Deep Learning and Distributed Image Classification of Galamsey Activities Using CNNs and Apache Spark

SAT-5165: Introduction to Big Data Analytics

Instructor: Dr. Neerav Kaushal

Prince Tawiah
Department of Applied Computing
Michigan Technological University
Email: ptawia3@mtu.edu

***Abstract*—Illegal gold mining—locally known as *galamsey*—is a major environmental and socio-economic threat in Ghana. Manually monitoring satellite imagery for evidence of galamsey is slow, subjective, and resource-intensive. This project develops a convolutional neural network (CNN) classifier, supported by distributed preprocessing with Apache Spark, to distinguish galamsey from non-galamsey regions in satellite imagery. A custom dataset of 200 screenshots from Google Maps and Google Earth was collected, tiled into more than 15,000 image patches, and augmented to increase variability. The final model achieves a test accuracy of 96.69% with an area under the ROC curve (AUC) of 0.996, demonstrating strong separability between the classes despite the limited number of original images. The work provides an end-to-end framework—from data collection to evaluation—that can be extended for large-scale environmental monitoring.**

***Index Terms*—Galamsey, illegal mining, convolutional neural network, Apache Spark, big data analytics, remote sensing, image classification.**

I. Introduction

Galamsey, an informal term for illegal small-scale gold mining in Ghana, has become a significant driver of environmental degradation. Forest reserves are cleared, river systems are contaminated with heavy metals and sediment, and arable land is converted into open pits. These activities threaten ecosystems, public health, and long-term economic stability.

Remote sensing technologies, particularly satellite imagery, provide a powerful means for monitoring land-use change over large areas. However, traditional monitoring approaches depend heavily on human experts visually scanning images, which is slow, expensive, and difficult to scale. Recent advances in deep learning, especially convolutional neural networks (CNNs), have transformed image recognition and offer a promising approach for automating the detection of galamsey-like patterns in imagery.

In parallel, big data platforms such as Apache Spark have become the de facto standard for distributed data processing. Integrating Spark with deep learning enables scalable pipelines for ingesting, transforming, and analyzing massive image collections.

This project leverages these developments to design, implement, and evaluate a CNN-based classifier for galamsey detection, with Spark-based preprocessing to simulate deployment in a big-data environment.

II. Background and Related Work

CNNs have achieved state-of-the-art performance in large-scale image recognition tasks such as ImageNet classification [1], [2]. In remote sensing, they have been used for land-cover mapping, deforestation monitoring, and urban change detection, frequently outperforming traditional feature-engineering approaches.

Distributed data processing frameworks like Apache Spark provide primitives for large-scale data ingestion, transformation, and caching, and have been successfully combined with deep learning libraries to handle large image and video datasets [3]. However, very few publicly documented systems focus specifically on illegal mining detection, in part because labeled datasets are scarce and imagery can be sensitive.

This project contributes a practical, reproducible pipeline for galamsey classification that combines CNN-based image analysis with distributed image preprocessing. While the dataset is modest, the methodology is designed to scale to larger archives.

Recent work has applied CNNs to a variety of remote-sensing tasks, including deforestation monitoring, land-cover mapping, and urban expansion analysis, demonstrating that learned hierarchical features often outperform traditional hand-crafted descriptors. For example, CNN-based models have been used to detect forest loss and agricultural change from multispectral imagery, and to map informal settlements at high spatial resolution. Compared to these studies, the present work focuses specifically on illegal small-scale mining in Ghana and emphasizes a practical, patch-based pipeline that can be scaled via Apache Spark to much larger image archives.

III. Dataset Description

A. Image Acquisition

Because no labeled galamsey dataset was publicly available, a custom dataset was curated. A total of 200 base images were collected:

- 100 images depicting clear evidence of galamsey (open pits, exposed soil, sediment-laden rivers), and
- 100 images showing non-galamsey environments such as forests, farms, settlements, and clean river systems.

Images were obtained by taking screenshots and snapshots of publicly viewable Google Maps and Google Earth satellite imagery. Screenshots were captured at zoom levels where surface disturbances and vegetation patterns were clearly visible.

B. Ethical and Legal Considerations

All imagery was used solely for academic, non-commercial purposes within the SAT-5165 course. The screenshots do not contain personally identifiable information or restricted facility details, and no protected or commercial satellite datasets were accessed. The aim is to explore methodology for environmental monitoring rather than operational surveillance, consistent with ethical guidelines for remote-sensing research.

