FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 12, December (Special Issue), 2025, pp 304 – 310
DOI: https://doi.org/10.33003/fjs-2025-0912-3949

APPLICATION OF U-NET ARCHITECTURE FOR FLOOD DETECTION IN BAYELSA STATE

***Emmanuella C. M. Obasi, Promise A. Nlerum and Francis C. Eze**

Department of Computer Science, Federal University Otuoke, PMB 126, Yenagoa, Nigeria

*Corresponding authors' email: obasiec@fuotuoke.edu.ng Phone: +2347036673665
ORCID iD: https://orcid.org/0009-0001-1513-9887

ABSTRACT

Floods are recurring disasters in Bayelsa State, Nigeria, causing significant damage to infrastructure, displacement of people, and loss of livelihoods. The research aims to develop an accurate and efficient method for identifying flood-affected areas using satellite imagery and deep learning techniques. The U-Net architecture, a convolutional neural network designed for image segmentation tasks, was adapted and trained on a dataset of high-resolution satellite images including both flood and non-flood periods. The model's performance was evaluated using various metrics, including precision, recall, and F1-score. Results demonstrate that the U-Net-based approach achieves high accuracy in delineating flood extents, outperforming traditional methods. The study also explores the model's ability to detect flood progression over time and its potential for real-time flood monitoring. The model achieved an accuracy of 88.66%, Recall of 0.90, Loss of 0.2846, Dice of 0.90, and IoU of 0.75. This research contributes to the development of advanced flood detection systems, which can aid in disaster management and mitigation efforts in Bayelsa State and similar flood-prone regions.

Keywords: Deep Learning, Disaster Management, Flood Monitoring, Image Segmentation, Satellite Imagery, U-Net Architecture

INTRODUCTION

Floods are a recurring natural disaster in Bayelsa State. They are among the most devastating natural disasters, causing significant economic losses, displacement of people, and loss of life worldwide (IPCC, 2021). Bayelsa State's unique geography, with its numerous rivers and low-lying terrain, makes it particularly vulnerable to flooding. Effective flood detection and management are crucial for mitigating the impact of these events. Uso and Onye (2025) carried out research on the Performance evaluation of five probability distribution models for the analysis of flood data at River Niger. Their study evaluates five statistical models to determine the most reliable method for predicting extreme flood events at River Niger, Onitsha. Climate change is increasingly linked to the intensification of flood disasters in coastal Nigeria (Adelekan, 2016). In Nigeria, particularly in Bayelsa State, floods are a recurring phenomenon, exacerbated by climate change, poor urban planning, and inadequate infrastructure (Nkwunonwo et al., (2020). The state's geographical location in the Niger Delta region makes it prone to flooding, with severe consequences on the environment, economy, and human lives (Ologunorisa, 2004). To mitigate the impact of floods, early detection and mapping are crucial. Traditional methods of flood detection, such as ground-based sensors and manual surveys, have limitations in terms of coverage, accuracy, and timeliness (Wang et al., 2019).

Remote sensing provides cost-effective, real-time monitoring of flood-prone areas, even in inaccessible regions (Schumann & Domeneghetti, 2016). Remote sensing and Geographic Information Systems (GIS) have emerged as crucial tools for flood detection and management. Satellite imagery, both optical and radar, plays a significant role in monitoring and mapping flood events (Lin et al., 2016; Opolot, 2013). These technologies offer cost-effective and timely access to spatial data, even in physically inaccessible areas. For instance, Sentinel-1 satellite imagery has been used to extract flooding information and develop machine learning models for flood detection (Tanim et al., 2022). Sentinel-1 SAR data have been increasingly applied for near real-time flood detection in developing countries (Twele et al., 2016). Interestingly, while advanced technologies are gaining prominence, traditional sensor-based monitoring techniques still retain significant advantages in practical applications. Future flood risk monitoring should focus on integrating multiple data sources to achieve real-time and accurate monitoring of urban flooding (Song et al., 2024). This approach could be particularly beneficial for regions like Bayelsa State, where a combination of traditional and modern techniques might yield the most comprehensive flood detection system. Early-warning systems powered by AI reduce disaster response times significantly (Guo et al., 2020). Hence, the need to employ machine learning techniques in flood detection in Bayelsa State. Machine learning algorithms outperform traditional thresholding methods for flood detection in complex terrains (Shen et al., 2019). Recent studies show that convolutional neural networks have revolutionized image segmentation tasks in environmental monitoring (Zhang et al., 2021).

