The Ground Truth: Novel Approach for Landslide Prediction with a Deep Learning Framework

EAEV-0330-JR | Arnav Saraf | Canyon Vista Middle School | 8th Grade

FreeSVG. (2019). *Landslide Near the City* [Landslide Near the City]. FreeSVG. https://freesvg.org/landslide-near-the-city

Background Information

- **Landslides** are defined as the **movement** of a mass of **rock**, **debris**, or **earth** down a **slope**.
- **Landslides** cause about **$20 billion** of property damage, and **thousands** of people are **killed** by landslides **each year** worldwide.
- **Landslides** are typically **triggered** by high amounts of **rainfall** or **earthquakes**.
- In the **United States alone**, there are **25 to 50** deaths and over **$1 billion** in property damage each year due to **landslides**.
- There were **254 fatalities**, **397 injuries**, and **118 people missing** due to the recent landslides in **Kerala, India**.

Research Questions

- What is the **most important geographical feature** for a model to predict landslide susceptibility in a region?
- What is the **best model** for landslide prediction?

Hypothesis

- **Topography** will be the **most important** feature.
- **Convolutional Neural Networks** will be the **most effective** at predicting landslide susceptibility.

Engineering Goals

- **Ensemble accuracy** greater than **90%** at predicting the susceptibility.
- Expandable **framework** that uses **multiple features** as input to **accurately predict** landslide susceptibility values.

Variables

Controlled	Independent	Dependent
• Satellite Datasets • USGS Landslide Inventory • Random State	• Geological Feature • Model Optimizers	• Accuracy of Model • Performance Metrics

Key Terms

- **Convolutional Neural Networks (CNNs):** A type of neural network that processes grid-like data and consists of multiple specialized layers.
  - **Convolutional Layers:** Consists of a filter kernel that identifies important features using dot products. Learns spatial relations.
  - **Dense Layers:** Consists of fully connected neurons that perform a matrix multiplication of a input 1D vector, trainable weights, and biases. Learns global relations.
  - **Activation:** Final layer that introduces non-linearity; allows learning of complex relationships. Applies nonlinear functions to data. Common functions include **ReLU, Sigmoid, Softmax, TanH**.
- **RMSE:** ***Root Mean Squared Error***. A loss function meant for regression models. It is the square root of **Mean Squared Error (MSE).** It penalizes large errors more severely because of squared function.
- **SCCE:** ***Sparse Categorical Cross Entropy***. A loss function for multi-classification models, computes cross-entropy loss by converting integer labels to one-hot vectors internally. Ideal for high class classifications.

Initial Exploration

At the start of the project, I decided to split the project into **two halves**, **remote sensing** and **local sensing**. For the local sensing, I built an **arduino-python interface** to log **soil moisture, water flow rate (inches of rain), and vibration data** to a csv file. However, even though the interface worked, there was a **lot of noise in data** because of low quality sensors, and the **approach was impractical** for real world scenarios. Hence this project was **focused on remote sensing.**

Images taken by the student researcher

Prototyping

Prototype 1	Used XGBoost model **Regression**	Low regression accuracy (**±30%**), prone to overfitting.
Prototype 2	XGBoost, Random Forest, LightGBM ensemble **Regression**	Low regression accuracy (**±18%**). Overfitting.
Prototype 3	**CNN** (Convolutional Neural Network). **Regression**	Low computational efficiency, low accuracy. (**±20%**)
Prototype 4 Final Model	**CNN, 25 Bucket Classification**	High accuracy (**±3%**), high efficiency.

Methodology Overview

Four separate CNN models will be **created** and **trained** using:

- **NAIP Vegetation Index** (NDVI).
- **NAIP Satellite Imagery.**
- **USGS Ned10m Topographical** image.
- **Slope** image **derived** from Ned10m topography.
- **USGS Landslide Susceptibility** Map

The **outputs** of these models will then be **used as input features** to an **XGBoost regression** model. The CNN models will be performing **feature processing** for **XGBoost**. I call this method **ConvEx** (Convolutional Extraction).

