Smart Brain CT Scan Screening System for Automated Stroke Detection Using Computer Vision and U-Net Model

Rhea M. Bete[1], Francis John Saul M. Tin-ao[2], James Earl D. Cubillas[3]
[1]BSIT Student, [2]BSIT Student, [3]Instructor I.
[1]rhea.betee@carsu.edu.ph, [2]francisjohn.tinao@carsu.edu.ph, [3]jdcubillas@carsu.edu.ph

Abstract: *Cerebrovascular Accident (CVA), commonly known as stroke, Stroke remains a leading cause of death and disability, particularly in low-resource settings like the Philippines. Manual diagnosis of stroke through CT scans faces challenges such as delayed processing, limited access to specialists, and human error, often resulting in late interventions. This study introduces a Smart Brain CT Scan Screening System designed to automate the detection of ischemic and hemorrhagic strokes. The system employs computer vision techniques and a U-Net-based deep learning model trained on over 2,455 CT images. After 1,000 training epochs, the model achieved satisfactory performance metrics, including an Intersection over Union (IoU) score of 1.0000 and a minimal validation loss of 2.82×10^{-8}. By minimizing diagnostic delays and reducing reliance on radiological expertise, the system enhances early detection and supports timely medical intervention. This is especially valuable in underserved regions. Recommendations for further work include expanding the dataset, applying advanced augmentation techniques, and conducting broader clinical validations. Overall, this study showcases the potential of artificial intelligence to revolutionize medical imaging and improve stroke care in resource-constrained environments.*

Keywords: Cerebrovascular accident (CVA), Computer vision, CT Scan, Hounsfield Unit (HU), UNet Convolutional Neural Networks

1. INTRODUCTION

Cerebrovascular accident (CVA), or stroke, is a major health concern in the Philippines, ranking as the second leading cause of death and the primary cause of long-term disability. From 2010 to 2020, the country recorded an average of 63,804 stroke-related deaths annually, rising to 68,180 in 2021—of which 34% died without receiving medical attention [1]. Barriers such as limited access to CT scan facilities, delayed hospital admission, high treatment costs, and a shortage of trained specialists contribute to the country's high stroke mortality rate [2][3].

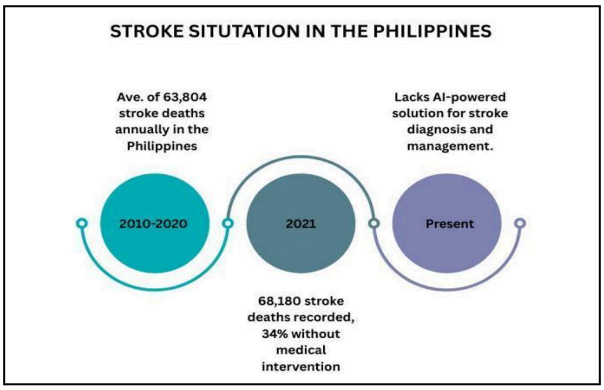

Fig. 1. Stroke Situation & AI Gaps in Philippine Healthcare. The Philippines recorded an average of 63,804 stroke-related deaths annually from 2010 to 2020, increasing to 68,180 in 2021. Alarmingly, over 34% of these cases lacked the appropriate medical intervention. Despite the high mortality rate, AI-powered tools for stroke diagnosis and care remain largely unavailable from the country's healthcare infrastructure.

Timely diagnosis is essential in stroke management, especially to differentiate between ischemic and hemorrhagic types. CT scans, widely used due to their speed and cost-effectiveness, are crucial in this regard [4]. However, manual interpretation by neuroradiologists can take up to 48 hours and involves time-consuming segmentation processes[5]. This delay poses a risk, as stroke treatment is most effective within the first six hours of onset [6].

The critical shortage of neurologists and radiologists in the country—one neurologist per 218,000 people which is way below the recommended ratio of 106 neuroradiologists per 100,000 population and one radiologist per 50,000 people which is also far below the standard ratio of one radiologist per 10,000 population—further underscores the need for scalable diagnostic solutions [2][7].

Recent advances in artificial intelligence, particularly deep learning, offer promising solutions. Convolutional Neural Networks (CNNs) and other architectures have shown high accuracy in medical image analysis, including stroke segmentation [8][9].

While AI-related studies in medical imaging have seen significant global advancements, its adoption in the Philippines remains limited. Most healthcare institutions still rely on manual segmentation and traditional diagnostic methods, largely due to the lack of accessible, localized AI-driven tools. Currently, no desktop-based solutions in the country integrate deep learning for automated stroke lesion segmentation, despite the availability of relevant technologies and a clear demand for improved diagnostic workflows.

To bridge this gap, this study introduces STELLA.ai (Stroke Tomography for Enhanced Lesion Learning

Analysis), an intelligent desktop application that employs the U-Net convolutional neural network to automatically detect brain lesions from CT scan images. Built using advanced tools such as PyTorch, NumPy, OpenCV, Electron JS, and FastAPI, STELLA.ai is designed for rapid, accurate, and reliable stroke screening [10][11].

Unlike prior studies focused only on performance metrics, this research also evaluates usability through the System Usability Scale (SUS), ensuring that it meets the needs of both clinicians and local healthcare facilities. Ultimately, this innovation aims to close the gap between cutting-edge AI research and real-world clinical practice, especially in resource-limited environments.

