the Holm method were conducted on the results of D1 and D2.

For D1 and D2, the Friedman test revealed significant differences among the data partitioning strategies under study at the 5% level of significance. The post hoc pairwise Conover comparison tests showed that k-fold cross-validation has more stable accuracy estimates than random splitting, stratified splitting, and bootstrap validation. The Conover test *p*-value results for pairwise comparison of data partitioning strategy across D1 and D2 are shown in Table 2.

Table 2. Conover Test *p*-Value Results for D1 and D2

Comparison	D1 *p*-Value	Significant? (D1)	D2 *p*-Value	Significant? (D2)
Random vs stratified	0.9012	No	0.9285	No
Random vs bootstrap	0.9000	No	0.9285	No
Random vs k-fold	0.021	Yes	0.0131	Yes
Stratified vs bootstrap	0.5124	No	0.4693	No
Stratified vs k-fold	0.045	Yes	0.0423	Yes
Bootstrap vs k-fold	0.0056	Yes	0.0041	Yes

Other performance metrics used to evaluate the model are bias and variance. The lower the bias and variance of the CNN model, the better the performance.

Table 3 shows the comparison of performance of the different data splitting techniques using bias and variance metrics.

Table 3. Bias and Variance Performance Comparison

Data Set	Bias				Variance			
	Random	Stratified	Bootstrap	k-fCV	Random	Stratified	Bootstrap	k-fCV
D1	0.1200	0.1500	0.1650	**0.1180**	0.0340	0.0660	**0.0090**	**0.0100**
D2	0.0750	0.1000	0.1111	**0.0025**	0.0500	0.0300	0.0284	**0.0190**

Note. Bold values indicate lowest bias and variance for each dataset. k-fCV = k-fold cross-validation.

Table 3 shows the bias and variance of the facial recognition model under the four different data splitting techniques. As observed from the table, k-fold cross-validation gave the least bias in both D1 and D2, and bootstrap validation produced the highest bias results in both D1 and D2.

The variance section shows that bootstrap validation has the least variance in D1, with k-fold cross-validation having a closer value. In D2, k-fold cross-validation gave the least variance when compared to other splitting techniques.

CONCLUSION

This study has shown that the data splitting technique employed in CNN-based facial recognition models has a significant impact on the estimation of their performance and generalization. It is often necessary to consider more than traditional methods like random splitting when building machine learning models, as they could provide more reliable and stable performance estimates. The results from this research also indicated that reliance on accuracy alone when evaluating a CNN-based facial recognition model is misleading. Other metrics like bias and variance are necessary to evaluate how the model will generalize to unseen data.

k-Fold cross-validation offered the most stable performance estimates of accuracy, bias, and variance among the splitting techniques examined in this study. However, it is noteworthy that it is computationally intensive. A hybrid model of cross-validation with other partitioning strategies like stratified splitting and random splitting could produce a more stable performance and potentially reduce computational costs.

This study, however, is not without limitation. Subject-independent data splitting was not enforced, and basic parameters were used in building the CNN models due to the small data set size. Future research could explore the impact of enforcing subject-independent data splitting, testing on larger data sets, and tuning hyperparameters. This would contribute to building more robust CNN-based facial recognition models.

ACKNOWLEDGMENTS

The authors acknowledge the following for their support:

i. Associates Programme, QLS Section, Abdus Salam International Centre for Theoretical Physics, Trieste, Italy

ii. Department of Statistics, University of Lagos, Nigeria

REFERENCES

Bischl, B., Mersmann, O., Trautmann, H., & Weihs, C. (2012). Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evolutionary Computation, 20*(2), 249–275. https://doi.org/10.1162/EVCO_a_00069

Camilo, L. M., Santos, M. C. D., Lima, K., & Martin, F. L. (2019). Improving data splitting for classification applications in spectrochemical analyses employing a random-mutation Kennard-Stone algorithm approach. *Bioinformatics, 35*(24), 5257–5263. https://doi.org/10.1093/bioinformatics/btz421

Conover, W. J. (1999). *Practical nonparametric statistics* (3rd ed.). Wiley.

Dantcheva, A., Elia, P., & Ross, A. (2016). What else does your biometric data reveal? A survey on soft biometrics. *IEEE Transactions on Information Forensics and Security, 11*(3), 441–467. https://doi.org/10.1109/TIFS.2015.2480381

Domingos, P. (2000). A unified bias-variance decomposition and its applications. In *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 231–238). Morgan Kaufmann.