Prerequisites

- **Google Colab (Pro recommended)**
- **NVIDIA L4 or better GPU (Colab)**
- **At least 32GB ram (Colab)**
- **My GitHub Repository**
- **Virtual Studio Code**
- **Google Earth Engine API Key**

Procedure

1. Collect diverse datasets (topography, slope, NDVI, rainfall, satellite imagery) from US landslide-prone and non-prone regions.
2. Normalize and process the data into formats suitable for model input.
3. Design and train custom CNN models for each dataset (NDVI, slope, topography, satellite imagery, USGS Susceptibility). *Use Google Colab.*
4. Test and fine-tune CNN models using accuracy metrics. *Use Colab.*
5. Use CNN predictions to generate features in format bucket.confidence.
6. Train and test the XGBoost regression model on the generated features and rainfall data to predict landslide susceptibility on a 0–100 scale.
7. Evaluate model performance using RMSE and R^2 metrics.
8. Visualize results with heatmaps, residuals, error distributions, and predicted vs. actual values.

Please Note:
GitHub repository available for access to all code and data I created.

Flowchart

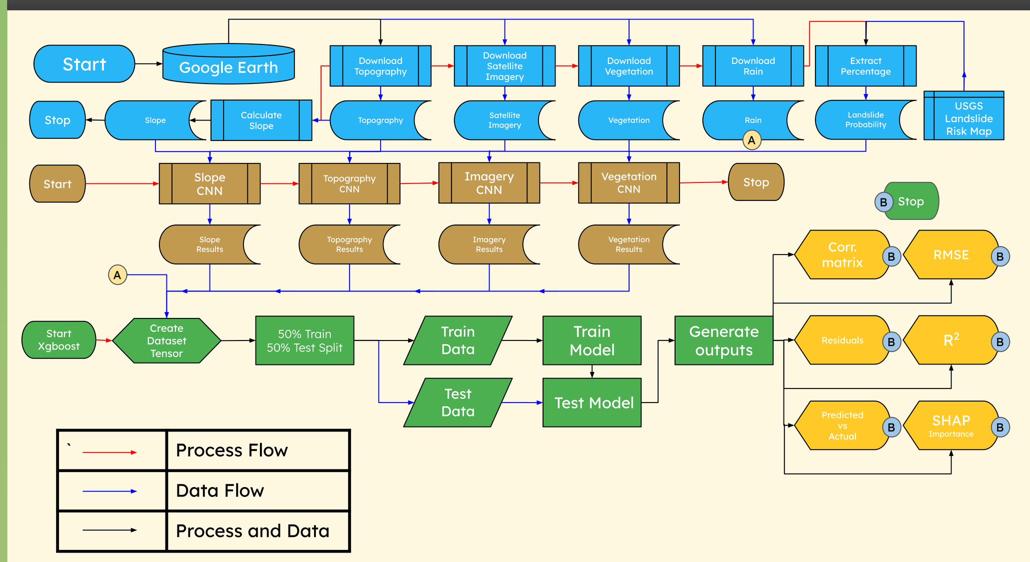


Diagram created by student researcher

Results

Model	Accuracy	Loss	SHAP Importance
Topography	**98%**	**SCCE:** 0.09	**30.8671**
Slope	**97%**	**SCCE:** 0.1	**2.1703**
Satellite Imagery	**99%**	**SCCE:** 0.06	**0.3174**
NDVI	**66%**	**SCCE:** 1.09	**0.4986**
Ensemble	**99%**	**RMSE:** 4.9	**N/A**

Conclusions

1. **CNN** are better suited for this task as they **better handle** and learn from **spatial relationships.**
2. **Topography** was the most **important factor** in predicting **landslides**.