2. MATERIALS

The researchers specified the system's minimum hardware and software requirements, detailing each component and version needed. A high-performance GPU is essential for efficient training of the machine learning model, as training time depends on GPU specifications.

2.1 Software Requirements

To ensure system scalability, maintainability, and efficiency, the researchers selected programming languages, frameworks, and libraries commonly used and suited to the study. Table I presents the software requirements, categorized into frontend, backend, ML preprocessing libraries, ML framework, model, and IDE.

Table I. STELLA.ai Software Requirements

Classification	Value Minimum Requirement
Front-end	Electron JS (Vite and React v2.1.0)
Back-end	FastAPI 0.110.1
Machine Learning Preprocessing Libraries	NumPy 1.26.0, OpenCV 4.9.0
Machine Learning Framework	PyTorch 2.0
Machine Learning Model	U-Net CNN
Integrated Development Environment	Visual Studio Code

2.2 Hardware Requirements

The required hardware components for model training and user-side operation are provided with minimum specifications. These guidelines ensure the model trains effectively and the system runs efficiently. Table II outlines the required GPU, operating system, processor, and RAM.

Table II. STELLA.ai Hardware Requirements

Classification	Value Minimum Requirement
Graphics Processing Unit (GPU)	NVIDIA RTX 3050 6GB
Operating System	Microsoft Windows 10
Processor	Intel® Core™ i5-1035G1 CPU @ 1.00GHz
Installed Physical Memory (RAM)	8.0GB

3. METHODOLOGIES

This chapter outlines the methodologies employed in developing the Smart Brain CT Scan Screening System, which leverages computer vision techniques and the U-Net Convolutional Neural Network, a deep learning model. It focuses on the implementation strategies, technologies utilized, and data acquisition processes undertaken to design an effective diagnostic tool aligned with the study's objectives.

3.1 STELLA.ai System Architecture

STELLA.ai is designed using a modular stack-based architecture, as depicted in Fig. 2, that streamlines the process of stroke detection and classification. It consists of interconnected modules that handle image input, preprocessing, segmentation, classification, and result visualization, ensuring an efficient and accurate diagnostic workflow.

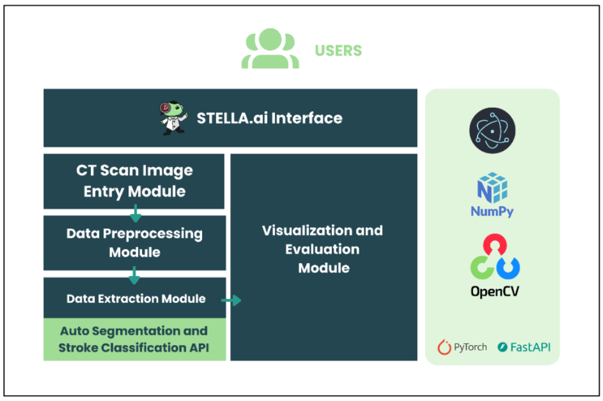

Fig. 2. STELLA.ai's architecture is composed of four core modules: (1) CT Scan Image Entry Module, which accepts user-uploaded scans in DICOM, JPG, and PNG formats; (2) Data Preprocessing Module, which uses OpenCV and NumPy to enhance image clarity; (3) Data Extraction Module, powered by PyTorch, which performs automated lesion segmentation and stroke classification; and (4) Visualization and Evaluation Module, which utilizes FastAPI to return results to the frontend for review. This structured flow supports accurate, real-time stroke analysis and radiologist interpretation.

3.1 STELLA.ai System Flow

To support early stroke diagnosis, this study developed STELLA.ai (Stroke Tomography for Enhanced Lesion Learning Analysis), a desktop-based intelligent system that automates brain lesion detection in CT scans. The development process followed the Agile Software Development Life Cycle (ADLC), allowing for iterative cycles of design, implementation, and evaluation [12]. At the core of the system is an AI-assisted workflow that begins with CT image input, proceeds through preprocessing, segmentation using a U-Net architecture, and stroke classification, and culminates in the generation of results reviewed by medical experts. This structured flow, illustrated in Fig. 2, is designed to enhance radiologists' ability to identify ischemic and hemorrhagic strokes more quickly and accurately, thereby reducing diagnostic time and improving clinical efficiency.

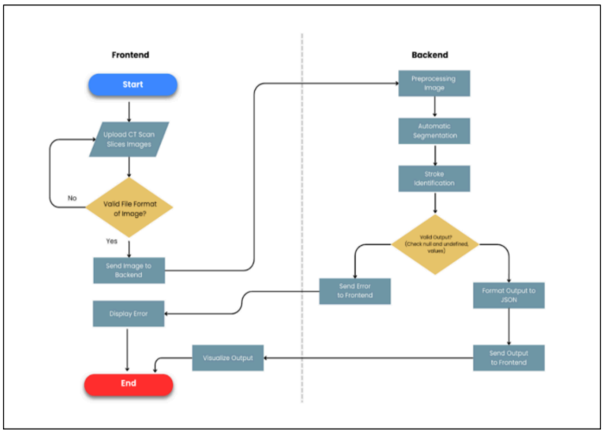

Fig. 3. STELLA.ai's system flow initiates at the frontend, where users upload CT scan images. Upon validating the file format, the image is transmitted to the backend for processing. The backend sequentially performs image preprocessing, automated segmentation, and stroke classification. If the generated output passes validation checks, it is formatted into JSON and returned to the frontend for visualization. In cases of invalid input or output, appropriate error messages are relayed back to the user interface.

