Landslide Detection Through Deep Learning - Group 6

Harold Haugen
School of Data Science
The University of Virginia
waa4bq@virginia.edu

Max Pearton
School of Data Science
The University of Virginia
mjp2da@virginia.edu

Elena Tsvetkova
School of Data Science
The University of Virginia
rrm3nh@virginia.edu

Daniel Sery
School of Data Science
The University of Virginia
dms3gv@virginia.edu

I. Abstract

Landslide detection using deep learning models is a challenging yet critical task for disaster management and mitigation. This study explores the application of EfficientNetB3 and other Convolutional Neural Network (CNN) architectures to classify satellite imagery into landslide and non-landslide categories. We employed techniques such as data augmentation (e.g., saturation, contrast, cropping adjustments, etc.) and fine-tuning with layer unfreezing to improve model performance. Among tested architectures, EfficientNetB3 demonstrated superior results in identifying landslide features when trained with top-level layer unfreezing and transfer learning strategies. Grad-CAM visualizations provided valuable interpretability by highlighting spatial regions critical to the model's predictions. Our results indicate that diverse datasets, careful augmentation, and transfer learning significantly enhance the model's ability to generalize, offering a promising approach to landslide detection tasks.

II. Literary Review

A. Image Classification w/ Deep Neural Networks (Satellite Imagery)

There is broad understanding that the use of CNNs is highly desired in the realm of image and object recognition due to their "ability to automatically recognize interconnected contextual clues inherent in image classification challenges". This is accomplished by a CNN's design of deep structured layers where each layer is able to self-learn image features that are noted as "symbolic representations of pixels in the original image" [4].

Specific to landslide classification using CNN, researchers found CNNs trained on a combination of inventories rather than local datasets have better generalization performance. Their approach was to first develop CNNs by "training and validating on event landslide inventories in four regions after earthquakes and/or extreme meteorological events." These teams then "trained CNNs to map landslides triggered by new events spread across different geographic regions" [5].

Regarding satellite imagery classification, reviewed research highlighted a team using a flexible CNN known as SAT CNN for improved scene categorization. This approach utilized convolutional layers with smaller convolutional kernels and deeper structures which had success in extracting features to solve for intra-class variability. Feature extraction and its importance was a notable focus area within the paper, with it stating that "[t]he features that have been extracted from the lower layers can be utilized as training features due to their ability to provide more advanced features." Their approach also noted the use of an ensemble classifier in conjunction with a Particle Swarm Optimization Classifier (PSO), with positive success. PSO may be used for either hyper-parameter optimization or feature selection and differs in design from gradient descent. This may be a tool to look into further during the course of the project. The approach covered in the paper also spoke to the importance of converting satellite imagery to greyscale during pre-processing for better performance [4]. "With regard to the image color shift from RGB to greyscale, and the use of smaller convolutional layers, based on our limited experience our team felt that these attributes may not offer large improvements in performance, but these settings will be considered as part of our model design and performance evaluation." [4]

III. Project Proposal

A. Overview

This project focuses on leveraging advanced machine learning techniques, particularly convolutional neural networks (CNNs), for the classification of satellite imagery into two categories: landslide and non-landslide regions. The primary objective is to develop a robust and efficient model capable of identifying landslides across diverse geographic terrains and environmental conditions. Given the importance that accurately detecting landslides has to mitigating risks and supporting disaster management, the project emphasizes the use of transfer learning and hyperparameter tuning to enhance model performance in this regard.

The approach involves testing and comparing multiple deep learning architectures, including baseline models and pre-trained CNNs such as EfficientNetB3, ResNet50, DenseNet121, and MobileNet. Transfer learning is a central strategy, with pre-trained weights on ImageNet providing a strong foundation for fine-tuning on landslide-specific data. Additionally, the project incorporates techniques such as data

augmentation, frozen layer training, and hyperparameter optimization to improve generalization and adapt the models to the specific challenges of satellite imagery classification.