Convolutional neural network like U-Net can be employed. The use of U-Net in biomedical imaging has inspired its adaptation for environmental applications (Falk et al., 2019). Deep learning architectures such as U-Net and ResNet have been widely applied for geospatial image segmentation tasks (Zhu et al., 2019). This study applies the U-Net deep learning architecture to satellite imagery for accurate flood detection in Bayelsa State, aiming to enhance disaster preparedness, support real-time monitoring, and provide a reliable framework for effective flood risk management.

Discussion on Techniques Used

The research employed a combination of remote sensing, deep learning, and image preprocessing techniques to achieve effective flood detection. The primary methodological framework was built around the U-Net architecture, a convolutional neural network (CNN) tailored for image segmentation tasks. This choice was motivated by U-Net's proven capability in capturing spatial details while preserving

contextual information through its encoder–decoder structure and skip connections.

i. Dataset and Metadata: The dataset consisted of 290 paired satellite images and binary flood masks sourced from the *Segmentation_of_Flood_Area* repository. Each image was standardized to 256×256 pixels for consistency. Metadata, including image paths and train-test splits, was stored in a structured CSV file, ensuring smooth mapping between images and their corresponding masks.
ii. Data Preprocessing: Input images were resized and normalized to fall within a [0,1] pixel value range to enhance gradient stability during training. Masks were thresholded and converted to Boolean arrays to preserve binary semantics. Bilinear interpolation was applied for resizing images, while nearest-neighbor interpolation maintained mask integrity.
iii. U-Net Architecture Design: The segmentation model followed the traditional U-Net structure without a pre-trained backbone.
    a. Encoder Path: Five down-sampling blocks progressively captured abstract spatial features, increasing filter depth from 16 to 256. Dropout layers were introduced to reduce overfitting.
    b. Bottleneck: A high-capacity block with 512 filters captured the most abstract representations of the flood-affected regions.
    c. Decoder Path: Symmetrical up-sampling blocks reconstructed image details, concatenating encoder features via skip connections to preserve spatial accuracy.
    d. Output Layer: A 1×1 convolution with sigmoid activation produced probability maps for per-pixel flood prediction.
iv. Training Protocol: Training was conducted using the Adam optimizer and binary cross-entropy loss function, suitable for two-class pixel labeling. The dataset was split into 90% training and 10% validation. Hyperparameters included a batch size of 8, 50 epochs, and real-time performance monitoring through TensorBoard. Early stopping was considered but not activated, allowing the model to run the full 50 epochs.
v. Evaluation Metrics: The model's performance was measured using multiple segmentation metrics:
    a. Accuracy (88.66%):Pixel-wise correctness.
    b. Precision (0.80): Reliability in identifying true flood pixels.
    c. Recall (0.90): Ability to capture actual flood regions.
    d. Dice Coefficient (0.90): Overlap measure between predicted and true masks.
    e. Intersection over Union (IoU = 0.75): Robustness of flood boundary delineation. The IoU metric is widely adopted as a robust performance indicator in flood segmentation models (Rahman & Wang, 2016)

