



Improved Stroke Classification Accuracy by Using Hybrid Inception and Xception Models

Desta Atikananda¹, Slamet Riyadi², Cahya Damarjati³, Annisa Divayu Andriyani⁴

^{1,2,3}Dept of Information Technology, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta, Indonesia

⁴Faculty of Electrical and Electronics Engineering Technology, Univeristi Malaysia Pahang Al-Sultan Abdullah, Malaysia

Abstract.

Purpose: The goal of this study is to use deep learning models on tabular data to create a good way to classify strokes. Stroke is still one of the most common causes of death and disability around the world, and finding it early is very important for getting better results from treatment. But there is a big problem with making reliable predictions because there is an imbalance between stroke and non-stroke data.

Methods/Study design/approach: The study uses the publicly available Stroke Prediction Dataset from Kaggle, which has 5,110 patient records and 12 risk factors. We suggest a hybrid deep learning architecture that combines Inception and Xception networks. It is meant to take advantage of multi-scale feature extraction and depthwise separable convolution. The Synthetic Minority Over-sampling Technique (SMOTE) is used during preprocessing to fix the problem of classes being out of balance. We divide the dataset into three parts: training, validation, and testing. Using accuracy, precision, recall, and F1-score, we can evaluate a model.

Result/Findings: The proposed hybrid model achieved the highest classification accuracy of 92.2%, outperforming the individual Inception (86.28%) and Xception (89.26%) models. This demonstrates that the use of SMOTE during preprocessing significantly improved class balance and overall prediction reliability.

Novelty/Originality/Value: This study uses a new hybrid method that combines two well-known CNN architectures, Inception and Xception, to classify tabular medical data. When these models are combined and applied with SMOTE, the classification performance improved significantly, demonstrating strong potential for implementation in automated stroke diagnosis systems.

Keywords: Stroke, Inception, Xception, Hybrid model, CNN, SMOTE, Tabular data

Received June 2025 / **Revised** November 2025 / **Accepted** December 2025

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



INTRODUCTION

A stroke is a medical emergency that occurs when blood flow to the brain is interrupted [1]. This condition can damage brain cells within minutes [2], and prompt treatment is crucial to minimize brain damage and other complications. There are two main types of stroke: ischemic stroke, caused by a blockage in a blood vessel and accounting for approximately 70–85% of all stroke cases [3], and hemorrhagic stroke, which occurs when a blood vessel in the brain ruptures [4], [5].

With the advancement of artificial intelligence technology, particularly deep learning, methods such as Convolutional Neural Networks (CNNs) have shown promising performance in detecting various types of strokes through medical images such as CT scans and MRIs [6], [7]. CNN models like Inception and Xception have proven effective in extracting features even from low-quality medical images [8], [9]. Inception V3 utilizes multiscale convolution and dimension reduction techniques to enhance feature extraction efficiency [10]. while Xception relies on depthwise separable convolution to accelerate the training process while maintaining high accuracy [11], [12].

However, image-based approaches are not the only path that can be used in stroke detection. Approaches based on tabular or structured data also hold significant potential [13]. Especially when patient clinical information is available in numerical or categorical formats. Nevertheless, the application of CNNs to tabular data such as electronic health records remain relatively limited. This is due to the fundamental characteristics of CNNs, which are designed to extract spatial features from visual data, making them inherently unsuitable for the non-spatial structure of tabular data. States that the performance of CNN and

²*Corresponding author.

Email addresses: riyadi@umy.ac.id (Riyadi)

DOI: [10.15294/sji.v8i1.29023](https://doi.org/10.15294/sji.v8i1.29023)

deep learning in general on tabular data still lags behind decision tree-based models such as XGBoost, unless architectural adaptations or data representation transformations are made [14]. The PLORAS study (2024) also shows that integrating tabular data and images in CNN can improve the accuracy of post-stroke classification. However, the direct application of CNN to pure tabular data is still rarely explored and has not shown consistent advantages over conventional models [15].

In addition to CNN, the latest Transformer-based approaches such as TabTransformer and FT-Transformer have shown high performance on tabular data due to their ability to capture contextual interactions between features without the need for spatial transformations [16], [17]. However, the specific application of these models for stroke classification based on tabular data is rarely reported in the literature and is not the main focus of this study.

But there is a big problem with class imbalance in tabular data-based stroke classification: there are a lot more non-stroke cases than stroke cases [18], [19], [20]. This can make the prediction model biased, which means it won't be very sensitive to real stroke cases[21]. The Synthetic Minority Over-sampling Technique (SMOTE) has been widely used in medical research to make class distributions more fair. This method makes fake samples for minority classes[22]. A wide range of machine learning models have been shown to work better with this method [23].

Oversampling strategies have made some traditional machine learning models better. These include Random Forest, LightGBM, and KNN [21], [24], [25]. In the field of tabular stroke classification, however, the use of powerful CNN architectures, especially in hybrid form, is still not well understood. This makes it hard to find out if modern CNNs [20], [26], [27], which were originally made for image data, can be adapted and used with structured health records [28], [29], [30].

The simultaneous application of two CNN architectures in a single hybrid model also has consequences in terms of increased computational complexity and resource requirements[31]. Therefore, an evaluation of the trade-off between accuracy and computational efficiency is necessary to determine whether the hybrid approach truly provides significant advantages over a single model [32].