To evaluate model effectiveness, the project uses a diverse dataset with training, validation, and testing splits, ensuring coverage of various landscapes, elevations, and resolutions. Special emphasis is placed on testing the models with independently sourced satellite images to simulate real-world scenarios. To gain deeper insights into model decision-making, Gradient-weighted Class Activation Mapping (Grad-CAM) was used to visualize the areas of images that the models deemed most relevant for classification. This technique provided an interpretability layer, helping identify whether the models focused on features directly associated with landslides or unrelated regions. Grad-CAM analysis revealed instances where the models correctly identified landslides based on key features such as ground disturbances or vegetation patterns, as well as cases where misclassifications occurred due to environmental noise or overlapping visual characteristics.

Preliminary results indicate promising improvements in training and validation accuracy, though challenges remain in achieving high performance on the diverse test set, particularly for regions with distinct environmental characteristics such as deserts and low-elevation areas. The findings, coupled with insights from Grad-CAM evaluations, provide a strong foundation for continued refinement. Future work will focus on enhancing model generalization, addressing dataset imbalances, and further utilizing Grad-CAM to guide adjustments in training and model architecture.

B. Motivation

As part of our initial assessment of this problem, our team identified that there are many government and academic research projects focused on developing susceptibility models and maps with the objective to "quantify the spatial variability in landslide potential" and for "showing where these hazards (e.g., landslides) are more and less likely, which is useful for risk-reduction and land-use planning." [1] As part of this process, these teams rely on the use of accurate inventories to train and test data models, and to evaluate and refine modeling assumptions.

Our primary motivator is the accurate classification of landslides. To properly develop models, research requires both accurate inventories of landslide locations and their corresponding dates. However, full information is not always available, particularly due to a lack of reporting of smaller events. We believe introducing a deep learning solution which could automatically assess sections of terrain would be valuable. The objective of this solution would be to a) identify landslide events in remote areas regardless of timing, and b) to then provide a rough age classification based on visual markers that may provide valuable insights to future susceptibility models and predictions.

The concern on time/age was further highlighted in "Parsimonious High-Resolution Landslide Susceptibility Modeling at Continental Scales" article [1], where it was noted that there may be over-representation of recent landslide events given advancements in technology and a higher frequency of documenting observed landslides that may result in bias. As further stated; "Although susceptibility maps do not explicitly address frequency or timescales, they are not immune to observational biases due to the range of timescales represented by the underlying landslide inventories used to create them. For example, due to increasing sophistication and access to monitoring technology (e.g., smart phones and satellites), there may be reporting bias in which more recent landslides that have impacted the built environment may be better represented in inventories than events that occurred hundreds or thousands of years ago in more-remote locations" [1].

Mapping landslides via identifying surface features in satellite imagery is not a novel approach, especially as Earth observation (EO) satellites have increased in number and quality. However, the challenge lies with the wide variation in surface features, data, geology, and so on, which can make assessment difficult, and thus largely a manual or, at best, semi-automated task. Modern approaches increasingly make use of trained Convolutional Neural Networks (CNNs) to map landslides from topographical data faster and more effective than by traditional means. However, one issue remains that many of these models struggle to be adapted to new events without having been pre-trained to an existing, geographically local inventory [6].

As outlined in the 'Intended Experiment' section, our objective is to develop a deep learning model using assorted Convolutional Neural Network (CNN) architecture-based methods alongside other multi-modal data sources to identify and temporally classify a landslide inventory using freely attainable satellite imagery.

C. Data Collection

Our dataset will be comprised of a combination of aerial imagery of landslide and non-landslide events, landslide identification inventories, and associated event variables (e.g., date, coordinates, elevation, slope, etc.) from these inventory and other sources.