Related Works

Recent developments have recorded application of machine learning in solving real life problems. Obasi and Timadi (2025) applied supervised machine learning algorithms, Feed Forward Neural Network and Random Forest with zero trust principles to prevent SQL and Malware attacks in a cloud database. A Feedforward Neural Network (FFNN) classified SQL queries as benign or malicious, achieving 98% accuracy with high precision in detecting SQL injection attempts. In parallel, a Random Forest classifier was used for malware traffic detection, attaining 99.37% accuracy by analyzing behavioral and statistical features. Their research was further expanded by combining high detection accuracy with explainable artificial intelligence (XAI) techniques, providing both transparency and reliability for modern cybersecurity defense systems. Obasi and Nlerum (2023) developed a model for the Detection and Prevention of Backdoor Attacks using CNN with Federated Learning. Timadi and Obasi (2025) researched on Integrating Zero-Trust Architecture with Deep Learning Algorithm to Prevent Structured Query Language Injection Attack in Cloud Database. Nnodi and Obasi (2025) researched on Leveraging Artificial Intelligence for Detecting Insider Threats in Corporate Networks. Obasi and Stow (2023) formulated a Predictive Model for Uncertainty Analysis Pertaining to Big Data through the Utilization of a Bayesian Convolutional Neural Network (CNN). Machine learning models can predict reaction yields with high accuracy, guilding chemists in selecting high-yielding reactions and optimizing synthesis routes. As a result of that, research on Leveraging Machine Learning Algorithms for Enhanced Prediction of Product Yields and Purity in Chemical Reactions was developed (Obasi and Abosede, 2025). In infectious disease testing, ML algorithms are capable of processing large datasets beyond human analytical capabilities, providing predictive and actionable insights (Tran et al., 2021). For malaria detection, ML models have been utilized to automate the analysis of medical images, including the classification of blood smear images, thereby reducing reliance on subjective human interpretation (Pattanaik et al., 2020). An interpretable Early Warning System for Malaria Outbreak in Bayelsa State using Deep Learning and Climate Data was developed in 2025(Stow and Obasi, 2025).

Again, Satellite imagery and deep learning algorithms can be used to detect flood events and provide early warnings. An Early Warning System using Satellite Imagery and Deep Learning Algorithm was developed (Obasi et. al, 2025). Effective flood detection and management require a multidisciplinary approach, combining various technologies such as remote sensing, GIS, LiDAR, and machine learning (Jain, 2024). Recent advancements in deep learning and remote sensing have enabled the development of more effective flood detection systems (Li et al., 2020). The U-Net architecture, a type of convolutional neural network (CNN), has shown remarkable performance in image segmentation tasks, including flood detection (Ronneberger et al., 2015). By leveraging satellite imagery and geospatial data, the U-Net model can accurately identify and map flood-prone areas, providing valuable insights for flood risk management and emergency response Mateo-Garcia, G., et al. (2020). For Bayelsa State, implementing a comprehensive flood detection system that integrates satellite imagery, ground-based sensors, and advanced machine learning algorithm like U-Net could significantly enhance flood preparedness and response capabilities. The architecture of U-Net is unique in that it consists of a contracting path and an expansive path. The contracting path contains encoder layers that capture contextual information and reduce the spatial resolution of the input, while the expansive path contains decoder layers that decode the encoded data and use the information from the contracting path via skip connections to generate a segmentation map as shown in figure 1.

Flood detection using U-Net is a deep learning approach that leverages the U-Net architecture to identify and map flood-prone areas from satellite imagery. By leveraging U-Net's strengths, researchers and practitioners can develop effective

flood detection systems, enhancing situational awareness and supporting disaster response.

This study aims to apply the U-Net architecture for flood detection in Bayelsa State, utilizing satellite imagery and geospatial data. The research seeks to evaluate the performance of the U-Net model in detecting floods and to explore its potential applications in flood risk management and mitigation strategies.