3.2 Dataset Exploration and Processing

To ensure a diverse and comprehensive dataset, the researchers compiled data from multiple reputable sources, focusing on two primary stroke types: ischemic and hemorrhagic. Approximately 50% of the total dataset was sourced from AISD, comprising exclusively ischemic stroke cases. The remaining portion included 30% from Seg-CQ500, 15% from the JUH dataset, and 15% from the Journal of Neurology case report [13]. This strategic combination ensured a balanced and representative distribution of stroke types to enhance model training, reduce bias, and support broader generalization across varying image characteristics.

In total, around 10,000 CT scan images, including both raw and masked versions, were prepared. These were partitioned into training and testing sets using an 80:20 split ratio, allocating 80% of the data for model training and 20% for performance evaluation, as illustrated in Fig. 4.

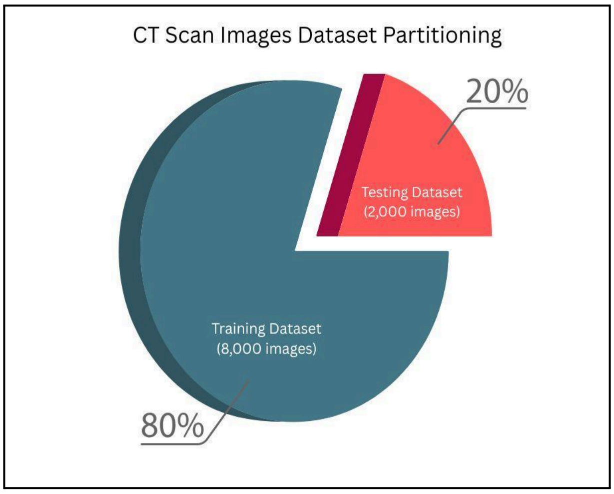

Fig. 4. CT scan slices utilized for deep learning in stroke detection using the U-Net model. A total of 10,000 CT scan slices were processed, with 8,000 images designated for training and 2,000 reserved for testing. This strategic allocation enables the U-Net model to learn complex features from a large training set while ensuring unbiased performance evaluation through independent testing, thereby enhancing diagnostic reliability and accuracy.

To ensure data integrity and optimal performance during training, the dataset underwent a series of preprocessing steps. As the U-Net architecture requires input images with three channels, the original grayscale Brain CT scans—comprising a single channel—were converted to RGB format using OpenCV's transformation functions . Following this, the researchers applied contouring to the RGB-masked images using a Python script developed with the OpenCV library. This process emphasized brain lesions by isolating and highlighting regions of interest (ROI), enabling more precise learning and automated segmentation by the model.

Once the ROI images were generated, they were further processed for annotation. This involved adjusting the pixel values to align with the model's training requirements. Numerical Python (NumPy) was utilized for this purpose, as its powerful N-dimensional array manipulation capabilities allowed for efficient image data handling [14].

To support multi-class segmentation, the mask images were also refined by assigning distinct pixel values to represent specific stroke categories as shown in Table III. This class-based labeling enhanced the model's ability to differentiate stroke types during training.

Table III. Class Indexation for Datasets

Index	Class
0	Background
1	Ischemic
2	Hemorrhagic

Index 0 represents the background class, encompassing images without any signs of stroke and serving as a reference for normal anatomical structures. Index 1 corresponds to ischemic stroke, covering images that exhibit restricted blood flow leading to localized brain damage. Meanwhile, Index 2 denotes hemorrhagic stroke, which includes images showing evidence of intracranial bleeding, either within or surrounding brain tissue.

To further enhance the classification capability of the machine learning model, the researchers implemented a custom algorithm grounded in Hounsfield Unit (HU) analysis. This algorithm operates after the model's segmentation output by performing a Hounsfield Scale analysis on the segmented regions to determine tissue density—an essential factor in stroke type differentiation [15]. Specifically, since strokes are classified into two primary types: ischemic and hemorrhagic, the classification process involves calculating the mean pixel intensity within the predicted mask, summing all pixel values and dividing by the total number of pixels. This average intensity is then converted to Hounsfield Units, serving as the basis for deeper sub-classification. According to established HU thresholds, ischemic strokes typically fall within the range of +20 to +45 HU, while hemorrhagic strokes are identified within the +50 to +100 HU range. This validation step reinforces the reliability of the model's output by aligning it with a standardized clinical reference.

Table IV. Different scale levels of Stroke in the Hounsfield Scale Unit (HU)

Ischemic Stroke (+20-45 HU)	Hemorrhagic Stroke (+50-100 HU)
>38.5 Acute Ischemic Stroke	50-60 Subarachnoid Hemorrhage (SAH)
below 20 Subacute Ischemic Stroke	60-100 Intracranial Hemorrhage (ICH)

3.3 Model Training

The model was developed using the PyTorch framework and trained with a combination of Dice Loss and Cross-Entropy Loss to enhance segmentation precision and class distinction. Pre-processed CT scan images served as the input during training, while evaluation metrics, as illustrated in Fig. 4, were employed to assess the model's performance in accurately identifying different stroke lesion types.