D. Intended Experiments

As part of our model development efforts and performance assessment, we intend to integrate and research the following machine learning methods:

1) ***Various CNN Architectures and Parameters****:* As part of the design phase, we'll study the effects of modifying the fundamental structure of the model's architecture (e.g., depth of layers, kernel size, mixture of convolutional and dense layers, use of inception modules, etc.), modification to input images through grey-scaling and augmentation, use of various pre-trained models and transfer learning, and hyperparameter tuning (e.g., across optimization, loss, normalization methods). In addition, we learned of the use of different optimizers and one in particular, 'Particle Swarm Optimizer' was cited as a good design choice when dealing with image classification that we will review further if able.

*2) **Transfer Learning**:* Transfer Learning is a fundamental method of improving model performance without the need for more powerful hardware, more training data, or longer training periods. By making use of pre-trained and related, yet more generalized models, we can use previously learned features on images to reduce the overall amount of training necessary for our specific task. By freezing the imported model's lower layer weights, we get better performance in recognizing elements such as edges or complex patterns at the initial layers. We can then apply additional custom layers to allow for task-specific classification, then fine-tune them by unfreezing at the higher layers. In our case, several teams and papers have made use of ResNet, a popular architecture with several branches that tends to outperform simpler CNNs for image classification, including in satellite imagery classification. We can also attempt to use more closely related models, such as BIFOLD's BigEarthNet v2.0 (which has a ResNet-50 based version), trained specifically for classifying satellite imagery I II.

*3) **Multi-Modal CNNs**:* From O' Reilly's Hands On Machine Learning textbook, it explains that instead of having to flatten input images to 1D before feeding into a neural network, utilization of a CNN allows each layer to be represented as 2D making it easier to match neurons with their inputs. However, we will experiment with utilizing multiple data modalities or data types besides images. This will parameter data such as slope, elevation and weather or temperature data at the time of the satellite image. We will also investigate the opportunity to bring geological data, such as whether an earthquake occurred during the period. Additional possibilities include incorporating image captions or embedded text, as well as potentially leveraging audio data, to support landslide prediction. In other words, instead of having a single input such as a single image of a landslide, we will experiment with the addition of weather data to start, and then explore other opportunities.

*4) **Grad-Cam**:* Gradient-weighted Class Activation Mapping is used to highlight regions in an image that the model focuses on. It uses the gradients of any target class flowing into the convolutional layer to produce a coarse localization map, indicating important areas for prediction. We will visualize which regions of the satellite images were most influential in the prediction which will make the model's decisions more interpretable. By finding which regions are most influential, we can enhance trust in the model by showing that the network is focusing on areas with landslide risk. We will train a CNN model on satellite images to classify regions as prone to landslides or not. We will then use Grad-CAM to generate image heatmaps to highlight areas the model considers important. We will then overlay the images to inspect the regions and see which areas contributed most to the prediction.

IV. Methods

A. Imagery Data and Processing

To address our imagery and data requirements for training and validation, we researched several pre-identified open-source image repositories that have been used in prior landslide segmentation research. After further evaluation of image quality and classification accuracy over several data sets (e.g., is a landslide clearly detectable by our team), we primarily used images saved in the CAS Landslide dataset. This data was compiled by the Chinese Academy of Sciences and includes a large collection of landslide imagery from satellites and unmanned aerial vehicles (UAVs) (20k images). The dataset provides imagery from multiple regions in China and Japan, making it a valuable addition to our global dataset for landslide detection. Individual repositories provide images at different elevations, resolution and saturation levels and provide a variety of image styles.

Specifically, our selected data source was made up of:

- Longxihe (UAV): ˜2.4k (512x512 px) high resolution, low contrast, low saturation images.
- Lombok (Sat): ˜400 (512x512 px) high elevation, high contrast images. Some mixed clouds and low probability landslide images intermixed.
- Moxitown (UAV) .2m resolution: ˜1.6k (512x512 px) high resolution, low elevation, high contrast images.
- Hokkaido Iburi-Tobu (Sat): ˜1.4k (512x512 px) high elevation, high contrast images.