Based on the background and research gaps, this study aims to develop a hybrid deep learning model that combines the Inception and Xception architectures for stroke classification based on tabular data. To address class imbalance, the SMOTE technique is used in the preprocessing stage before model training. Performance evaluation is conducted using standard metrics such as accuracy, precision, recall, and F1-score. This research is expected to contribute to exploring the potential of CNNs for tabular data while evaluating the effectiveness of the hybrid approach compared to single architectures using a public stroke dataset.

METHODS

Some of the stages of the research conducted are depicted in Figure 1.

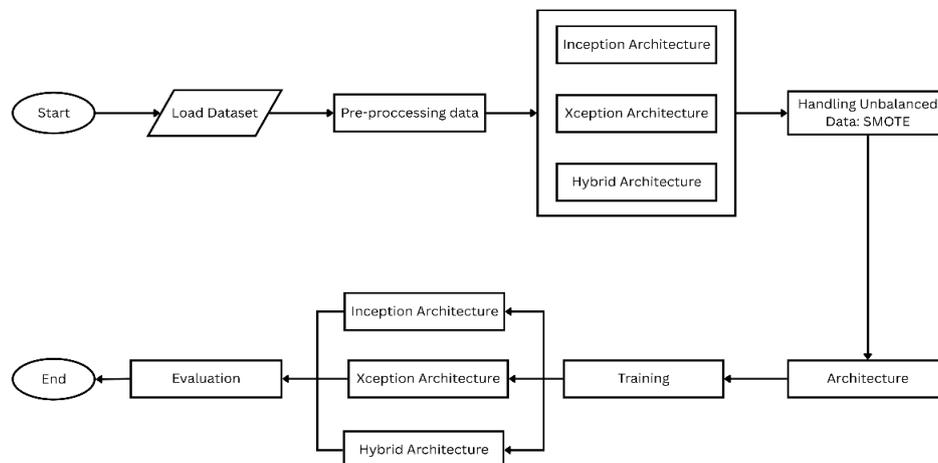


Figure 1. Flowchart analysis

Figure 1 shows the stages of research, starting with the process of loading the stroke dataset, followed by data pre-processing and handling data imbalance using SMOTE. The CNN model used consists of three architectures: Inception, Xception, and a combination of the two in the form of a hybrid architecture. Each architecture was trained separately, then evaluated based on classification metrics such as accuracy, precision, recall, and F1-score.

Data Collection

This experiment utilized the Stroke Prediction Dataset from the Kaggle platform, which contains data from 5,110 patients[33]. Each entry's target label consists of 12 features: gender, age, hypertension, heart disease, marital status, occupation type, habitation type, average glucose level, body mass index (BMI), smoking status, and stroke status (1: has a history of stroke, 0: has no history of stroke). Due to space limitations, only 3 records are shown in Table 1. A more complete sample is provided in Stroke Prediction Dataset | Kaggle [34]

Table 1. Dataset stroke

ID	Gender	Age	Hyper-tension	Heart disease	Ever married	Work type	Residence type	Avg Glucose level	BMI	Smoking status	Stroke
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	1
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1

Pre-Processing Data

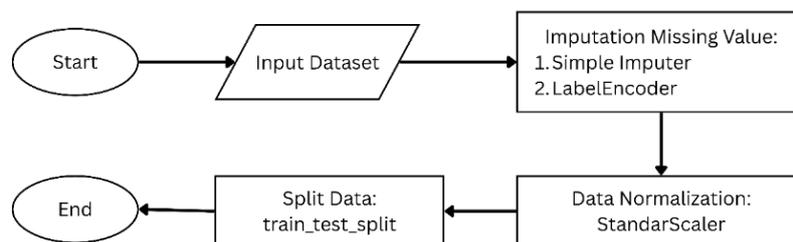


Figure 2. Flowchart preprocessing

The dataset is divided into 70% for training, 20% for validation, and 10% for testing. The preprocessing steps applied to the dataset are illustrated in Figure 2. This division is done in a stratified manner to ensure that the distribution of stroke and non-stroke classes remains balanced in each subset as shown in Table 2.

70% for training: This portion ensures that the model receives a sufficiently large and diverse sample to learn the underlying patterns and feature relationships comprehensively. A larger training set strengthens the model's ability to generalize to new, unseen data [35].

20% for validation: The validation set is used during model development to perform hyperparameter tuning, select the optimal architecture, and prevent overfitting. This proportion balances stability in performance estimation without excessively reducing training data [36].

10% for testing: This subset is held back to objectively assess the model's final performance on completely unseen data. The 10% split is commonly adopted in the literature to simulate real-world application without overconsuming training resources [37].

Table 2. Splitting data

Train	Valid	Test
70 %	20 %	10 %
3577	1073	459

Inception

Utilizes parallel paths of 1×1 , 3×3 , 5×5 and max pooling convolutions. This block is designed to extract features at multiple scales [38]. Details of the Inception architecture are provided in Figure 3.