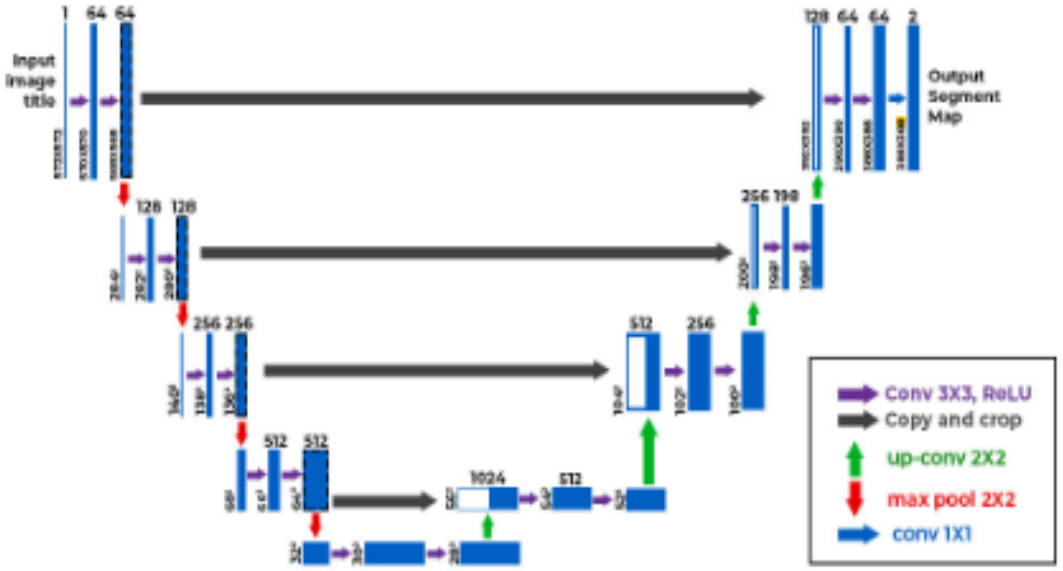

Figure 1: U-Net Architecture

MATERIALS AND METHODS

The methodology describes in detail the experimental procedures and architectural design adopted for flood-area segmentation using a U-Net model without any pre-trained backbone. The methodology is organized into the following sections: (1) Dataset and Metadata, (2) Data Preprocessing, (3) U-Net Architecture, (4) Training Protocol, and (5) Evaluation Metrics and Validation Strategy as shown in figure 2. The imagery and corresponding binary masks were obtained from the "Segmentation_of_Flood_Area" repository, comprising satellite or aerial photographs of size 256×256 pixels and their pixel wise flood labels. Metadata (e.g., file paths, train/test splits) were stored in a CSV file (metadata.csv), which mapped each image filename to its mask counterpart. Dataset comprises of 290 image–mask pairs, RGB (3 channels) and single-channel binary masks, where "1" denotes flooded pixels and "0" denotes background. A 90:10 random split was applied at training time (via validation_split=0.1 in model.fit()) to reserve 10% of the data for validation in each epoch, ensuring the model's generalization capacity is continually monitored. During data processing, each raw image (originally varying in dimension) was loaded using Pillow and converted to a NumPy array. Both images and masks were resized to 256×256 to enforce uniform input dimensions. Resizing employed bilinear interpolation for images and nearest neighbor or constant padding for mask arrays to preserve binary semantics.