For other aerial images to populate the non-lanslide class, we identified and captured images from the following repositories:

- DeepGlobe Land Cover Classification Dataset: Kaggle Image repository with ˜1.1k (2444x2444 px) images representing urban, agriculture, rangeland, forest, water, and barren locations at high resolution and high elevation [8].
- Wildfire Prediction Dataset (Satellite Images): Kaggle Imagery repository with ˜42k (350x350 px) images separated into two classes, wildfire and non-wildfire image sets. Wildfire images sets are set in forested areas, whereas non-wildfire sets are a mix of urban and mixed terrain images. Images are high elevation and high contrast [9].
- Satellite-Image-Deep-Learning: Hugging Face SODA-A comprises 2513 high-resolution images of aerial scenes at (4800x2744 px) [10].
- NASA Earth Observatory and related landslide articles. [11].

Because of the homogeneous nature of the above landslide images and our objective to have data include relatable and known landslide events in the United States and abroad, we further used the following landslide inventories sourced from a) NASA (+3.9k) [3] and b) the USGS (+600k) [2] to identify real world landslide events that could be image captured and associated to other data features. Latitude and longitude, as well as event date were used to search historic images within Google Earth in order to acquire high-resolution imagery close to the actual event date, as well as before and after combinations if able. Other data useful in this approach were 'event id', 'confidence of extent and nature of landslide',

'landslide size', 'rating', 'elevation', and 'slope' among other data points.

1) Pre-processing: For the training, validation and testing phases of modeling we created datasets of varying size and complexity to determine their impact to model performance as described further under the Methods section below. With that stated, the data process to prepare image data for model input included:

- Saving of .tif, .jpg and .png image files into a secure DropBox cloud repository for storage and sharing.
- All images not in `.jpg` format were transitioned to .jpg in order to comply with Keras requirements related to the `('image_dataset_from_directory()')` method. Supported image fomats in Keras include .jpg, .png, .bmp, .gif. This was performed using Photoshop batch transformations on the data housed in DropBox.
- Large images within the DeepGlobe and Hugging Face 'Satellite' data were cropped, saturated, and reduced in pixel size between 800 and 1000 px for better memory efficiency and workflow.
- Images were collected into file structures to represent the number of classes needed, e.g., landslide and non-landslide for a two classification model. These file structures were zipped and a URL created to pull images into our coded notebooks.
- As part of image pre-processing within the Keras workflow, each image was set to 300px or similar size.

2) Developed Training, Validation, and Test Sets: Our team developed the following datasets for model experimentation. Data was structured into two class directories; 0: Landslide and 1: Non-Landslide:

Fig. 1: Example Landslide Image: NASA Sourced, Debris flow near the city of Maracay, Venezuela Sept. 9, 2020 Coordinates: 10.321638, -67.64242, Google Earth [3]

Dataset	Type	Total Size	Class Size	Ref.
Set 1: Images from the Longxi River UAV set, with non-landslide images sourced from DeepGlobe Land Cover Classification data.	Train/ Validate	2000	0: 1000; 1: 1000	[7], [8]
Set 2: Images from the Longxihe, Lombok, Moxitown, and Hokkaido data. Non-landslide images sourced from DeepGlobe and Wildfire Prediction data to provide more image variance.	Train/ Validate	3200	0: 1600; 1: 1600	[7]–[9]
Set 3: Identical to Set 2 but expanded by count. Non-landslide images expanded with the addition of the "satellite-image-deep-learning" (Hugging Face) data adding more high-resolution land and cityscape aerial views.	Train/ Validate	6974	0: 3034; 1: 3940	[7]–[10]
Set 4: Combined Landslide dataset merging Set 3 and 168 new images evenly classified from our NASA-based test images.	Train/ Validate	7132	0: 3115; 1: 4017	[3], [7]–[10]