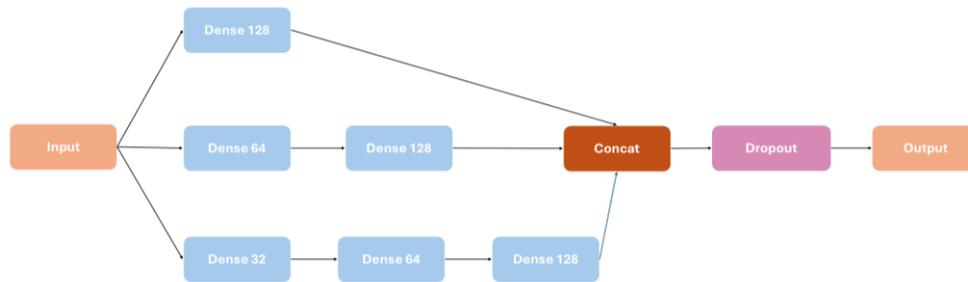


Figure 3. Inception architecture

This study uses a deep learning approach by comparing three different neural network model architectures to predict stroke events based on structured data as shown in Table 3. Each architecture is designed to explore the network's ability to efficiently extract and combine features from various transformation paths. The first model is a multi-path dense network. This architecture consists of three parallel paths: the first path has one Dense layer with 128 neurons, the second path consists of two multi-level Dense layers with 64 and 128 neurons, respectively, while the third path has three consecutive Dense layers with 32, 64, and 128 neurons. These three paths are combined using the Concatenate operation, followed by Dropout to prevent overfitting, before finally being passed to the output layer for classification.

Xception

Adopt depthwise separable convolution with skip connection, use parallel dense layer and L2 regularization to enrich feature learning [39]. Details of the Exception architecture are provided in Figure 4.

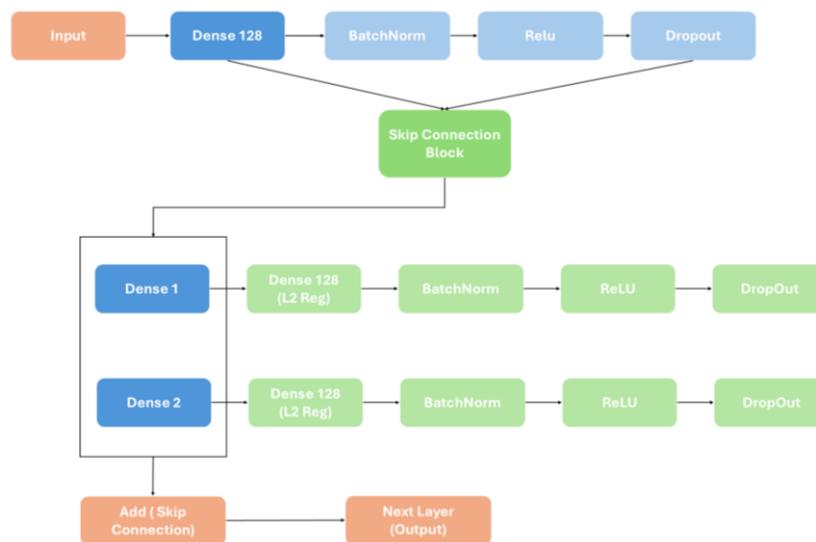


Figure 4. Xception architecture

The second model uses a residual approach with skip connections that adopts the basic principles of Residual Network (ResNet). This model aims to maintain the flow of information from the initial input and overcome the vanishing gradient problem. The architecture starts from the input, which is processed through a Dense layer with 128 neurons, then stabilized with Batch Normalization, activated using the ReLU function, and given Dropout. The input is then passed to two parallel Dense layers that form a residual block. The result of this block is then added (Add) to the initial input as a skip connection. The output of this addition is passed to the output Dense layer for final classification. Adopt depthwise separable convolution with skip connection, use parallel dense layer and L2 regularization to enrich feature learning[39].

Hybrid Inception dan Xception

It is a combination of Inception and Xception in two parallel paths. The outputs of both paths are concatenated, followed by Global Average Pooling, Dense, and Dropout before heading to the output layer.

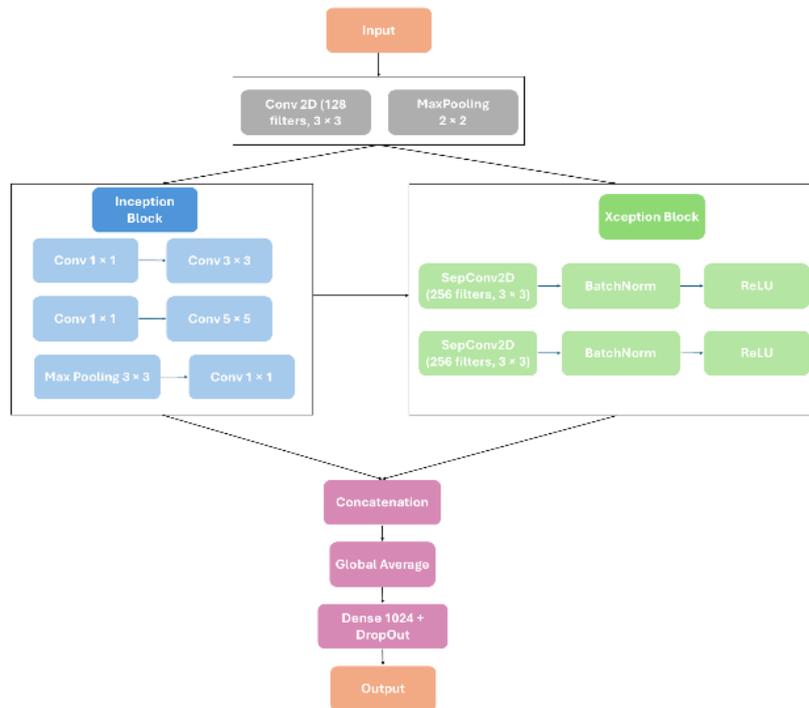


Figure 5. Hybrid architecture

The third model is a hybrid architecture that combines principles from two popular CNN architectures, namely Inception and Xception as shown in Figure 5. Although originally developed for image data, both have been modified to be applicable to tabular or structured data. The data is first processed through Conv2D and MaxPooling layers, then divided into two main blocks. The Inception block consists of Conv2D paths with varying kernel sizes (1x1, 3x3, and 5x5) and MaxPooling, the results of which are then combined. Meanwhile, the Xception block utilizes depthwise separable convolution to extract features with high efficiency, accompanied by Batch Normalization and ReLU activation. These two blocks are then combined and processed through GlobalAveragePooling, then passed to a Dense layer with 1024 neurons, Dropout, and finally to the output layer.