Genc¸, B., & Tunc¸, H. (2019). Optimal training and test sets design for machine learning. *Turkish Journal of Electrical Engineering and Computer Sciences, 27*(2), 1534–1545. https://doi.org/10.3906/elk-1807-212

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). *Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments* [Data set]. University of Massachusetts Amherst. http://vis-www.cs.umass.edu/lfw/

Karkuzhali, S., Murugeshwari, R., & Umadevi, V. (2024). Facial emotion recognition and synthesis with convolutional neural networks. *International Journal of Computer Applications, 186*(11), 975–986. https://doi.org/10.5120/ijca2024923159

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)* (Vol. 2, pp. 1137–1143). Montreal, Canada.

Lopez, E., Gorla, G., Etxebarria-Elezgarai, J., Amigo, J. M., & Seifert, A. (2025). The importance of choosing a proper validation strategy in predictive models. Part 2: Recipes for (avoiding) overfitting. *Analytical Chimica Acta, 1275,* 344838. https://doi.org/10.1016/j.aca.2025.344838

Moss, H. B., Leslie, D. S., & Rayson, P. (2018). Using J-k fold cross validation to reduce variance when tuning NLP models. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 2978–2989). Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Nurhopipah, A., & Harjoko, A. (2018). Motion detection and face recognition for CCTV surveillance system. *Indonesian Journal of Computing and Cybernetic Systems (IJCCS), 12*(2), 107–118. https://doi.org/10.22146/ijccs.18198

Nurhopipah, A., & Uswatun H. (2020). Dataset splitting techniques comparison for face classification on CCTV images. *Indonesian Journal of Computing and Cybernetic Systems (IJCCS), 14*(4), 341–352. https://doi.org/10.22146/ijccs.58092

Ramezan, C. A., Warner, T. A., & Maxwell, A. E. (2019). Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. *Remote Sensing, 11*(2), 185. https://doi.org/10.3390/rs11020185

Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *Journal of Open Source Software, 3*(24), 638. https://doi.org/10.21105/joss.00638

Samaria, F., & Harter, A. (1994). *ORL Database of Faces* [Data set]. AT&T Laboratories Cambridge. https://cam-orl.co.uk/facedatabase.html

Schnaubelt, M. (2019). *A comparison of machine learning model validation schemes*

for non- stationary time series data (Technical report). ResearchGate. https://doi.org/10.13140/RG.2.2.29545.24168

Tantithamthavorn, C., McIntosh, S., Hassan, A. E., & Matsumoto, K. (2017). An empirical comparison of model validation techniques for defect prediction models. *IEEE Transactions on Software Engineering, 43*(1), 1–18. https://doi.org/10.1109/TSE.2016.2584050

Udomboso, C. G., Irmiya, E. S., Akinyemi, J. D., Ladoja, K. T., Akinseye, I. E., & Omoyajowo, A. C. (2024). A statistical learning model for number plate recognition for vehicular security. *University of Ibadan Journal of Science and Logics in ICT Research, 12*(1), 74–88.

Udomboso, C. G., Sigauke, C., & Adinya, I. (2025). *Fusion sampling validation in data partitioning for machine learning*. arXiv. https://www.arxiv.org/abs/2508.01325

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PLoS One, 14*(11), e0224365. https://doi.org/10.1371/journal.pone.0224365

Wang, F., Sahana, M., Pahlevanzadeh, B., Pal, S. C., Shit, P. K., Piran, M. J., Janizadeh, S., Band, S. S., & Mosavi, A. (2021). Applying different resampling strategies in machine learning models to predict head-cut gully erosion susceptibility. *Alexandria Engineering Journal, 60*(6), 5813–5829. https://doi.org/10.1016/j.aej.2021.04.026

Wang, J., & Li, Z. (2018). Research on face recognition based on CNN. *IOP Conference Series: Earth and Environmental Sciences, 170*(3), 032110. https://doi.org/10.1088/1755-1315/170/3/032110

Xu, Y., & Goodacre, R. (2018). On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing, 2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics Conference Series, 1168*(2), 022022. https://doi.org/10.1088/1742-6596/1168/2/022022