C. Image Characteristics and Variability

The base images vary in resolution (from approximately 800×800 to more than 3000×3000 pixels), geographic region and terrain type. Galamsey patterns themselves also vary: some regions exhibit clusters of small circular pits, while others show large open pits or discolored river segments. This variability makes the classification task challenging and motivates the use of deep learning models capable of learning complex spatial features.

D. Patch Size Selection

A patch size of 128×128 pixels was chosen as a compromise between capturing sufficient spatial context and maintaining a manageable input dimension for the CNN. Smaller patches (e.g., 64×64) risk losing critical spatial patterns such as the arrangement of pits along riverbanks or the contrast between disturbed and undisturbed areas. Conversely, much larger patches (e.g., 256×256 or more) would increase the number of parameters in the early convolutional layers and reduce the number of independent training samples obtainable from each base image. In preliminary exploratory experiments, 128×128 patches provided visually coherent regions that preserved key galamsey signatures while keeping the model lightweight enough for CPU-based training.

E. Patch Extraction

To transform the relatively small set of base images into a dataset suitable for CNN training, each image was tiled into non-overlapping 128×128 pixel patches. Patches were manually labeled based on the class of their parent image.

This process produced:

- 10,716 galamsey patches, and
- 4,412 non-galamsey patches,

for a total of 15,128 labeled patches. The imbalance arises because galamsey images generally contained larger contiguous disturbed areas than non-galamsey scenes, yielding more usable patches.

F. Train–Validation–Test Splitting

A critical methodological choice was to avoid data leakage between splits. Instead of randomly shuffling patches, the split was carried out at the *source image* level:

- 70% of base images were assigned to the training set,
- 15% to the validation set, and
- 15% to the test set.

All patches derived from a given base image were placed in the same split. This ensures that the model is evaluated on visually distinct scenes, providing a more realistic assessment of generalization. The final patch-level dataset is summarized in Table I.

TABLE I
Patch-level dataset statistics and split proportions.

Split	Galamsey	Non-galamsey	Total
Training	7,777	3,181	10,958
Validation	1,546	476	2,022
Test	1,393	755	2,148
Total	10,716	4,412	15,128

IV. Preprocessing and Spark Integration

A. Data Augmentation

To mitigate overfitting and increase diversity, extensive data augmentation was applied to the training patches only. The augmentation pipeline included:

- random rotations up to $\pm 20^\circ$,
- horizontal and vertical flips,
- width and height shifts of up to 20%,
- random zooms of up to 20%, and
- brightness and slight color perturbations.

After augmentation, the effective training set size increased from 10,958 to 43,832 patches (approximately a fourfold expansion), while the validation and test sets remained unchanged.

B. Normalization

All images were resized (if necessary) to 128×128 pixels with three RGB channels and their pixel intensities were normalized to the range $[0, 1]$. This standardization improves optimization stability and makes the model less sensitive to absolute intensity differences.

C. Apache Spark Pipeline

Apache Spark was used to orchestrate the preprocessing pipeline. Spark handled:

- distributed loading of the raw screenshots from storage,
- parallel patch extraction,

- mapping of patches to metadata (source image, class label, split assignment), and
- efficient caching of intermediate results.

Although the dataset size is moderate, using Spark demonstrates how the same code path could scale to much larger archives with minimal modification, aligning with the big data focus of SAT-5165.

V. CNN Architecture and Training

A. Model Architecture

The CNN architecture was designed to balance expressiveness and computational cost. The model accepts $128 \times 128 \times 3$ RGB inputs and consists of three convolutional blocks followed by dense layers:

- **Block 1:** Conv2D with 32 filters, 3×3 kernel, ReLU activation; batch normalization; max pooling with 2×2 window.
- **Block 2:** Conv2D with 64 filters, 3×3 kernel, ReLU; batch normalization; max pooling; dropout with rate 0.25.
- **Block 3:** Conv2D with 128 filters, 3×3 kernel, ReLU; batch normalization; max pooling; dropout with rate 0.25.
- **Dense Head:** Flatten; fully connected layer with 128 units and ReLU activation; dropout with rate 0.5; output layer with a single sigmoid unit for binary classification.

This architecture is capable of learning hierarchical texture features while remaining relatively lightweight (approximately 4.3 million trainable parameters), which is suitable for CPU-based training in a multi-VM environment.