Model Training

The model was compiled using the ReLU activation function, binary_crossentropy loss function, and Adam's optimizer. The training process includes the use of EarlyStopping and ModelCheckpoint to prevent overfitting and save the best model. Hyperparameters, including learning rate, batch size, and epoch count, were evaluated across multiple setups. The training and validation procedure is illustrated graphically. The performance comparison among Inception, Xception, and Hybrid models is presents in Table 3.

Table 3. Inception, Xception, and Hybrid comparison

Model	Optimizer	Batch Size	Epoch
Inception (83)	Adam	512	100
Inception (86)	Adam	512	100
Xception (86)	Adam	512	100
Xception (89)	Adam	512	100
Hybrid (87)	Adam	512	100
Hybrid (92)	Adam	512	100

Evaluation Metrics

The 'accuracy' standard is used as an evaluation metric to look at several work-related veins. In addition to the above stresses, the F1 Score is also evaluated [40]. Accuracy is a measure of how many predictions are correct from the tested data as a whole [41]. This metric is often used as an early indicator of model performance as it includes correct predictions for both positive and negative classes [42]. Equation 1 shows the proportion of correctly instances among all tested data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

Sensitivity, also called Recall, tells you how well the model can find positive data from all the data that is genuinely positive. This statistic is useful for finding diseases or fraud, for example, because missing positive cases can have catastrophic effects [43]. Equation 2 measures the ability of the model to correctly identify positive cases.

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

Specificity tells you how well the model can locate negative data among all the true negative data. This score demonstrates how successfully the model discovers negative data without mistakenly labeling them as positive [44]. Equation 3 evaluates the ability of the model to correctly identify negative cases.

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (3)$$

Precision is the ratio of correct positive predictions to the total number of positive forecasts. It shows how accurate the model's positive predictions are. A high accuracy means that the model doesn't often make incorrect positive predictions [45]. Equation 4 measures the proportion of correctly predicted positive cases among all predicted positives, indicating how reliable the model's positive predictions are.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (4)$$

The F1-Score is a single number that combines Precision and Recall by finding their harmonic mean. This score balances the model's ability to find positive data (Recall) with its ability to make accurate positive predictions (Precision)[46]. Equation 5 represent the harmonic mean of Precision and Sensitivity (Recall), balancing the model's ability to detect positive cases with the accuracy of its positive predictions.

$$F1 - Score = 2 \times \frac{Sensitivity \times Precision}{Sensitivity + Precision} \times 100\% \quad (5)$$

Evaluation was conducted before and after SMOTE. Before SMOTE, the models tended to fail to detect stroke classes. After SMOTE, recall and F1-score improved significantly especially in the Hybrid model as shown in Table 4 and Figure 6.

Table 4. Before and after SMOTE comparison

Metric	Class	Before SMOTE			After SMOTE		
		Inception	Xception	Hybrid	Inception	Xception	Hybrid
Accuracy		95.15	95.15	95.15	86.26	89.26	92.2
Recall	1	1.00	0.84	1.00	0.85	0.84	0.91
	2	0.00	0.94	0.00	0.88	0.94	0.93
Precision	1	0.95	0.93	0.95	0.90	0.93	0.93
	2	0.00	0.86	0.00	0.82	0.86	0.92
F1-Score	1	0.98	0.98	0.98	0.87	0.89	0.92
	2	0.00	0.90	0.00	0.85	0.90	0.92

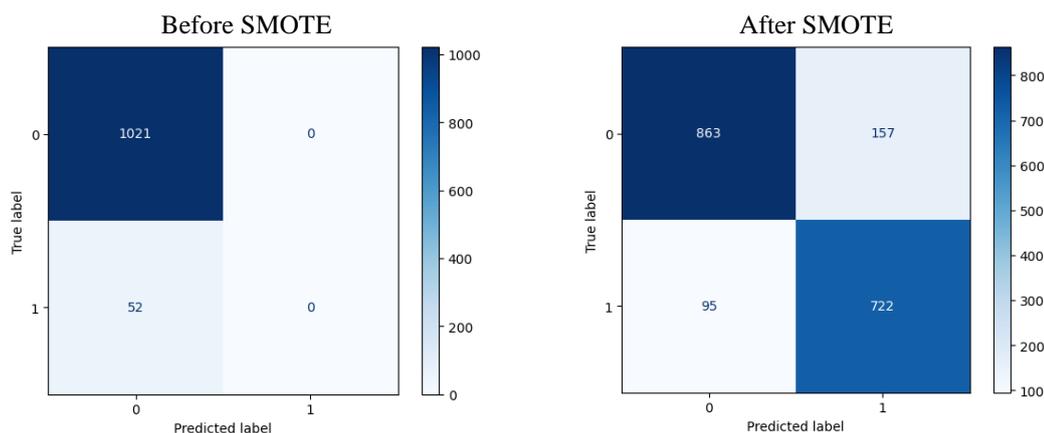


Figure 6. Comparison of confusion matrix before and after SMOTE

RESULT AND DISCUSSION

For the purpose of classifying stroke data using tabular data, this study compares the effectiveness of three convolutional models: Inception, Xception, and Hybrid Inception-Xception. The confusion matrix and the four cornerstone evaluation metrics—recall, accuracy, precision, and F1-score—form the basis of the evaluation. After SMOTE equilibration, all models were tested on the same data. The stability of the training set and the model's generalizability were evaluated using accuracy and loss graphs.