The **engineering goals** of building an **expandable framework** and **+90% accuracy** were **achieved**. The **proposed framework** was able achieve an astounding **99% accuracy** in predicting **susceptibility values.**

Comparison and Future Work

Comparison:
Previous methods for predicting landslide susceptibility, such as the USGS landslide inventory, relied on geologists' calculations. The proposed deep learning framework achieved 99% of the accuracy of the geologists' evaluations while fully automating the process. This significantly improves scalability of the proposed method.

Future Work:

- **Develop a Mobile/Web App**
- **Highlight Prone Areas:** Visually emphasize high-risk regions in images.
- **Integrate Real-Time Data:** Include live inputs from live satellite datasets and weather APIs.
- **Automate Data Retrieval:** Fetch weather and satellite data from APIs for seamless user experience.

Sources Cited

GeeksforGeeks. (2024, March 14). Introduction to Convolution Neural Network. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-convolution-neural-network/

Jain, A. (2024, February 12). All about convolutions, kernels, features in CNN - Abhishek Jain - Medium. Medium; Medium. https://medium.com/@abhishekjainindore24/all-about-convolutions-kernels-features-in-cnn-c656616390a1

Ober, H. (2023, June 23). UCLA geologists are using artificial intelligence to predict landslides. UCLA. https://newsroom.ucla.edu/releases/artificial-intelligence-can-predict-landslides

Saha, S. (2021, November 28). XGBoost vs LightGBM: How Are They Different. Neptune.ai. https://neptune.ai/blog/xgboost-vs-lightgbm

Wang, Y., Gao, H., Liu, S., Yang, D., Liu, A., & Mei, G. (2024). Landslide detection based on deep learning and remote sensing imagery: A case study in Linzhi City. Natural Hazards Research. https://doi.org/10.1016/j.nhres.2024.07.001

Youssef, K., Shao, K., Moon, S., & Bouchard, L.-S. . (2023). Landslide susceptibility modeling by interpretable neural network. Communications Earth & Environment, 4(1). https://doi.org/10.1038/s43247-023-00806-5

Acknowledgements

Mr. Vikram Sabnis - Provided guidance in the selection of AI models, the optimizing the hyperparameters of models, creation of test and train datasets, and evaluating the performance.

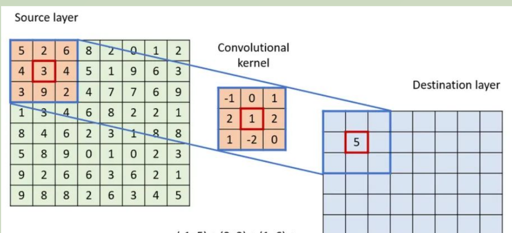
Prof. Prashant Joshi - Provided project guidance and mentorship for the entire project.

GitHub Repository

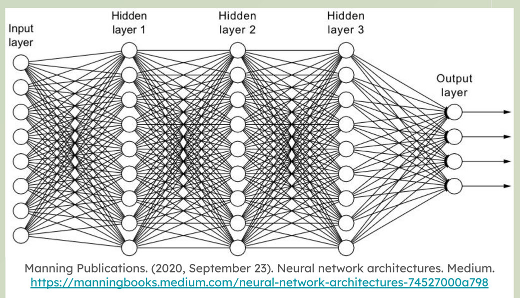
All the code developed and utilized for this project is readily accessible in the designated GitHub repository. This repository contains the complete source code, including scripts, models, and any other necessary dependencies. This ensures others can review, replicate, and build upon the work without needing to recreate it from scratch.
Licensed under GPL - 3.0

Visuals

Model Diagrams

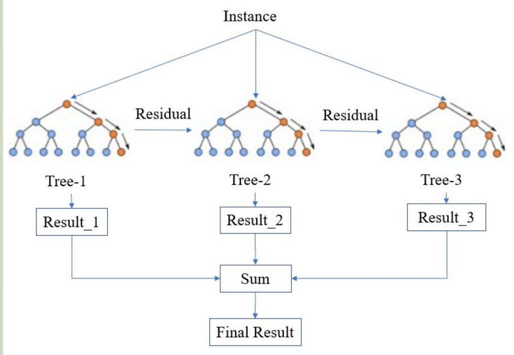


Jain, A. (2024, February 12). *All about convolutions, kernels, features in CNN.* Medium. https://medium.com/@abhishekjainindore24/all-about-convolutions-kernels-features-in-cnn-c656616390a1