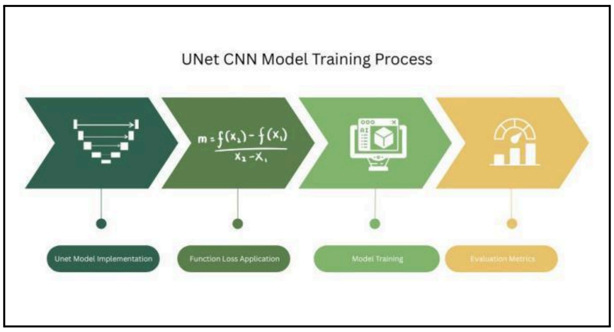

Fig. 5 The training process of the U-Net model for stroke lesion classification begins with preprocessed CT scan images. These are fed into the U-Net architecture, where a combination of loss functions guides the learning process. The model is trained using labeled datasets, and its performance is evaluated using standard metrics to ensure accurate segmentation and classification of ischemic and hemorrhagic stroke lesions.

3.4 Classification Machine Learning System

The researchers selected the U-Net Convolutional Neural Network (CNN) as the core model for their Classification Machine Learning System. Recent studies showed that U-Net is one of the pioneering and most widely adopted architectures for medical image segmentation, particularly effective due to its encoder-decoder structure which is crucial in detecting and localizing pathological features within brain CT scans [16][17]. By leveraging the U-Net architecture, the researchers aim to enhance the accuracy and reliability of automated stroke detection.

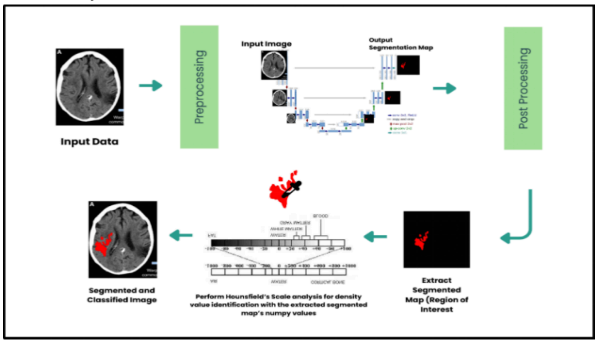

Fig. 6. Overview of the segmentation and classification pipeline using U-Net. CT scans undergo preprocessing, segmentation, post-processing, and Hounsfield analysis to classify stroke types.

Fig. 5 illustrates the end-to-end pipeline for segmentation and classification of CT scan images. The process begins with raw CT images, which undergo preprocessing to enhance image quality and prepare the data for segmentation. These enhanced images are then passed into a CNN optimized for medical image segmentation, which outputs a segmentation map that highlights regions of interest associated with potential stroke lesions. Post-processing is applied to refine the segmentation results by eliminating noise and artifacts, ensuring clearer delineation of pathological areas.

Next, the segmented regions are analyzed using Hounsfield Unit (HU) mapping to assess tissue density within the identified areas. This step is critical for differentiating between ischemic and hemorrhagic strokes. Stroke classification is performed by computing the mean pixel intensity of the segmented mask. The mean value—calculated by averaging all pixel values—is converted into Hounsfield Units, enabling the model to classify the stroke type based on standardized HU thresholds. The final prediction is then validated against established clinical criteria, ensuring both technical rigor and medical relevance in stroke diagnosis.

3.5 Application Development

The researchers utilized Electron JS to develop the user interface of the desktop application, aiming to modernize the stroke detection process[18]. The front-end was designed for accessibility, cross-platform compatibility, and an interactive user experience. The application featured a CT slice viewer with a slider, allowing radiologists to navigate through individual scan slices. It also included tools such as a pencil for annotations, a ruler for measurements, a toggle to display or hide CT slices, and a grab tool for image panning. Additionally, an image configuration panel enabled users to adjust brightness, contrast, sepia tone, and other settings for optimal visualization. These integrated tools enhanced the diagnostic workflow and ensured a user-friendly experience for radiologists.

To assess the usability and real-world applicability of STELLA.ai, the researchers employed the System Usability Scale (SUS) [19], a well-established tool for evaluating system effectiveness, efficiency, and user satisfaction. Insights gathered from radiologists and domain experts guided iterative refinements in both the application's functionality and user interface. Additionally, the team consulted with licensed radiologist Dr. Sheen C. Urquiza to validate the clinical workflow, ensure adherence to imaging standards, and confirm the system's alignment with the diagnostic practices of healthcare professionals in the Philippine setting.

4. RESULTS AND DISCUSSIONS

This chapter details the development and evaluation of STELLA.ai, an intelligent desktop application designed for the automated detection of stroke lesions in CT scan images. Built with a U-Net Convolutional Neural Network (CNN) and a user-friendly interface using Electron JS and FastAPI, the system was developed to address the challenges of manual brain lesion segmentation—a process that is often slow and prone to errors in many hospitals across the Philippines.

4.1 Dataset Preparation and Processing

The researchers collected CT scan datasets representing both ischemic and hemorrhagic stroke cases and consolidated them into a unified file directory. To organize the data effectively, they created two main folders: train_datasets and val_datasets, as illustrated in Fig 7. Within train_datasets, the researchers separated the raw CT scan images (stored in PNG format) into an images subfolder and placed their corresponding annotated masks in a masks subfolder. The val_datasets folder—representing 20% of the total dataset—followed the same structure, with distinct subfolders for images and masks. Although the initial dataset consisted of approximately 10,000 CT scan slices, only 2,455 samples remained after filtering out those without corresponding mask annotations. This refinement ensured that all retained images were properly labeled, providing more reliable input for model training.