B. Training Configuration

The model was implemented in TensorFlow/Keras and trained with the following configuration:

- optimizer: Adam with learning rate 0.001,
- loss function: binary cross-entropy,
- batch size: 32,
- epochs: up to 10,
- callbacks: early stopping (patience = 3) on validation loss, model checkpointing based on validation accuracy, and learning-rate reduction on plateau.

TABLE II
Summary of key training hyperparameters.

Hyperparameter	**Value**
Optimizer	Adam
Initial learning rate	0.001
Batch size	32
Maximum epochs	10
Loss function	Binary cross-entropy
Early stopping	Patience = 3 (monitoring validation loss)
Model checkpoint	Best validation accuracy
LR reduction	Factor = 0.5 (patience = 2)
Dropout rates	0.25 (Conv blocks), 0.50 (Dense)
Input patch size	$128 \times 128 \times 3$

The key training hyperparameters are shown in Table II. Training and validation metrics (accuracy, precision, recall, and loss) were recorded at each epoch to analyze convergence behavior.

VI. Experimental Results

A. Classification Report

Table III summarizes the performance of the trained model on the held-out test set in terms of precision, recall, F1-score, and support for each class, along with accuracy and macro/weighted averages.

TABLE III
Classification Report on Test Set

Class	Precision	Recall	F1-score	Support
Non-Galamsey	0.95	0.91	0.93	608
Galamsey	0.96	0.98	0.97	1540
Accuracy		0.96		2148
Macro avg	0.96	0.94	0.95	2148
Weighted avg	0.96	0.96	0.96	2148

The model attains an overall accuracy of 96%. Both classes achieve high precision and recall, indicating that the classifier is effective at identifying galamsey regions while also maintaining good performance on natural environments.

B. Confusion Matrix

Fig. 1 shows the confusion matrix on the test set. Out of 608 non-galamsey patches, 552 are correctly classified, while 56 are misclassified as galamsey. For the 1,540 galamsey patches, 1,511 are correctly identified and 29 are misclassified as non-galamsey. As shown in Fig. 1, the model correctly identifies the majority of galamsey patches, while a smaller fraction of non-galamsey patches are misclassified as galamsey, often in visually ambiguous regions with bare soil or seasonal vegetation changes.

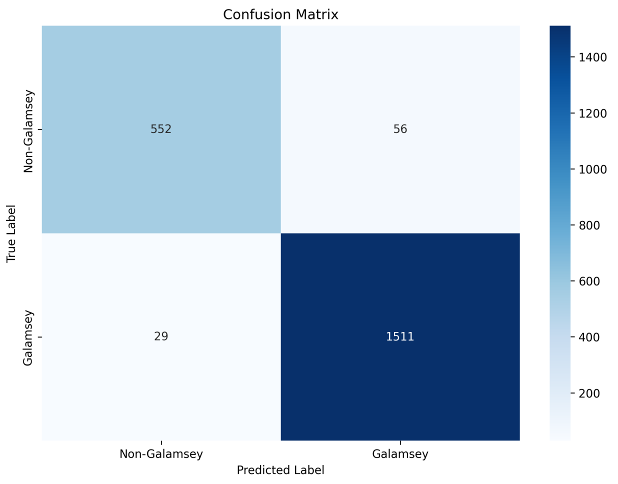

Fig. 1. Confusion matrix for galamsey vs. non-galamsey classification.

C. ROC Curve

The receiver operating characteristic (ROC) curve in Fig. 2 illustrates the trade-off between true positive rate and false positive rate across classification thresholds. The area under the curve (AUC) is 0.991, demonstrating excellent separability between the two classes.

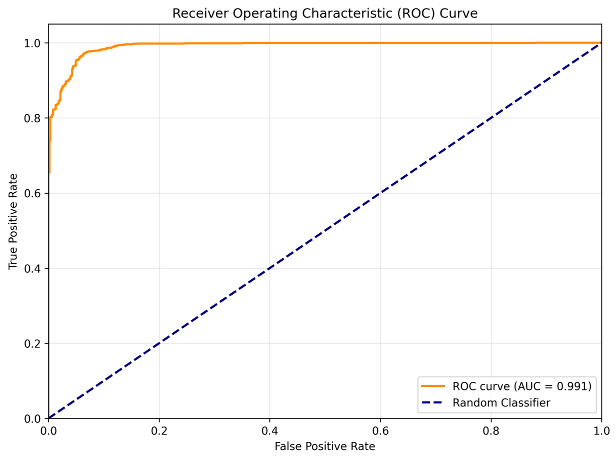

Fig. 2. ROC curve for the CNN classifier (AUC = 0.991).