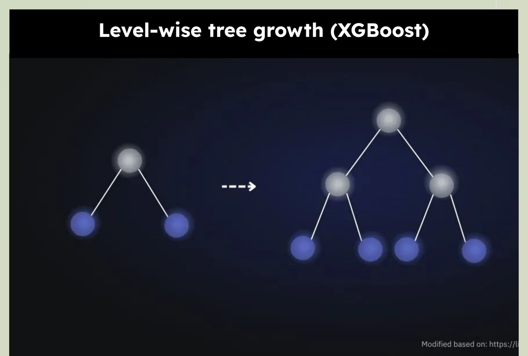


Manning Publications. (2020, September 23). Neural network architectures. Medium. https://manningbooks.medium.com/neural-network-architectures-74527000a798

Fully Connected Dense Layer Architecture

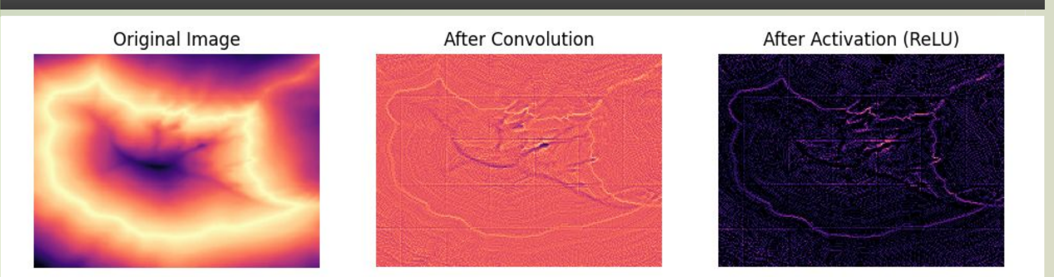


Chen, Y., Hong, L., Feng, Y., & Dai, X. (2021). Simplified structure of XGBoost [Image]. In *ResearchGate.* https://www.researchgate.net/figure/Simplified-structure-of-XGBoost_fig2_348025909



Neptune.ai. (n.d.). XGBoost vs LightGBM: How Are They Different. https://neptune.ai/blog/xgboost-vs-lightgbm

Convolution Example



As you can see in the picture above, each layer of a CNN will reduce the image to a more meaningful image for the AI. For example, this time the AI made a ridge map that shows all sudden edges