As illustrated in Figure 8, the Inception model yielded 863 correct negatives and 722 correct positives, with 157 incorrect positives and 95 incorrect negatives. The confusion matrix of the Inception model, shown in Figure 7. The results demonstrate that the model is sensitive to stroke instances, but it still makes a lot of mistakes, particularly in the negative category. Figure 9 displays a consistent training pattern for the accuracy and loss graphs. Overfitting is not shown by validation loss, which follows the drop in training loss. The fact that the validation accuracy is almost the same as the training accuracy further demonstrates the model's data-learning and extrapolation capabilities.

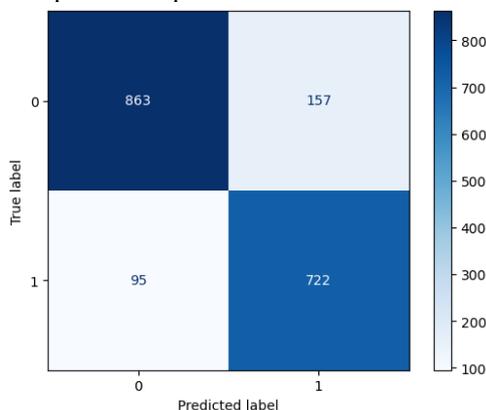


Figure 7. Confusion matrix inception

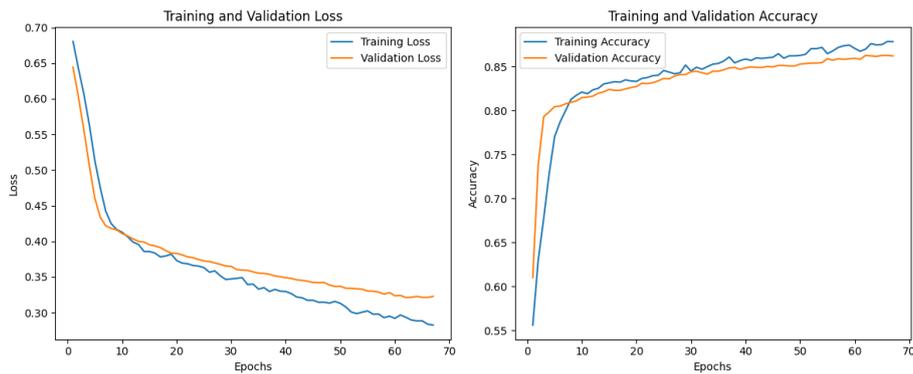


Figure 8. Training and validation of loss and accuracy graphs inception

The Xception model correctly classified 861 non-stroke data and 961 stroke data. The number of false positives and false negatives was 159 and 60, respectively (Figure 10). This model showed increased sensitivity to the positive class compared to Inception. As shown in Figure 11, the training loss decreased sharply from the beginning, and although the validation loss fluctuated, it stabilized after the 60th epoch. The validation accuracy increased significantly and was close to the training accuracy, indicating that the model learned well over time.

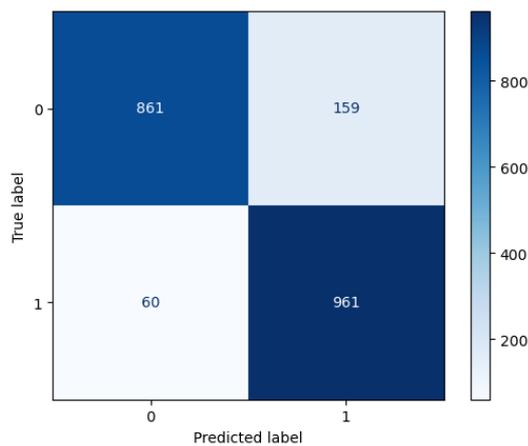


Figure 9. Confusion matrix Xception

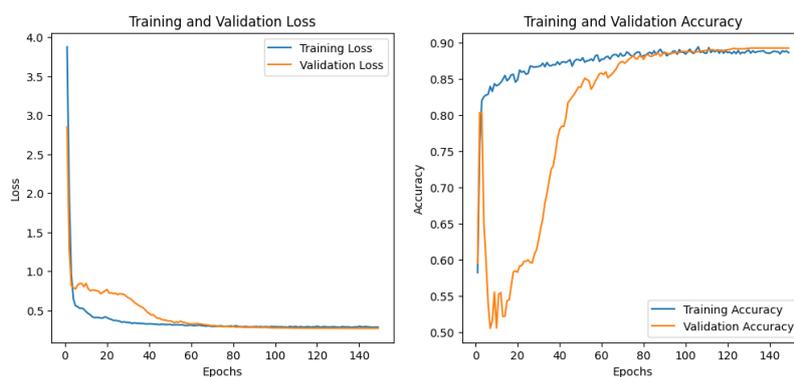


Figure 10. Training and validation of loss and accuracy graphs Xception

The hybrid model showed the best performance with 932 true negatives and 950 true positives. The number of misclassifications was only 88 false positives and 71 false negatives at Figure 12. This makes the Hybrid model the most reliable approach in this study. Figure 13 shows that both training and validation loss decreased consistently and converged. Training and validation accuracies also increased steadily without

sharp fluctuations, with validation accuracy exceeding 92%. This indicates that the model not only learns efficiently, but also has high generalization ability.