TABLE I: Training / Validation Data Sets

Dataset	Type	Total Size	Class Size	Ref.
Set 5: Independent Test Set – A separately developed test set identified on the NASA known landslide inventory and captured from Google Earth using latitude/longitude coordinates.	Test	93	0: 44; 1: 49	[3]
Set 6: Independent Test Set Enhanced – An enhanced testing set based on Set 5 with additional image count.	Test	168	0: 84; 1: 84	[3]
Set 7: Separate Test Set – Second test set developed from other unique images through Google Images and research papers studying landslide events/ aerial imagery. None of the images duplicate with Set 5 or 6 and are intended to be used when training with Set 4 for Training/Validation.	Test	62	0: 31; 1: 31	[3], [11]

TABLE II: Testing Data Sets

B. Modeling

Based on our outlined experiments in section II/D above, our team assessed several scenarios across model design, transfer learning, model architecture, hyper-parameter tuning, generalization, feature analysis, and various data cases.

Our team's initial architecture began with two general model approaches and minimal fine-tuning of parameters. This approach was agreed upon to a) design a model process to ensure the data to model workflow worked successfully and

b) to start with a simplified approach, in order to obtain a broad understanding of how the initial models would work under various data sets and to inform us of areas where further research and refinement were needed.

1) Baseline CNN: The first model set included a baseline Convolutional Neural Network (CNN) consisting of 5 layers including three Conv_2D and two Dense layers outputting to two class probabilities. Padding across the Conv_2D layers was set to 'same' and activation set to 'relu'. Two versions were created using BinaryCrossentropy and SparseCategoricalCrossentropy as loss functions.

2) CNN Based Pretrained Transfer Learning: The second set relied on transfer learning as a primary design principle, with various architectural elements added. Each of the following models have been pre-trained with ImageNet weights, were kept frozen with exception to the final Dense layer for classification and with limited image augmentation.

Pre-trained Model	Size (MB)	Parameters	Depth
Baseline CNN	10.47	2.8M	5
EfficientNetB3	48	12.3M	210
ResNet50	98	25.6M	107
MobileNet	16	4.3M	55
DenseNet121	33	8.1M	242
NasNetLarge	343	88.9M	533

TABLE III: Key CNN model architectures used, with key features

EfficientNet has several models that are built to offer better performance on high-resolution imagery, which should be consistent with our goals. ResNet50, a common architecture for image classification in use by several other prior research teams for the purpose of landslide classification, was considered as a benchmark. For this project, DenseNet and MobileNet were also used, both of which are known for their ability to perform well in image classification tasks. DenseNet, particularly the DenseNet121 variant, is known for its dense connections between layers, that allows for better feature reuse and efficiency in learning. MobileNet, on the other hand, is designed for efficiency with fewer parameters, making it suitable for real-time applications where computational resources are limited. One attribute of all the models assessed is the variety of parameter count and layer depth explored.

Additionally, further experimentation with data augmentation/ normalization was performed to assess result improvements. Several optimizers were also tested (e.g. RMSprop, Adam, Nadam, etc.), however results are inconclusive as performance was already incredibly high, possibly stemming from data issues further expounded upon below.

3) Class Weights: Because some of the Training/ Validation data sets were not balanced across both classes (landslide and non-landslide), we integrated Class Weights within the model fitting process to handle any class imbalances present.

4) Augmentation: Given our limited ability to obtain an appropriately diverse and large dataset, transfer learning on larger pre-trained models was necessary. We realized after further experimentation with our initial, yet smaller dataset, (e.g., Set 1/Table 1) over-fitting was present from the early model runs. We believe this was due to the very homogeneous nature of the imageset, where there was a high level of similarity of terrain type, landslide size, proximity to urban areas, slide type (e.g., rock slide, debris slide, mudslide), coloration (similar hues), etc. In order to broaden the distribution of images for better generalization, we experimented with Augmentation to expand the image set artificially. We established following augmentation layers through Keras in our model workflow:

- Cropping [350x350 px to 300x300 px] - Provides an opportunity to have each training epoch train on a shifted image, so that the center-point will be in different location which helps our smaller NASA set diversify. This was not a particular issue with Set 3, where the dataset already included duplicate although shifted images of single landslides, making up many different landslide views.
- Saturation - Certain data sets in Set 4 had similar hue/saturation, therefore by allowing for random altering of saturation levels we hoped to expand image diversity.
- Translation, Flip - Similar motive to Cropping.
- Contrast, Brightness - Similar motive to Saturation.