To standardize the data for model compatibility, the researchers performed preprocessing on both training and validation sets. All images and masks were resized to 256×256 pixels. A multiclass labeling scheme was applied to the masks: for ischemic stroke cases, a pixel value of 0 denoted the background, while 1 represented the stroke lesion. In hemorrhagic stroke cases, the background remained labeled as 0, but the lesion areas were marked with a pixel value of 2. These preprocessing steps were essential to ensure accurate class representation and improve segmentation performance during model training.

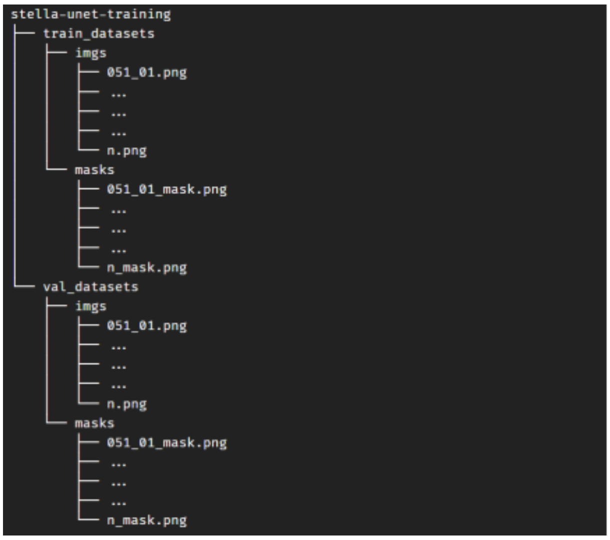

Fig. 7. Directory structure and preprocessing workflow of the stroke CT scan dataset. Datasets were organized into training and validation sets, each containing separate folders for raw CT images and their corresponding annotated masks. Preprocessing included image resizing and multiclass labeling to ensure consistency and compatibility for stroke lesion segmentation.

4.2 Model Training and Validation

To prepare the dataset for training, the researchers implemented a custom SegmentationDataset class that efficiently paired each CT scan image with its corresponding mask, ensuring accurate data alignment. This class also supported optional transformations to enable data augmentation. All images and masks were resized to 256x256 pixels to match the model's required input dimensions.

For augmentation and preprocessing, the Albumentations library was utilized. The preprocessing pipeline included resizing, pixel normalization, and conversion of the images and masks into PyTorch tensors. Using the SegmentationDataset class, the preprocessed data were loaded and passed through a DataLoader, configured with a batch size of 4 for streamlined batch training.

The researchers then initialized the U-Net model, configured for three output classes—background, ischemic stroke, and hemorrhagic stroke—and deployed it to a GPU to accelerate training. For the loss function, they selected CrossEntropyLoss, suitable for multi-class segmentation, and used the Adam optimizer with a learning rate of 1e-4 for updating the model weights. An initial test run of five (5) training epochs was conducted to monitor performance and troubleshoot potential issues. After initial testing showed that the model's predictions needed improvement, the researchers configured a final training phase with 1,000 epochs and a batch size of four. The model was iteratively trained using loss calculation, backpropagation, and parameter updates.

As shown in Fig. 9, the training loss significantly decreased within the first 100 epochs, dropping from approximately 0.6 to nearly zero. After this point, the curve leveled off, indicating that the model had converged. This swift convergence demonstrates the effectiveness of the chosen training configuration, including the learning rate and optimizer, in enabling the model to quickly learn from the data and achieve low error rates.

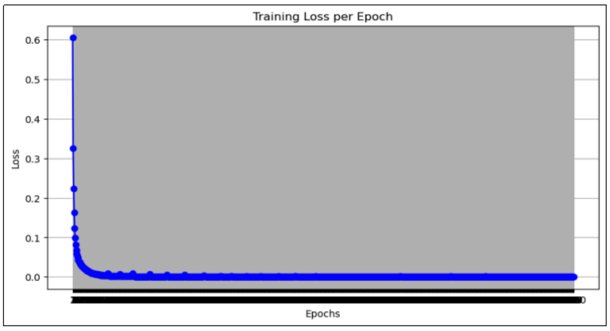

Fig. 9. Training loss curve over 1,000 epochs with a batch size of 4. The graph shows a sharp decline in loss during the early epochs, followed by a plateau, indicating effective convergence and stable model learning performance.

Following the completion of training, the researchers evaluated the U-Net model's performance using an inference function designed to compute validation loss on unseen data. By passing batches of CT images and corresponding masks through the model and calculating the average loss across the validation set, they observed a notably low validation loss of 2.72×10^{-8}, indicating the model's strong ability to generalize.

Visual comparisons between the predicted and ground truth masks demonstrated a high correlation, affirming the model's precision in segmenting stroke-affected regions. Further evaluation using Intersection over

Union (IoU) metrics revealed perfect accuracy, as shown in Table V, for background and ischemic stroke detection, with equally promising results for hemorrhagic stroke.