Image created by student researcher using matplotlib

Final Model Statistics

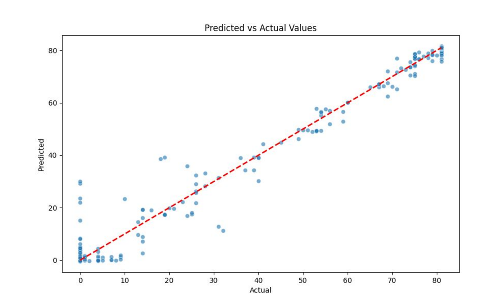


Image created by student researcher using matplotlib

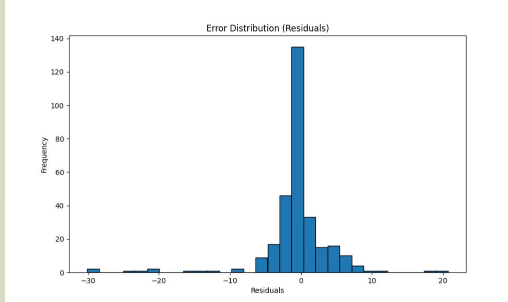


Image created by student researcher using matplotlib

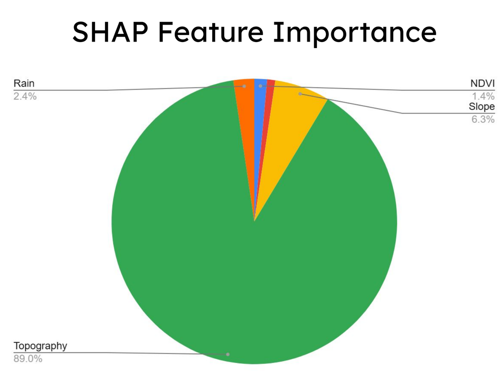


Image created by student researcher using google sheets

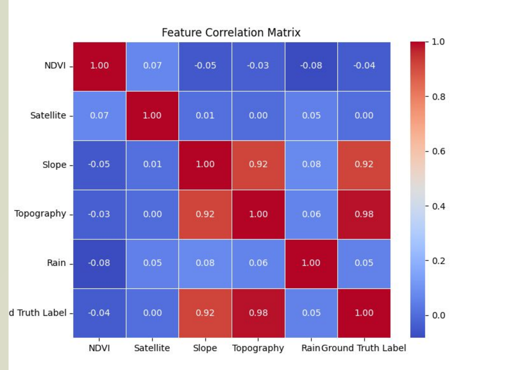


Image created by student researcher using matplotlib

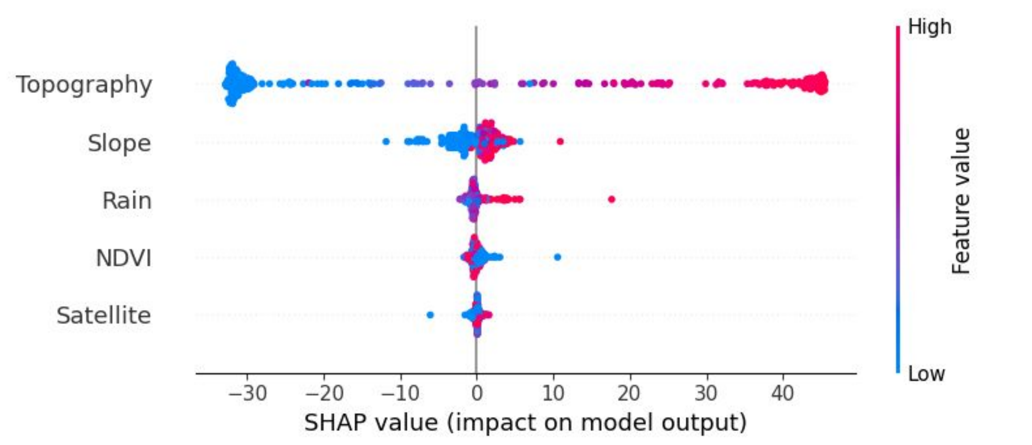
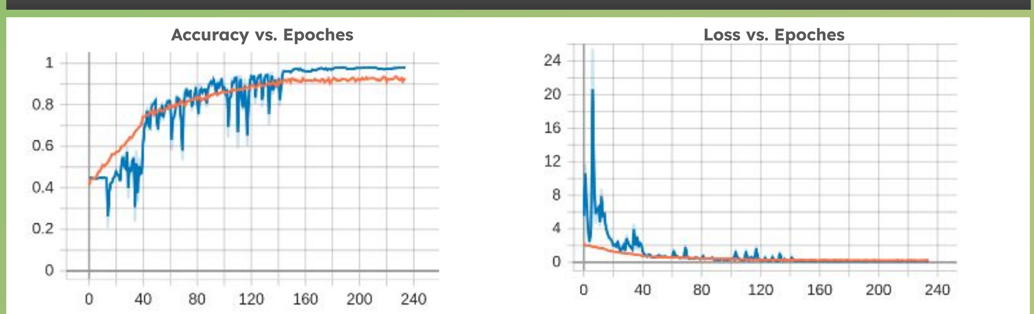