5) Fine-Tuning and Layer Unfreezing: As a further enhancement to Transfer Learning, we explored Fine-Tuning and the ability to unfreeze top-level feature layers in the model to allow for further training of weights while maintaining the pre-trained state of low-level features saved by the EffcientNetB3 model. As noted by Medium.com; "These low-level features remain relevant across tasks, while the higher-level features in later layers may require adaptation for task-specific nuances." [12].

While using EfficientNetB3, there are a total of 210 layers that have been trained on Imagenet data. One notable design feature in Keras is that when assessing the layer structure using model.summary(show_trainable=True), we count a total number of layers at 389, which takes into account each line item in the design table. Our selected unfreeze value in our model pipeline directly opens the number of line items listed in the EfficientNet summary, avoiding layers that do not allow for training (e.g., such as BatchNormalization that must remain frozen).

Following our Phase 1 exploration, we found that ENB3 and ResNet50 held the most promise for our application. Starting with the base ENB3 and ResNet50 models, we first trialed a basic approach where we unfroze the models layer by layer, with fewer epochs to try and limit over-fitting while getting preliminary performance figures. Choosing an arbitrary amount of top layers to unfreeze, (excluding Batch Normalization), we found that ResNet50 would tend to perform better during training, yet under-perform on the prediction evaluation, having an overall lower Accuracy, Precision, and

Recall. We then explored the model architecture to see if there could be a less arbitrary approach to which layers to unfreeze.

In our 2 models, layers tend to be grouped into CNN blocks, separated by various activation, batch normalization, or dropout layers. By unfreezing these 'blocks', we found that we could achieve better performance at lower epochs, particularly with ENB3, with the idea that each block functioned together as a logical cluster. However, once we extended our dataset to include less homogeneous data, this advantage was reduced, and we did not see a marked improvement over our arbitrary layer choices, which suggests that with sufficiently deep models (ENB3 is 210 layers), the difference between unfreezing 1 (22 layers), about 1.5 (30 layers), or 2 blocks (35 layers) versus going layer by layer is relatively minor, with learning rate and epoch adjustment having a greater effect, likely as they more heavily influenced over-fitting.

We selected a number of layers arbitrarily staring with a lower value so that only a few high-level features were opened to further supervised training.

6) Weight Transfer to a Secondary Model: Another modeling approach we explored was use of a two model architecture. This would entail a transfer learned pre-trained EfficientNetB3 model with layer fine-tuning using Train/Validate Set 2 and 3 data with the weights and design then saves as a '.keras' file. The weights and model design would then transition into a new model that would be trained with various levels of fine-tuning using Test Set 5 and 6 (e.g., the more diverse landslide image sets). The hypothesis for this approach was; if we could transition a fine-tune model that was previously exposed to landslide images across various layers, then this weight set could be enhanced with a much smaller and diverse dataset to improve the model for better generality and prediction. After several trials with varying 1st model training fine tuning (e.g., Base, 15 and 40 Layer fine-tuning) and 2nd model fine-tuning, our results with Sets 5 and 6 did not perform well when observing Validation accuracy over the new image sets.

7) Multi-Modal: One of our additional goals was to try to incorporate some form of multi-modal approach to further improve prediction and add the ability to classify landslides. We experimented with implementing a simple multi-modal model which combines output from a ENB3 based visual model alongside a simple embeddings model to gather data from the landslide size labels. Initial results are poor, however this is likely because we need much further development beyond the scope of this project.