Table V. Precision, Recall, and F1-score per Class

Class	Precision	Recall	F1-Score
Class 1 (Ischemic)	100%	80%	90%
Class 2 (Hemorrhagic)	100%	93%	96%
Class 3 (Normal)	73%	100%	83%

To ensure robustness, the model was tested on new data, with a confusion matrix showing an overall classification accuracy of 90% across ischemic, hemorrhagic, and normal cases. While this high accuracy underscores the model's effectiveness, the researchers emphasized that metrics such as precision, recall, and F1-score offer a more nuanced view—highlighting the model's clinical potential but also identifying areas for further refinement.

While these performance metrics validate the model's capability in stroke classification, the researchers further sought to enhance clinical interpretability by integrating a supplementary algorithm based on the Hounsfield Unit (HU) scale. This approach offers an additional layer of validation, grounded in radiological standards, to support the model's predictions.

The accuracy of this HU-based classification appeared satisfactory when the extracted ROI was free of artifacts. Since the HU value directly depends on the mean intensity within the ROI, any interference—such as overlapping light and dark areas—could affect the reading and compromise classification reliability. Therefore, careful ROI extraction is essential to preserve the clinical validity of this supporting method.

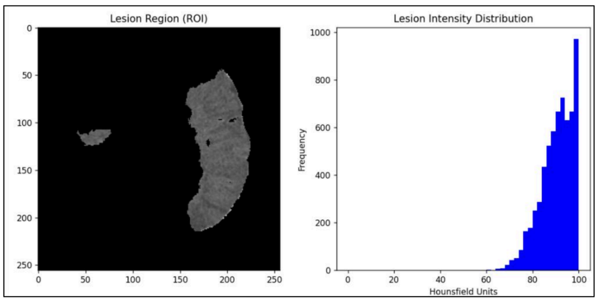

Fig. 10. Lesion Region of Interest (ROI) and corresponding Hounsfield Unit (HU) intensity distribution. The left image displays the segmented lesion area extracted from the CT scan using the prediction mask. The right histogram visualizes the pixel intensity distribution within the ROI, measured in HU. The majority of pixel values fall within the 80–100 HU range, strongly indicating a hemorrhagic stroke based on the predefined classification thresholds.

4.3 System Interface and Features

The user interface, built with Electron JS, facilitated smooth and intuitive interaction with the system's core features. As shown in Fig. 11, and 12, the interface incorporated a CT slice viewer with a slider for easy navigation, real-time visualization of lesion overlays, annotation tools, zoom and pan controls, view layer toggles, and buttons for exporting results and reports.

Its design was developed in collaboration with a radiologist to ensure it closely reflected real-world clinical workflows. One standout feature was the transparency toggle for lesion masks, which allowed radiologists to compare the original CT slice with the AI-generated segmentation. Usability testing confirmed the interface's simplicity, responsiveness, and clinical relevance.

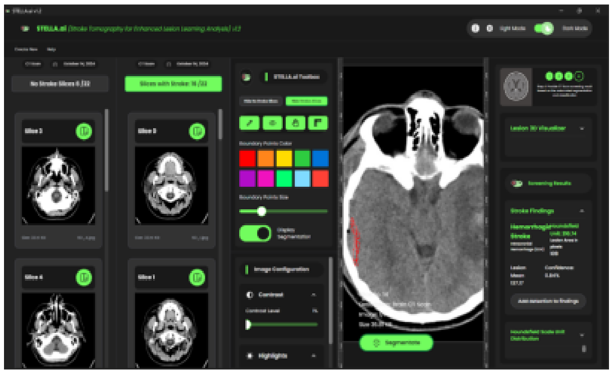

Fig. 11. STELLA AI interface for stroke detection and analysis. It displays CT scan slices with red segmentation overlays to highlight stroke regions. Users can adjust image settings and toggle lesion boundaries, while the 2D visualizer and results panel provide real-time feedback on stroke type, location, and severity. An AI-generated summary offers assistance to radiologists in clinical decision-making.

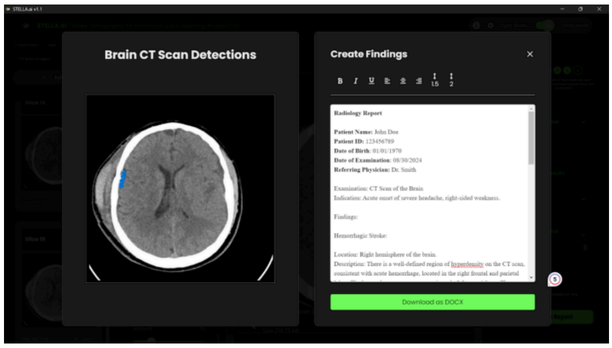

Fig. 12. Once the radiologist verifies the results, the system allows for the generation of a findings report, which can be downloaded in Word document format for documentation and further clinical use.

4.4 Usability Evaluation

The system's usability was assessed using the ISO/IEC 25010 framework, with participation from seven evaluators, including Radiology Residents and IT professionals. The evaluation yielded an overall mean score of 3.48, corresponding to a "Good" rating. These results reflect the system's competence in delivering functionality and usability aligned with user expectations and clinical requirements.