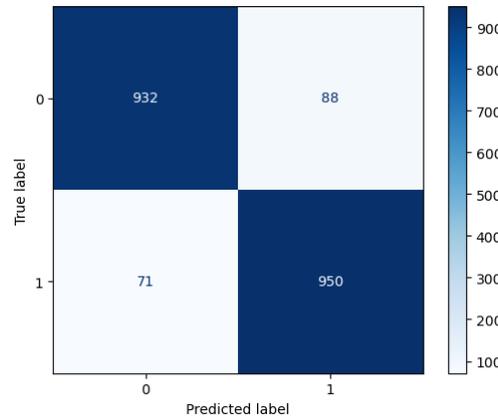


Figure 11. Confusion matrix Hybrid Inception-Xception

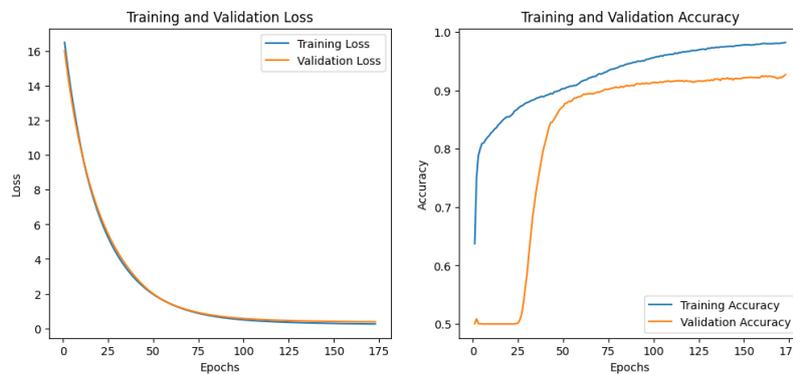


Figure 12. Training and validation of loss and accuracy graphs Hybrid Inception Xception

To compare the three models, Table 4 presents the key evaluation metrics. The Hybrid model excels in almost all metrics, with 92.2% accuracy and balanced precision, recall, and F1-score values above 0.92 as shown in Table 2. The Xception model comes in second, and Inception last, mainly due to lower recall and F1-score values for class 1 (stroke).

Table 5. Accuracy comparison of Inception, Xception, and Hybrid Inception-Xception

Model	Kelas	Precision	Recall	F1-Score	Accuracy
Inception	0	0.90	0.85	0.87	86.28%
	1	0.82	0.88	0.85	
Xception	0	0.93	0.84	0.89	89.26%
	1	0.86	0.94	0.90	
Hybrid	0	0.93	0.91	0.92	92.2%
	1	0.92	0.93	0.92	

These results show that the use of a hybrid architecture provides significant advantages, especially in terms of the balance between precision and recall as shown in Table 5. This is crucial in medical contexts such as stroke classification, where detection of positive cases is more important than just high accuracy. In addition, the application of SMOTE was shown to increase sensitivity to minority classes, drastically reducing false negative values. Compared to previous approaches using only one architecture or traditional models, the Hybrid model in this study shows great potential to be applied in tabular data-based automated medical diagnosis systems.

CONCLUSION

This research proves that the Hybrid architecture deep learning model, which combines the strengths of Inception and Xception, provides the best performance in stroke data classification based on tabular data that has been balanced using SMOTE. Compared to the Inception and Xception models separately, the Hybrid model showed the highest accuracy of 92.2% with balanced precision and recall values and the least number of misclassifications. These results show that the combined approach of convolutional architecture can improve the performance of stroke case detection, especially in the context of imbalanced data. The stable training process without overfitting and high generalization ability make this model very potential to be applied in automated medical detection systems, especially in real environments that demand accuracy and efficiency. This research makes a real contribution to the development of decision support systems in the health sector with a deep learning approach based on tabular data.