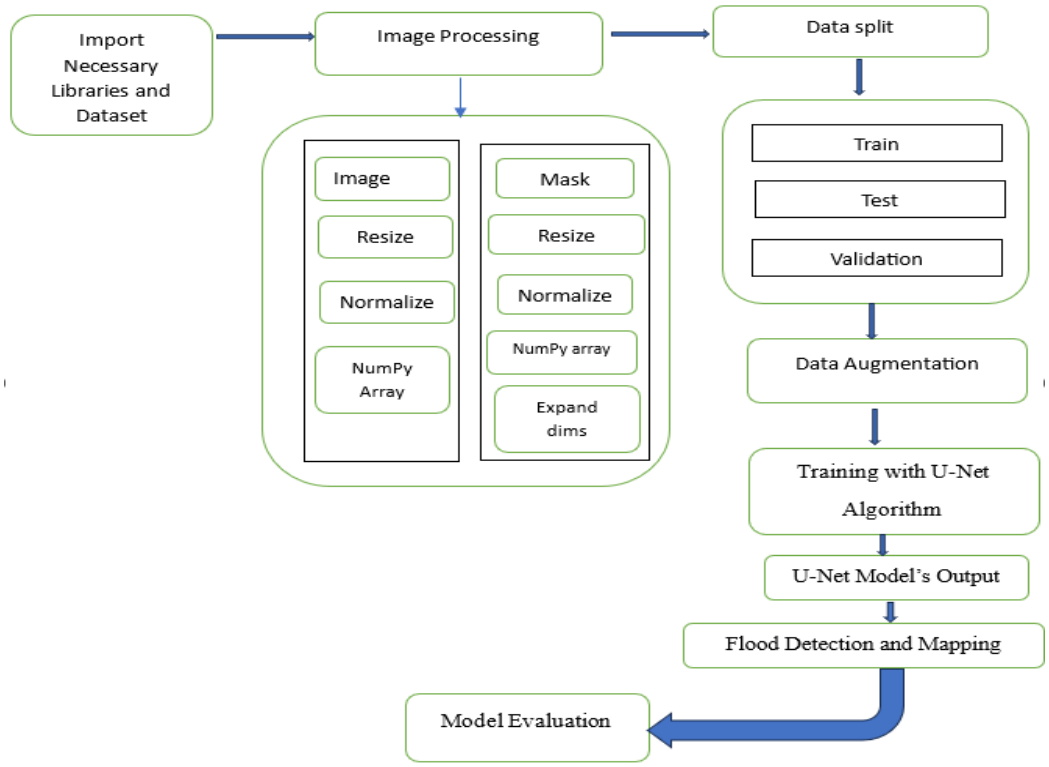

Figure 2: Architecture of the Flood Detection Model

Pixel values of input images were scaled to the range [0,1] via a Lambda layer (x → x/256) immediately upon input, facilitating stable gradient flow. Masks were read in grayscale, thresholded (automatically by resize preserving range), and stored as Boolean arrays (dtype=bool). An explicit channel dimension was appended (shape: 256×256×1).
The core segmentation model follows the classical U Net design, comprising a contracting path (encoder) and an expanding path (decoder), but constructed entirely from scratch, without leveraging any pre trained encoder backbone.

Input Layer
Shape: 256×256×3 (RGB), Initial normalization via a Lambda layer.

Encoder (Downsampling Path)
Five encoding blocks (Blocks 1–5) gradually increase feature depth while halving spatial resolution via MaxPooling:
Block 1: Three Conv2D layers (16 filters, 3×3, ReLU), Dropout(0.1), followed by MaxPool.
Block 2: Similar structure with 32 filters, Dropout(0.1).
Block 3: 64 filters, Dropout(0.2).
Block 4: 128 filters, Dropout(0.2).
Block 5: 256 filters, Dropout(0.3).

Bottleneck (Block 6)
Two Conv2D layers (512 filters, 3×3, ReLU), Dropout(0.3), capturing the most abstract representations.

Decoder (Upsampling Path)
Five decoding blocks (Blocks 7–11) symmetrically mirror the encoder:
Each block begins with Conv2DTranspose (up-sampling by factor 2), concatenation with the corresponding encoder feature maps (skip connection), followed by two or three Conv2D + ReLU layers and Dropout matching the encoder's dropout rate.

Output Layer
A final Conv2D (1×1, 1 filter) with sigmoid activation produces a 256×256×1 per-pixel probability map for the flood class.

Model Compilation
Optimizer: Adam (default learning rate). The Adam optimizer is among the most reliable for training deep segmentation models (Kingma & Ba, 2015).
Loss function: Binary cross-entropy (appropriate for two-class pixel labeling).
Metrics: Accuracy (pixel-wise).

Training Protocol
Hyperparameters (Batch size: 8, Epochs: 50, Validation split: 10%)
Callbacks: TensorBoard for real time monitoring (optional EarlyStopping commented out).

Training Workflow
Initialize model weights with Glorot uniform (default for Conv2D).
Fit on (X,Y) with validation_split=0.1.
Monitor training and validation loss/accuracy curves to identify potential overfitting.
While early stopping was configured at patience=2 (monitoring val_loss), it remained commented; model checkpoints were not explicitly saved, but TensorBoard logs enabled manual selection of best epochs.

RESULTS AND DISCUSSION
The plot_training () function displays training and validation loss and accuracy across epochs as shown in figure 3, marking best epochs for both metrics. This aids in diagnosing convergence and selecting final model weights.