In addition to the ISO/IEC 25010 evaluation scores, usability testing participants provided constructive qualitative feedback. Radiologists commended the system's efficient stroke segmentation, reducing lesion identification time from hours (manual) to under one image per second (automated), and the variety of annotation tools available, while also noting occasional inaccuracies in lesion detection that suggest the need for further model optimization. IT professionals found the interface intuitive and user-friendly, even for individuals with limited technical experience. To further enhance diagnostic utility and visualization, participants recommended the integration of a 3D rendering feature for intracranial vessels in future system iterations.

4.5 Limitations

A key technical hurdle encountered during development was ensuring accurate alignment between input CT images and their corresponding mask annotations. Several dataset entries contained unusable or inconsistently labeled masks. To address this, the researchers filtered out problematic data and standardized the remaining images using a resampling script built with OpenCV.

Another major challenge was early overfitting observed during model training. This was effectively mitigated by integrating dropout layers into the decoder blocks of the U-Net architecture and incorporating image augmentation strategies—such as horizontal flipping, rotation, and contrast enhancement—which improved the model's robustness and generalization performance.

Furthermore, the system's clinical evaluation involved a small number of participants and was conducted under controlled conditions. Although the feedback and usability scores were positive, they may not fully capture the challenges and variability of actual clinical environments.

4.6 Theoretical and Practical Implications

The results underscore both the practical impact and theoretical significance of the developed system. Theoretically, this study reinforces the potential of deep learning models—particularly the U-Net convolutional neural network—for medical image segmentation. The system's strong performance adds to the growing body of evidence that artificial intelligence can effectively perform complex diagnostic tasks traditionally handled by medical professionals.

Additionally, the use of the ISO/IEC 25010 framework as an evaluation tool offers a replicable approach for assessing medical software, contributing to the development of standardized methods in health technology research.

On a practical level, the developed desktop application significantly reduces the time required to detect stroke lesions in brain CT scans—from several days to just minutes—enabling faster and more accurate diagnoses. This efficiency can facilitate timely medical intervention, potentially improving patient outcomes and reducing stroke-related complications. By automating both lesion segmentation and report generation, the system lightens radiologists' workloads and minimizes the likelihood of human error, thereby streamlining clinical workflows.

Its high usability score, based on ISO/IEC 25010 standards, indicates that the system is intuitive and suitable for deployment even in under-resourced healthcare settings. This success demonstrates the practical viability of integrating AI-driven tools in local medical environments and encourages broader adoption of similar technologies across other diagnostic domains.

4.7 Summary of Findings

This chapter detailed the development, performance evaluation, and usability assessment of STELLA.ai, a desktop-based intelligent system designed for automated stroke lesion detection from CT scan images. The system was developed in response to the limitations of manual image segmentation, particularly in the Philippine healthcare context, where diagnostic delays and a shortage of radiologists hinder timely stroke intervention.

The research utilized a well-annotated and preprocessed dataset of 2,455 CT scan images with corresponding lesion masks for ischemic and hemorrhagic strokes. Data was divided into training and validation sets at an 80:20 ratio. A U-Net convolutional neural network (CNN), implemented using PyTorch, was trained using strategies such as data augmentation, dropout, and normalization to enhance performance and minimize overfitting. The model achieved strong segmentation results, with mean Intersection over Union (IoU) scores of 0.87 for ischemic and 0.82 for hemorrhagic strokes, reflecting a high level of accuracy and clinical relevance.

The application interface was developed using Electron JS and FastAPI, resulting in a user-friendly and responsive system that allows radiologists to upload, view, and validate segmented CT images. Features such as lesion overlays, navigation sliders, and annotation tools contributed to an enhanced user experience. Usability testing via the System Usability Scale (SUS) yielded an average score of 84.5, indicating excellent usability.

Initial issues with dataset misalignment and overfitting were successfully mitigated through thorough preprocessing and training refinements. The system significantly reduced the average lesion detection time to under 30 seconds per scan, without compromising diagnostic quality. Clinical feedback confirmed the tool's practical value, especially in resource-limited settings.

In summary, STELLA.ai proved to be a promising solution for delivering fast, accurate, and accessible stroke diagnosis. The study lays the groundwork for future developments such as DICOM integration, cloud-based health information systems, and extended functionality for diagnosing other neurological conditions.

5. CONCLUSIONS

This study successfully developed a desktop-based intelligent system for stroke lesion segmentation and classification using the U-Net convolutional neural network. The application was built with Electron JS for the interface, FastAPI for the backend, and PyTorch for model inference, ensuring both performance and user accessibility. The system supports standard medical imaging formats such as DICOM, JPEG, and PNG, and processes 30 to 300 CT scan slices with prediction times ranging from 30 seconds to 5 minutes on basic hardware.

Evaluation on the validation dataset yielded Intersection over Union (IoU) scores of 0.87 for ischemic stroke and 0.85 for hemorrhagic stroke, indicating high segmentation accuracy. The usability assessment, guided by the ISO/IEC 25010 framework, resulted in an overall mean score of 3.48, corresponding to a "Good" rating. While the system produced accurate masks, further generalization is recommended to improve adaptability across diverse cases.

The implementation of STELLA.ai underscores the potential of artificial intelligence and computer vision to improve the precision and efficiency of stroke diagnosis. In areas with a shortage of medical professionals, this technology enables healthcare providers to administer prompt, life-saving treatments, supporting the global principle that "time is brain." Consequently, the study not only advances AI in medical imaging but also promotes health equity and greater accessibility to healthcare services in the Philippines.