REFERENCES

- [1] A. P. Akerman *et al.*, “External validation of artificial intelligence for detection of heart failure with preserved ejection fraction,” *Nat. Commun.*, vol. 16, no. 1, p. 2915, Mar. 2025, doi: 10.1038/s41467-025-58283-7.
- [2] C. D. Kulathilake, J. Udupihille, S. P. Abeysundara, and A. Senoo, “Deep learning-driven multi-class classification of brain strokes using computed tomography: A step towards enhanced diagnostic precision,” *Eur. J. Radiol.*, vol. 187, p. 112109, Jun. 2025, doi: 10.1016/j.ejrad.2025.112109.
- [3] B. Borsos, C. G. Allaart, and A. van Halteren, “Predicting stroke outcome: A case for multimodal deep learning methods with tabular and CT Perfusion data,” *Artif. Intell. Med.*, vol. 147, p. 102719, Jan. 2024, doi: 10.1016/j.artmed.2023.102719.
- [4] Q. Q. Thabit, A. I. Dawoo, and B. A. Issa, “Early Prediction of Stroke Based on Deep and Machine Learning by Applying Medical Imaging and Tabular Data,” *Mathematical Modelling of Engineering Problems*, vol. 11, no. 12, pp. 3499–3508, Dec. 2024, doi: 10.18280/mmep.111228.
- [5] J. K. Park, S. B. Mun, Y. J. Kim, and K. G. Kim, “Deep learning-based classification of speech disorder in stroke and hearing impairment,” *PLoS One*, vol. 20, no. 5, p. e0315286, May 2025, doi: 10.1371/journal.pone.0315286.
- [6] R. Mena, E. Pelaez, F. Loayza, A. Macas, and H. Franco-Maldonado, “An artificial intelligence approach for segmenting and classifying brain lesions caused by stroke,” *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, vol. 11, no. 7, pp. 2736–2747, Jan. 2024, doi: 10.1080/21681163.2023.2264410.
- [7] N. Alamatsaz, L. Tabatabaei, M. Yazdchi, H. Payan, N. Alamatsaz, and F. Nasimi, “A lightweight hybrid CNN-LSTM explainable model for ECG-based arrhythmia detection,” *Biomed. Signal Process. Control*, vol. 90, p. 105884, Apr. 2024, doi: 10.1016/j.bspc.2023.105884.
- [8] N. H. Putri, J. Jasril, M. Irsyad, S. Agustian, and F. Yanto, “Klasifikasi Citra Stroke Menggunakan Augmentasi dan Convolutional Neural Network EfficientNet-B0,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 2, p. 650, Apr. 2023, doi: 10.30865/mib.v7i2.5981.
- [9] J. T. Marbun, Seniman, and U. Andayani, “Classification of stroke disease using convolutional neural network,” *J. Phys. Conf. Ser.*, vol. 978, p. 012092, Mar. 2018, doi: 10.1088/1742-6596/978/1/012092.
- [10] J. A. Gliner, G. A. Morgan, and N. L. Leech, “Measurement reliability and validity,” *Research Methods in Applied Settings*, pp. 319–338, Feb. 2021, doi: 10.4324/9781410605337-29.
- [11] K. Avery *et al.*, “Fatigue Behavior of Stainless Steel Sheet Specimens at Extremely High Temperatures,” *SAE International Journal of Materials and Manufacturing*, vol. 07, no. 3, pp. 2014-01-0975, Apr. 2014, doi: 10.4271/2014-01-0975.
- [12] K. Shaheed, Q. Abbas, A. Hussain, and I. Qureshi, “Optimized Xception Learning Model and XgBoost Classifier for Detection of Multiclass Chest Disease from X-ray Images,” *Diagnostics*, vol. 13, no. 15, p. 2583, Aug. 2023, doi: 10.3390/diagnostics13152583.

- [13] A. White, M. Saranti, A. d'Avila Garcez, T. M. H. Hope, C. J. Price, and H. Bowman, "Predicting recovery following stroke: Deep learning, multimodal data and feature selection using explainable AI," *Neuroimage Clin.*, vol. 43, p. 103638, 2024, doi: 10.1016/j.nicl.2024.103638.
- [14] V. Borisov, T. Leemann, K. Sebler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep Neural Networks and Tabular Data: A Survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7499–7519, Jun. 2024, doi: 10.1109/TNNLS.2022.3229161.
- [15] A. White, M. Saranti, A. d'Avila Garcez, T. M. H. Hope, C. J. Price, and H. Bowman, "Predicting recovery following stroke: Deep learning, multimodal data and feature selection using explainable AI," *Neuroimage Clin.*, vol. 43, p. 103638, 2024, doi: 10.1016/j.nicl.2024.103638.
- [16] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," Dec. 2020.
- [17] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting Deep Learning Models for Tabular Data," Oct. 2023.
- [18] M. El-Melegy *et al.*, "Prostate Cancer Diagnosis via Visual Representation of Tabular Data and Deep Transfer Learning," *Bioengineering*, vol. 11, no. 7, p. 635, Jun. 2024, doi: 10.3390/bioengineering11070635.
- [19] M. Artzi *et al.*, "Classification of Pediatric Posterior Fossa Tumors Using Convolutional Neural Network and Tabular Data," *IEEE Access*, vol. 9, pp. 91966–91973, 2021, doi: 10.1109/ACCESS.2021.3085771.
- [20] L. Buturović and D. Miljković, "A novel method for classification of tabular data using convolutional neural networks," May 03, 2020. doi: 10.1101/2020.05.02.074203.
- [21] K. Swain *et al.*, "Enhancing Stroke Prediction Using LightGBM With SMOTE-ENN and Fine-Tuning: A Comprehensive Analysis," 2024, doi: 10.7759/s44389-024.
- [22] S. Riyadi, A. M. Masyhur, and A. D. Andriyani, "Deep Learning for Hate Speech Intensity Analysis: DistilBERT Classification Algorithm," in *2024 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, IEEE, Jul. 2024, pp. 1–6. doi: 10.1109/ISIEA61920.2024.10607351.
- [23] H. M. Merdas, "Elastic Net – MLP – SMOTE (EMS)-Based Model for Enhancing Stroke Prediction," *Medinformatics*, vol. 1, no. 2, pp. 73–78, Apr. 2024, doi: 10.47852/bonviewMEDIN42022470.
- [24] F. Yagin, I. Cicek, and Z. Kucukakcali, "Classification of stroke with gradient boosting tree using smote-based oversampling method," *Medicine Science / International Medical Journal*, vol. 10, no. 4, p. 1510, 2021, doi: 10.5455/medscience.2021.09.322.
- [25] O. J. Abiodun and A. I. Wreford, "Stroke Prediction Using Smote for Data Balancing, XGBoost and KNN Ensemble Algorithms," *Journal of Applied Physical Science International*, pp. 42–53, Aug. 2023, doi: 10.56557/japsi/2023/v15i18349.
- [26] S. K. UmaMaheswaran, F. Ahmad, R. Hegde, A. M. Alwakeel, and S. Rameem Zahra, "Enhanced non-contrast computed tomography images for early acute stroke detection using machine learning approach," *Expert Syst. Appl.*, vol. 240, p. 122559, Apr. 2024, doi: 10.1016/j.eswa.2023.122559.
- [27] A. Sharma and D. Kumar, "Classification with 2-D convolutional neural networks for breast cancer diagnosis," *Sci. Rep.*, vol. 12, no. 1, p. 21857, Dec. 2022, doi: 10.1038/s41598-022-26378-6.
- [28] U. Islam, G. Mehmood, A. A. Al-Atawi, F. Khan, H. S. Alwageed, and L. Cascone, "NeuroHealth guardian: A novel hybrid approach for precision brain stroke prediction and healthcare analytics," *J. Neurosci. Methods*, vol. 409, p. 110210, Sep. 2024, doi: 10.1016/j.jneumeth.2024.110210.
- [29] J. Höhn *et al.*, "Combining CNN-based histologic whole slide image analysis and patient data to improve skin cancer classification," *Eur. J. Cancer*, vol. 149, pp. 94–101, May 2021, doi: 10.1016/j.ejca.2021.02.032.
- [30] A. D. Kulkarni, "Fuzzy Convolution Neural Networks for Tabular Data Classification," *IEEE Access*, vol. 12, pp. 151846–151855, 2024, doi: 10.1109/ACCESS.2024.3479882.