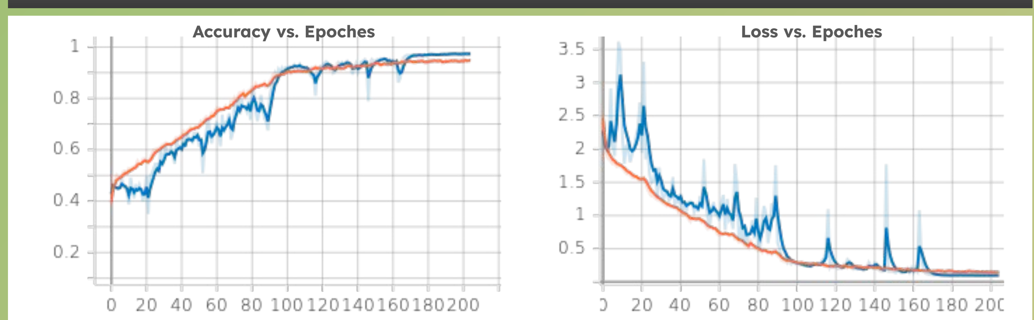
Image created by student researcher using matplotlib

Topography CNN Stats



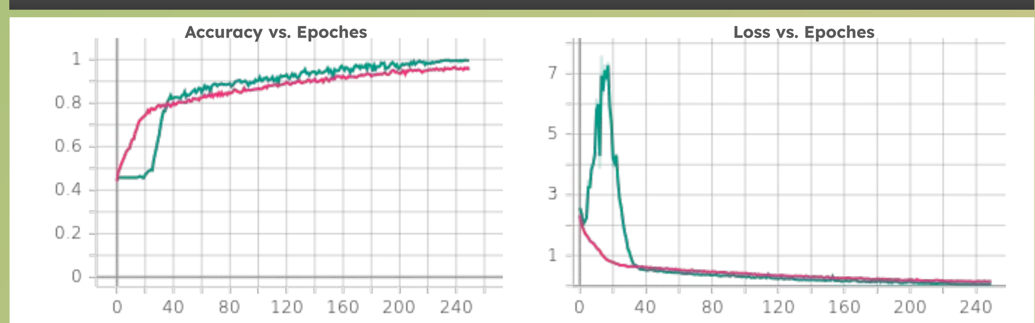
Key Takeaways
98% Accuracy, Loss 0.09 SCCE
Validation accuracy is greater than training accuracy, which means that the model generalizes to unseen data well. Topography is a good way to predict landslide susceptibility.
Graphs created by student researcher.

Slope CNN Stats



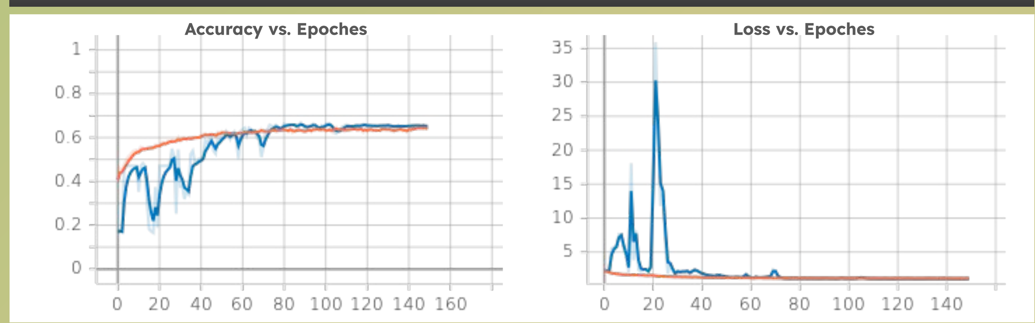
Key Takeaways
97% Accuracy, Loss 0.1 SCCE
Validation accuracy is greater than training accuracy, which means that the model generalizes to unseen data well. Slope is a good way to predict landslide susceptibility.
Graphs created by student researcher.