8) EfficientNetB3 Architecture: The architecture illustrated in Figure 3 showcases the final deep learning pipeline for landslide image classification using the EfficientNetB3 model. The process begins with input images (300x300 pixels) undergoing preprocessing and initial convolutional layers, which extract low-level features like edges and textures. These features are further processed through the EfficientNetB3 blocks, consisting of MBConv (Mobile Inverted Bottleneck Convolution) layers with increasing channel dimensions and spatial downsampling. This enables the model to progressively learn hierarchical representations, capturing complex patterns and details relevant to landslide detection. The final stage of the model includes global average pooling, dropout for regularization, and a fully connected layer that outputs class probabilities (landslide or non-landslide). Additionally, the Grad-CAM visualization path overlays heatmaps onto the original images, highlighting regions most influential in the model's decision-making process. This interpretability mechanism provides valuable insights into the model's focus areas during classification.

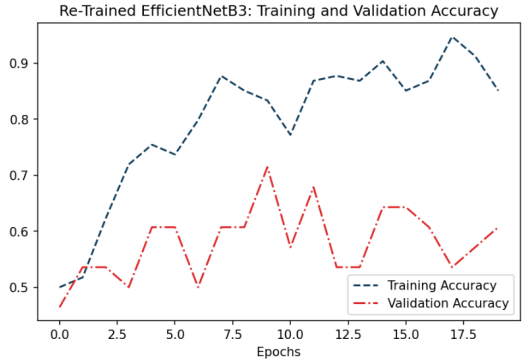


Fig. 2: ReTrained EfficientNetB3 Model - Training 40 Layers from Prior 40 Layer Fine-Tuned .Keras EfficientNetB3 Model

9) Deeper EfficientNet Methodologies: We also experimented with using deeper versions of our models, such as ENB4 (258 vs 210), though we saw equivalent or worse prediction performance compared with our best ENB3 models, with far longer training times, again likely explained by over-fitting.

V. Results

A. Phase I

Results from Phase I model execution utilizing data from Set 2 training/validation and Set 5 testing were as follows:

Model	Training Accuracy	Validation Accuracy	Test Accuracy	Test Precision	Test Recall
Baseline CNN	.8898	.8953	.5053	.5800	.7230
EfficientNetB3	.9905	.9969	.6340	.6140	.5380
ResNet50	1.000	.9984	.5252	.6538	.3863
MobileNet	.9688	.9589	.5269	N/A	N/A
DenseNet121	.9062	.97	.5526	N/A	N/A
NasNetLarge	.6903	.7437	.6270	.7310	.2920

TABLE IV: Landslide CNN Execution and Performance Across Various Models (over 10 Epochs)

From this initial execution, we found that EfficientNetB3 performed well over Training/Validation and most aspects of Testing, therefore we chose to focus our future experimentation with this model family during our Phase II.