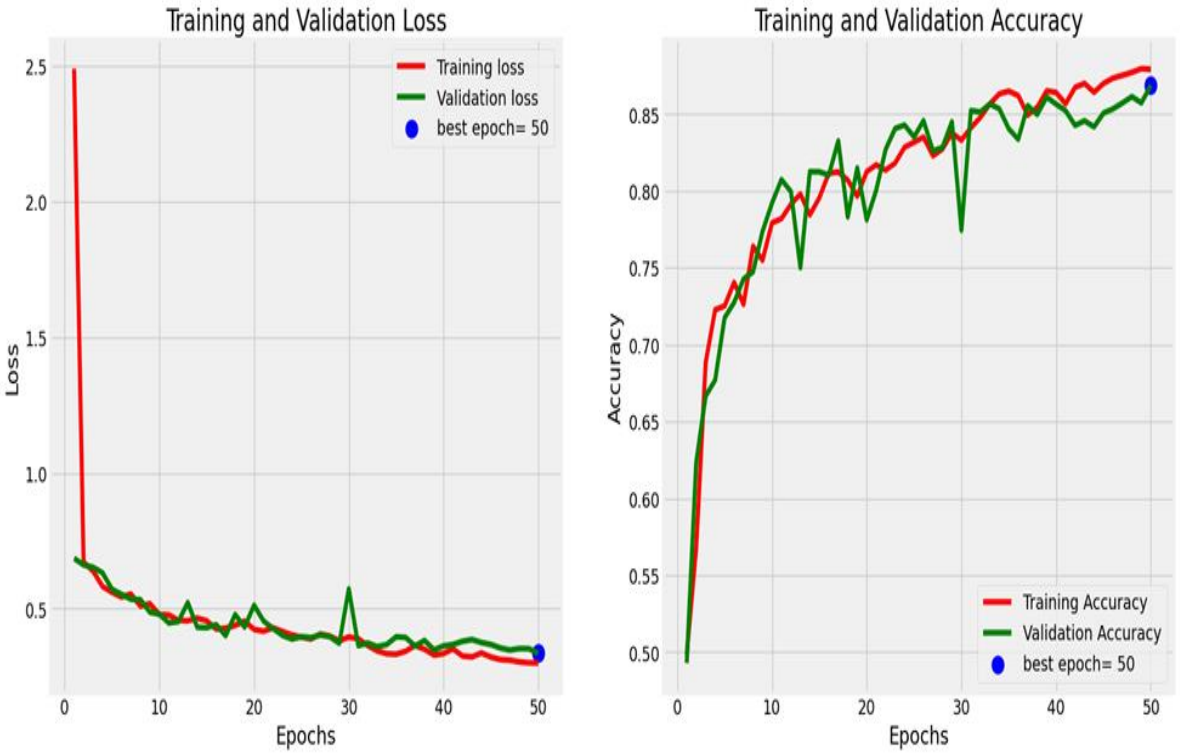

Figure 3: Training and Validation Loss

The plots in figure 3 show the performance of your model (likely part of a "Real-time Anti-Sleep Alert Algorithm") during the training process of over 50 epochs.: Both the training loss (red line) and the validation loss (green line) decrease significantly in the initial epochs. This indicates that the model is learning and improving its ability to make predictions. For convergence, both loss curves appear to level off and become relatively stable after around 20-30 epochs. This suggests that the model's learning rate has slowed down, and it's approaching a minimum loss value. The validation loss is consistently slightly higher than the training loss, which is expected. The training loss measures performance on the data the model sees during training, while the validation loss measures performance on unseen data. The small gap is normal. There is no clear indication of significant overfitting in the plot. Overfitting would typically be characterized by the training loss continuing to decrease while the validation loss starts to increase after a certain point. The curves track each other reasonably well. The plot highlights "best epoch = 50" for the validation loss. This suggests that the model achieved its lowest validation loss at the end of the training process (epoch 50), or that early stopping was not triggered before 50 epochs and the best performance up to that point was at epoch 50. Again, both the training accuracy (red line) and the validation accuracy (green line) increase rapidly in the initial epochs, mirroring the decrease in loss. This shows the model is becoming better at correctly classifying or segmenting (depending on the task of the algorithm) the data. Similar to the loss curves, the accuracy curves start to level off after around 20-30 epochs, indicating that the model's improvement rate has decreased. There are some noticeable fluctuations in both the training and validation accuracy curves, particularly between epochs 20 and 40. This could be due to the stochastic nature of mini-batch gradient descent, the specific data batches encountered in those epochs, or the learning rate schedule. The validation accuracy is generally close to the training accuracy, with some minor variations. This again suggests that the model is generalizing reasonably well to unseen data. The plot highlights "best epoch = 50" for the validation accuracy. This indicates that the highest validation accuracy was achieved at epoch 50, or that early stopping was not triggered and this was the best performance reached. The accuracy values are around 0.85-0.86 at this point.

Figure 4 provides a visual representation of your model's performance on a single example from the dataset.