Satellite Imagery CNN Stats



Key Takeaways
99% Accuracy, Loss 0.06 SCCE
Validation accuracy is greater than training accuracy, which means that the model generalizes to unseen data well. Satellite imagery is a good way to predict landslide susceptibility.
Graphs created by student researcher.

Vegetation Index CNN Stats



Key Takeaways
66% Accuracy, Loss 1.09 SCCE
Validation accuracy is on par with training. However because of early stopping and the low accuracy after testing, vegetation is not a accurate way of deducing landslide susceptibility.
Graphs created by student researcher.

Dataset Samples

Satellite Imagery

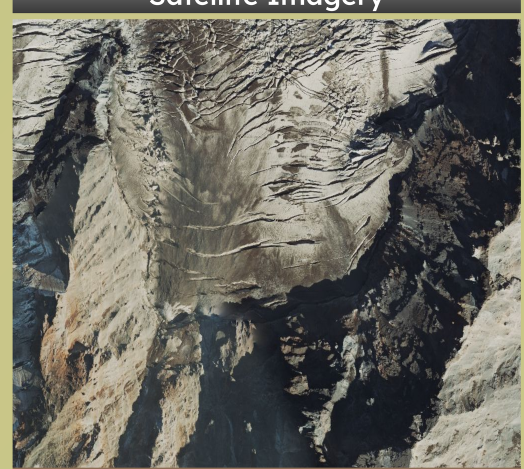


Image derived from USGS NAIP 10m Earth Engine Dataset