- [31] W. Gamaleldin, O. Attayyib, M. M. Alnfai, F. A. Alotaibi, and R. Ming, "A hybrid model based on CNN-LSTM for assessing the risk of increasing claims in insurance companies," *PeerJ Comput. Sci.*, vol. 11, p. e2830, Apr. 2025, doi: 10.7717/peerj-cs.2830.
- [32] Y. Ioannou *et al.*, "Decision Forests, Convolutional Networks and the Models in-Between," Mar. 2016.
- [33] "Stroke Prediction: Effect of Data Leakage | SMOTE." [Online]. Available: <https://kaggle.com/code/tanmay111999/stroke-prediction-effect-of-data-leakage-smote>
- [34] F. Soriano, "Stroke Prediction Dataset." [Online]. Available: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>
- [35] M. Davidian, A. Lahav, B.-Z. Joshua, O. Wand, Y. Lurie, and S. Mark, "Exploring the Interplay of Dataset Size and Imbalance on CNN Performance in Healthcare: Using X-rays to Identify COVID-19 Patients," *Diagnostics*, vol. 14, no. 16, p. 1727, Aug. 2024, doi: 10.3390/diagnostics14161727.
- [36] C. Renggli, L. Rimanic, N. M. Gürel, B. Karlaš, W. Wu, and C. Zhang, "A Data Quality-Driven View of MLOps," Feb. 2021.
- [37] M. N. Chan and T. Tint, "A Review on Advanced Detection Methods in Vehicle Traffic Scenes," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, IEEE, Jan. 2021, pp. 642–649. doi: 10.1109/ICICT50816.2021.9358791.
- [38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.
- [39] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jul. 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.
- [40] W. Seo, N. Kim, C. Park, and S.-M. Park, "Deep Learning Approach for Detecting Work-Related Stress Using Multimodal Signals," *IEEE Sens. J.*, vol. 22, no. 12, pp. 11892–11902, Jun. 2022, doi: 10.1109/JSEN.2022.3170915.
- [41] H. A. Mengash, "Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems," *IEEE Access*, vol. 8, pp. 55462–55470, 2020, doi: 10.1109/ACCESS.2020.2981905.
- [42] S. Riyadi, A. Divayu Andriyani, and S. Noraini Sulaiman, "Improving Hate Speech Detection Using Double-Layers Hybrid CNN-RNN Model on Imbalanced Dataset," *IEEE Access*, vol. 12, pp. 159660–159668, 2024, doi: 10.1109/ACCESS.2024.3487433.
- [43] S. Riyadi, A. D. Andriyani, and A. M. Masyhur, "Improving Hate Speech Detection Accuracy Using Hybrid CNN-RNN and Random Oversampling Techniques," in *2024 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, IEEE, Jul. 2024, pp. 1–5. doi: 10.1109/ISIEA61920.2024.10607232.
- [44] T. F. Monaghan *et al.*, "Foundational Statistical Principles in Medical Research: Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value," *Medicina (B Aires)*, vol. 57, no. 5, p. 503, May 2021, doi: 10.3390/medicina57050503.
- [45] S. Riyadi, A. D. Andriyani, A. M. Masyhur, C. Damarjati, and M. I. Solihin, "Detection of Indonesian Hate Speech on Twitter Using Hybrid CNN-RNN," in *2023 International Conference on Information Technology and Computing (ICITCOM)*, IEEE, Dec. 2023, pp. 352–356. doi: 10.1109/ICITCOM60176.2023.10442041.
- [46] R. Diallo, C. Edalo, and O. O. Awe, "Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score," 2025, pp. 283–312. doi: 10.1007/978-3-031-72215-8_12.