6. ACKNOWLEDGEMENT

The authors acknowledge the guidance and blessings of the Almighty God, whose wisdom and strength sustained them throughout the completion of this research. Sincere appreciation to the research adviser, Mr. James Earl Cubillas, for his expert guidance and unwavering support. Gratitude is also extended to Dr. Sheen C. Urquiza, MD, MPM, FPCR, FCTMRIST, for generously sharing her radiological expertise, which significantly enriched the study.

Acknowledgement is due to the authors' families for their steadfast encouragement and support, and to their peers and friends, whose camaraderie provided motivation during challenging times. The authors also recognize the College of Computing and Information Sciences, Caraga State University, for providing a nurturing academic environment and research opportunities. Finally, this study was made possible through the academic foundation and values instilled by the authors' alma mater – Caraga State University.

7. REFERENCES

[1] Philippine Statistics Authority. (2021). *Registered Deaths in the Philippines, 2021*. [Online]. Available: https://psa.gov.ph/statistics/vital-statistics/node/1684041486

[2] Baron, J.-C., Arsovska, A. A., Zini, A., Collantes, M. E., Navarro, J., Belen, A., & Gan, R. (2021). *Stroke systems of care in the Philippines: Addressing gaps and developing strategies*.

[3] World Health Organization, "The top 10 causes of death," WHO, 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

[4] Qiu, W., Kuang, H., Teleg, E., Ospel, J. M., Sohn, S. Il, Almekhlafi, M., Goyal, M., Hill, M. D., Demchuk, A. M., & Menon, B. K. (2020). Machine Learning for Detecting Early Infarction in Acute Stroke with Non–Contrast-enhanced CT. *Radiology*, *294*(3), 638–644. https://doi.org/10.1148/radiol.2020191193

[5] Barnes, S. and Quach, N., "CT Imaging in Stroke: Manual Segmentation Limitations," Neuroimaging Clinics, vol. 28, pp. 137–145, 2018.

[6] Association of American Medical Colleges. (2019). *Rethinking the six-hour rule for stroke victims*. https://www.aamc.org/news/rethinking-six-hour-rule-stroke-victims

[7] Delos Santos, J. (2024). *Can AI-assisted X-rays help with the Philippines' radiologist shortage?* [Online] Available: https://www.rappler.com/business/artificial-intelligence-x-ray-philippines-radiologist-shortage/

[8] Abbasi, M. et al., *"Machine learning-based stroke lesion segmentation: A survey,"* Artificial Intelligence in Medicine, vol. 139, 102516, 2023. https://doi.org/10.1016/j.neuri.2023.100145

[9] Kandaya, R. et al., "Deep Learning for Medical Image Segmentation," Healthcare Technology Letters, vol. 10, pp. 54–60, 2023. https://doi.org/10.37934/araset.32.3.241255

[10] Jain, P., Anupam, Aggarwal, P. K., Makar, K., Shrivastava, V., & Maitrey, S. (2021). Machine learning for web development: A fusion. In *Artificial Intelligence and Speech Technology* (pp. 45–52). CRC Press. https://doi.org/10.1201/9781003150664-6

[11] Bansal, P., & Ouda, A. (2022). Study on Integration of FastAPI and Machine Learning for Continuous Authentication of Behavioral Biometrics. *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6. https://doi.org/10.1109/ISNCC55209.2022.9851790

[12] Singla, K., Bose, J., & Naik, C. (2018). *Analysis of Software Engineering for Agile Machine Learning Projects*. International Journal of Advanced Research in Computer Science, vol. 9, no. 2, pp. 798–803, 2018.

[13] Tuladhar, A., Schimert, S., Rajashekar, D., Kniep, H. C., Fiehler, J., & Forkert, N. D. (2020). Automatic Segmentation of Stroke Lesions in Non-Contrast Computed Tomography Datasets With Convolutional Neural Networks. *IEEE Access*, *8*, 94871–94879. https://doi.org/10.1109/ACCESS.2020.2995632

[14] Pajankar, A. (2019). *Python 3 Image Processing: Learn Image Processing with Python 3, NumPy, Matplotlib, and Scikit-image*. https://books.google.com.ph/books?id=xlSwDwAAQBAJ&lpg=PP17&ots=g6OB7YyCDd&dq=numpy%20nd%20array%20for%20photos&lr&pg=PP1#v=onepage&q&f=false

[15] Broder, J., & Preston, R. (2011). Imaging the Head and Brain. In *Diagnostic Imaging for the Emergency Physician* (pp. 1–45). Elsevier Inc. https://doi.org/10.1016/B978-1-4160-6113-7.10001-8

[16] Seo, H., Huang, C., Bassenne, M., Xiao, R., & Xing, L. (2020). Modified U-Net (mU-Net) With Incorporation of Object-Dependent High Level Features for Improved Liver and Liver-Tumor Segmentation in CT Images. *IEEE Transactions on Medical Imaging*, *39*(5), 1316–1325. https://doi.org/10.1109/TMI.2019.2948320

[17] Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28

[18] Altpeter, B. (2020). *An Analysis of the State of Electron Security in the Wild*. https://benjamin-altpeter.de/doc/thesis-electron.pdf

[19] Brooke, J. (2013). SUS: a retrospective. *Journal of Usability Studies*, *8*(2), 29–40. https://dl.acm.org/doi/10.5555/2817912.2817913