Satellite Imagery Processed

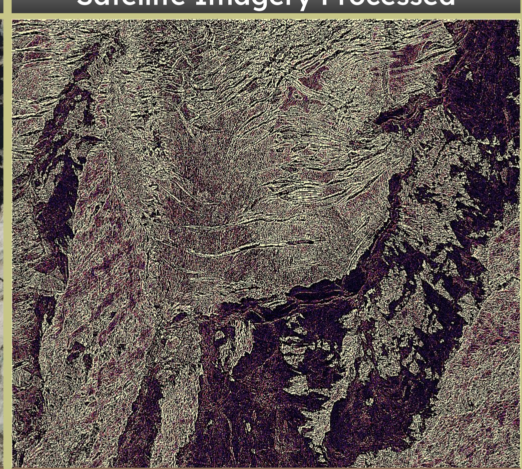


Image created by student researcher

Topography

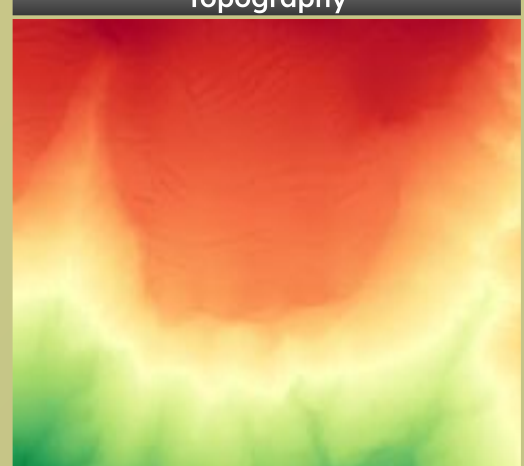


Image derived from USGS Ned10m 3DEP Earth Engine Dataset

Topography Processed

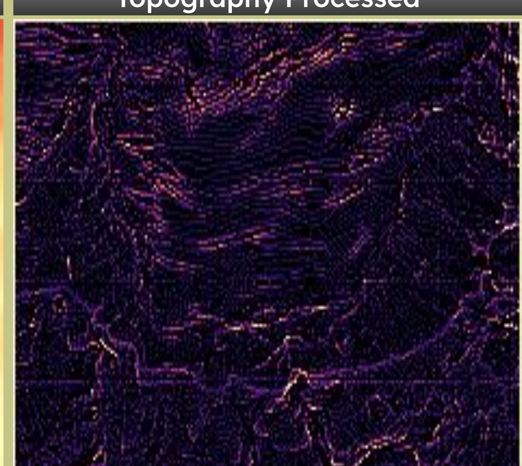


Image created by student researcher

Dataset Samples

Slope

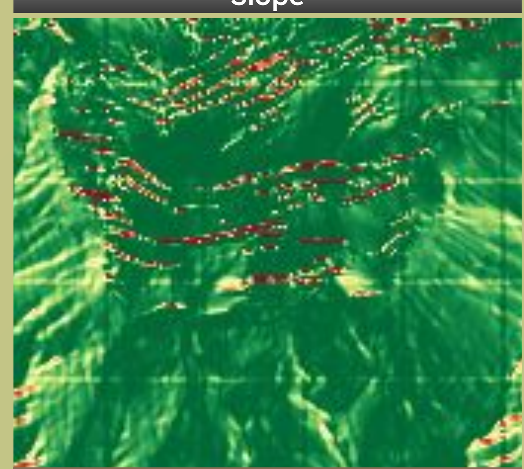


Image created by student researcher

Slope Processed

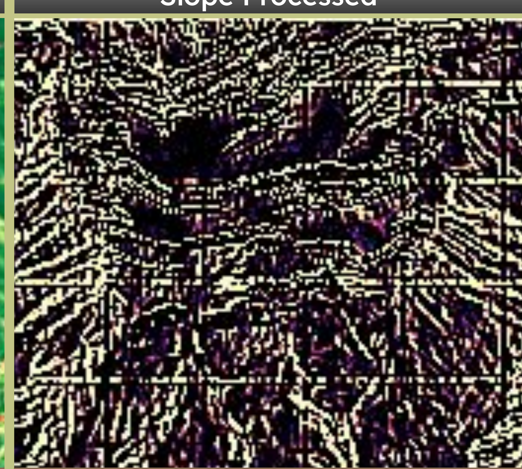


Image created by student researcher

Vegetation Indices

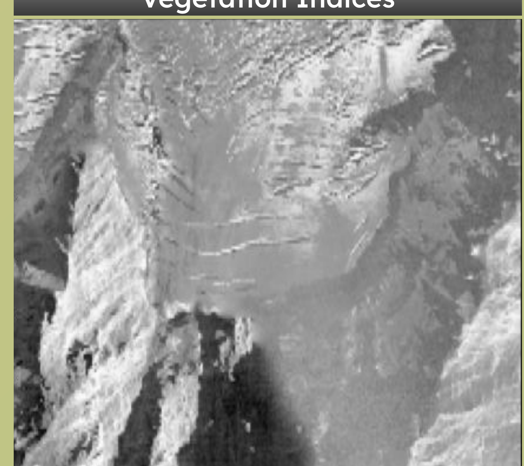


Image derived from USGS NAIP 10m Earth Engine Dataset

Vegetation Processed

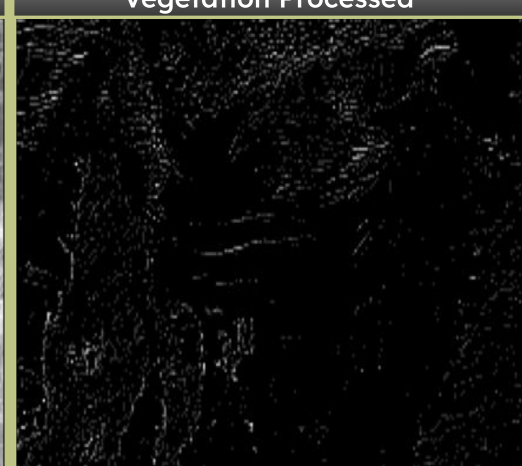


Image created by student researcher