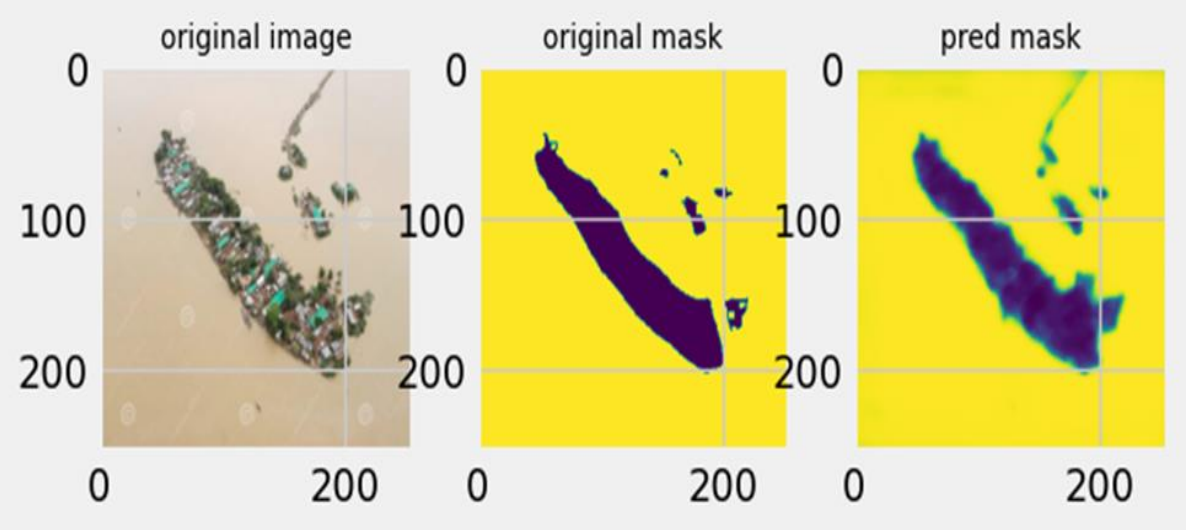

Figure 4: Visual Representation of the Result

The three panels in figure 4 show original image, original mask and pred mask. Original image is the input image provided to the model. Original mask is the ground truth mask corresponding to the original image. The segmentation mask predicted by our model for the original image is the Pred mask. It portrays the areas that our model identified as "flood". Similar to the original mask, the darker area represents the predicted flood pixels, and the lighter area is the predicted non-flood pixels.

CONCLUSION

By comparing the "pred mask" to the "original mask", one can visually assess how well the model performed on this specific image. The predicted mask captures the general shape and location of the flood area reasonably well compared to the original mask. The model achieved an accuracy of 88.66%, Precision = 0.80, Recall = 0.90, Loss = 0.2846, Dice = 0.90, IoU = 0.75. The high Dice score (0.90) and recall (0.90) show that the model reliably captures flooded regions, which is vital in disaster response. The lower precision (0.80) and IoU (0.75) point to occasional over segmentation.

ACKNOWLEDGEMENT

I want to acknowledge my co-researchers, Prof. Nlerum Promise and Mr. Eze Francis for contributing their best to the successful completion of this research.

REFERENCES

Adelekan, I. O. (2016). Flood risk management in the coastal city of Lagos, Nigeria. *Journal of Flood Risk Management*, 9(3), 255–264.

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., & Ronneberger, O. (2019). U-Net: Deep learning for cell counting, detection, and morphometry. Nature Methods, 16(1), 67–70.

Guo, H., Wang, L., & Chen, F. (2020). Artificial intelligence applications in flood management: Opportunities and challenges. *Water*, 12(6), 1695.

IPCC. (2021). Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

Li, Y., et al. (2020). Deep learning for flood detection: A review. Journal of Hydrology, 586, 124311.

Lin, L., Sun, Z., Rahman, M. S., Shrestha, R., Kang, L., Deng, M., Tang, J., Di, L., Hu, L., Zhang, C., & Yu, E. G. (2016, July). A review of remote sensing in flood assessment. In 2016 5th International Conference on Agro-Geoinformatics (Agro-Geoinformatics) (pp. 1–4). IEEE.

Mateo-Garcia, G., et al. (2020). Flood detection with satellite images using deep learning. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10), 8411–8423.

Nkwunonwo, U. C., et al. (2020). Flood risk assessment and mapping in Nigeria: A review. *Journal of Flood Risk Management*, 13(2), e12615.