D. Training History

Fig. 3 presents the training and validation curves for accuracy, loss, precision, and recall across epochs. The curves indicate steady improvement without severe divergence between training and validation metrics, suggesting that the regularization and augmentation strategies were effective in controlling overfitting.

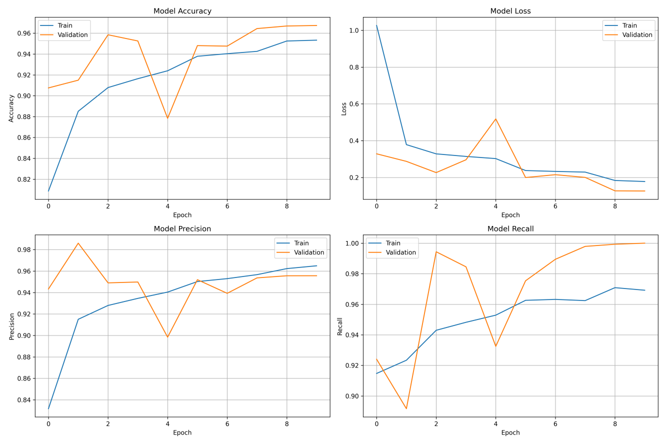

Fig. 3. Training and validation accuracy, loss, precision, and recall over epochs.

VII. Discussion

The results demonstrate that a relatively compact CNN can learn discriminative features for galamsey detection from a small but carefully constructed dataset. The high AUC and F1-scores indicate that the model is not simply memorizing the training data; it generalizes well to unseen scenes derived from different base images.

Several design decisions were crucial:

- **Source-level splitting** prevented overly optimistic estimates of performance that would arise if correlated patches from the same image appeared in multiple splits.
- **Patch-based modeling** enabled the network to focus on local patterns such as pits, exposed soil, and river discoloration instead of entire scenes.
- **Aggressive augmentation and dropout** helped mitigate overfitting, as evidenced by the close alignment of training and validation curves.

The patch extraction process resulted in a noticeable class imbalance, with galamsey patches constituting approximately 71% of the dataset and non-galamsey patches the remaining 29%. This reflects the fact that galamsey scenes often contain large contiguous disturbed regions, which yield more usable patches than visually heterogeneous natural environments. To mitigate potential bias, augmentation was applied symmetrically at the patch level, and performance was evaluated using precision, recall, F1-score, and the confusion matrix, rather than relying on accuracy alone. In future work, explicit class-weighting or resampling strategies could be incorporated to further address this imbalance.

At the same time, errors in the confusion matrix reveal limitations. Some natural landscapes with bare soil or seasonal changes are misclassified as galamsey, and some early-stage or subtle galamsey patterns are missed. These errors highlight the inherent ambiguity of certain regions and the limited diversity of the current dataset.

VIII. Limitations and Future Work

The present study has several limitations:

- The dataset originates from only 200 base images and may not fully capture the geographical and seasonal diversity of Ghana.
- Only RGB information was used; no multispectral bands were incorporated.
- The model performs binary classification only, without grading the severity of galamsey activity.

Future extensions may include:

- incorporating multispectral imagery from sensors such as Sentinel or Landsat,
- leveraging transfer learning with deeper architectures (e.g., ResNet, EfficientNet),
- extending the model to multi-class classification of disturbance severity,
- performing temporal analysis to track the expansion or reclamation of mining sites, and
- deploying the pipeline as part of a near real-time monitoring system on a larger Spark cluster.

IX. Conclusion

This project presented an end-to-end framework for detecting galamsey activities from satellite imagery using a CNN classifier and Spark-based preprocessing. Despite the limited number of base images, tiling, augmentation, and careful regularization enabled the model to achieve an accuracy of 96% and an AUC of 0.991 on the test set. The methodology demonstrates how deep learning and big data tools can be combined to support environmental monitoring and provides a foundation for future, larger-scale systems.

Acknowledgment

The author thanks Dr. Neerav Kaushal for guidance and feedback in the SAT-5165 course, as well as the Michigan Technological University computing environment that supported experimentation.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] M. Zaharia *et al.*, "Apache Spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[4] G. Hilson, "The environmental impact of small-scale gold mining in Ghana: Identifying problems and possible solutions," *The Geographical Journal*, vol. 168, no. 1, pp. 57–72, 2002.

[5]