Nnodi, J. T., & Obasi, E. C. M. (2025). Leveraging artificial intelligence for detecting insider threats in corporate networks. *University of Ibadan Journal of Science and Logics in ICT Research*, 13(1), 130–144.

Obasi, E. C. M., & Abosede, O. (2025). Leveraging machine learning algorithms for enhanced prediction of product yields and purity in chemical reactions. *Nile Journal of Engineering and Applied Sciences*.

Obasi, E. C. M., Anekwe, U. L., & Timadi, M. E. (2025). Development of an early flood warning system using satellite imagery and deep learning algorithm. *Nile Journal of Engineering and Applied Sciences.*

Obasi, E. C. M., & Timadi, M. E. (2025). Application of Machine Learning Algorithms with Zero Trust Principles for Preventing Malware and SQL Injection Attack in a Cloud Database. International Journal of Computer Applications, 187(53), 8–19.

Obasi, E. C. M., & Nlerum, P. A. (2023). A model for the detection and prevention of backdoor attacks using CNN with federated learning. *University of Ibadan Journal of Science and Logics in ICT Research*, 10(1), 9–21.

Obasi, E. C. M., & Stow, M. T. (2023). A predictive model for uncertainty analysis on big data using Bayesian CNN. University of Ibadan Journal of Science and Logics in ICT Research, 9(1), 52–62.

Ologunorisa, T. E. (2004). Flood risk assessment and management in Nigeria: Challenges and opportunities. *Journal of Environmental Sciences*, 16(2), 161–171.

Opolot, E. (2013). Application of remote sensing and geographical information systems in flood management: A review. *Research Journal of Applied Sciences, Engineering and Technology,* 6(10), 1884–1894.

Rahman, M. A., & Wang, Y. (2016). Optimizing intersection-over-union in deep neural networks for image segmentation. International Symposium on Visual Computing, 234–244.

Ronneberger, O., et al. (2015). U-Net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention, 234–241.

Schumann, G. J. P., & Domeneghetti, A. (2016). Exploiting remote sensing data for flood prediction. Hydrology and Earth System Sciences, 20(10), 4017–4039.

Shen, X., Li, J., & Wang, Y. (2019). Flood detection using machine learning techniques: A review. Remote Sensing, 11(15), 1769.

Song, J., Shao, Z., Zhan, Z., & Chen, L. (2024). State-of-the-art techniques for real-time monitoring of urban flooding: A review. *Water*, 16(17), 2476.

Stow, M. T., & Obasi, E. C. M. (2025). An interpretable early warning system for malaria outbreak in Bayelsa State using deep learning and climate data. International Journal of Advanced Research in Computer and Communication Engineering, 14(8), 58–101.

Tanim, A. H., Goharian, E., Mcrae, C. B., & Tavakol-Davani, H. (2022). Flood detection in urban areas using satellite imagery and machine learning. Water, 14(7), 1140.

Timadi, M. E., & Obasi, E. C. M. (2025). A zero trust hybrid machine learning algorithms for threat detection and Prevention with Explainable Threat Intelligence, UIJSLICTR, 15(1), pp.246-261.

Timadi, M. E., & Obasi, E. C. M. (2025). Integrating zero-trust architecture with deep learning algorithm to prevent structured query language injection attack in cloud database. *University of Ibadan Journal of Science and Logics in ICT Research,* 13(1), 52–62.

Twele, A., Cao, W., Plank, S., & Martinis, S. (2016). Sentinel-1-based flood mapping at high spatial resolution: A case study in Germany. *Remote Sensing*, 8(6), 456.

Uso, N. O., & Onye, O. J. (2025). Performance evaluation of five probability distribution models for the analysis of flood data at River Niger. FUDMA Journal of Sciences, 9(4), 307–323.

Wang, Y., et al. (2019). Flood detection using remote sensing images: A review. *Remote Sensing*, 11(11), 1344.

Zhang, H., Chen, Y., & Ma, J. (2021). Deep learning for environmental monitoring: Applications and challenges. IEEE Access, 9, 86143–86157.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2019). Deep learning in remote sensing: A review. IEEE Geoscience and Remote Sensing Magazine, 5(4), 8–36.