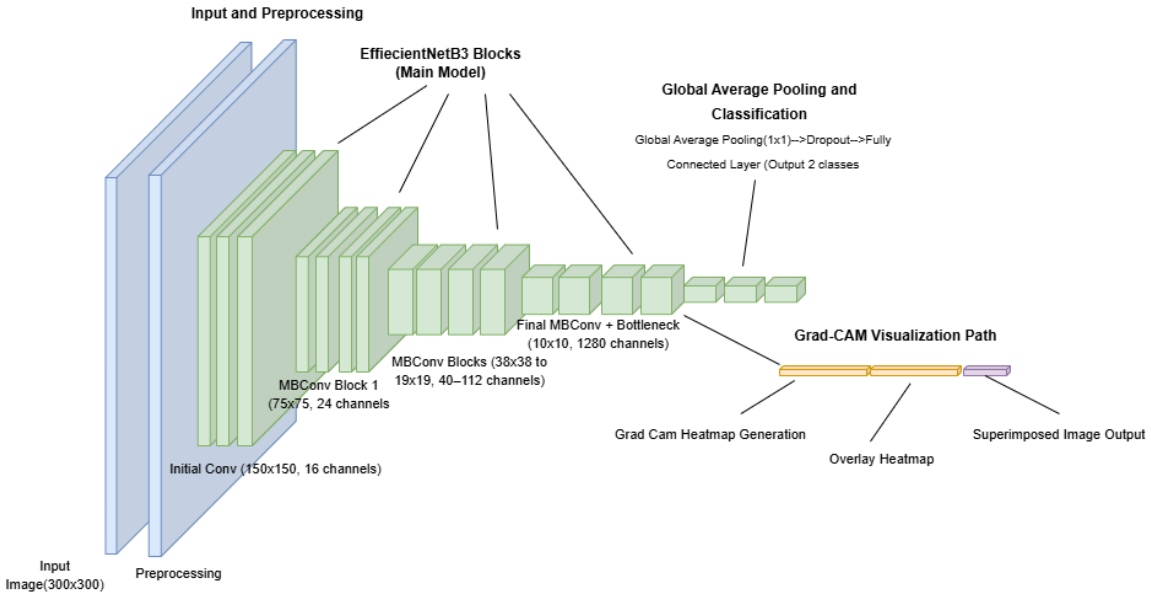


Fig. 3: Architecture of the EfficientNetB3 Model

B. Phase II

Once we identified a model family that appeared to perform well in both landslide identification and overall accuracy, we set out to perform the above experimentation with model design and parameter modification. Our primary objective was to find a design that would primarily optimize Test Recall, or the ability for the model to identify all landslide events in any test population. Precision; the accuracy of Landslide predictions, Specificity; the accuracy of Non-Landslide prediction and overall Accuracy were also monitored. Results for several scenarios are detailed as follows:

1) Impact of Dataset Diversity and Size: A diverse dataset that incorporates satellite imagery from multiple geographical regions, terrains, and environmental conditions significantly improved the model's ability to generalize and identify landslides across different settings. Imagery from different sources helps the model avoid focus on latent factors, as various sources offer differing levels of fidelity, zoom, and other image qualities. By utilizing data augmentation techniques such as cropping, saturation adjustments, and contrast variations, the dataset was further expanded to include a wider range of image variations, helping the model to become more robust and adaptable to different conditions. Additionally, the inclusion of a larger dataset allowed the model to learn from rare or unique landslide features, which is essential for ensuring accurate predictions in real-world scenarios, where landslides may present differently than in the training data. As the dataset improved in size/diversity from Set 1 to Set 4 (see Fig. 4 and 5), we observed improvement in the model's ability to predict the landslide class, as well as overall accuracy. We also were able to clearly identify the model over-fitting under Set 1 where training/validation performance was relatively high, but inference results were fairly low. With that stated, challenges remain in acquiring large amounts of labeled data for this specific problem case, especially for less common landslide types, highlighting the need for further dataset expansion to improve model accuracy and generalization.

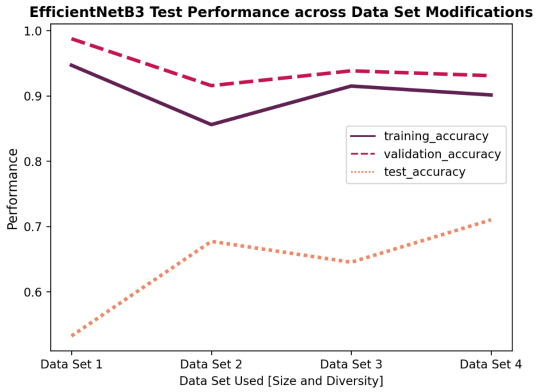


Fig. 4: EfficientNetB3 Training/Validation/Test Performance as Training Data Improves from Set 1 to Set 4

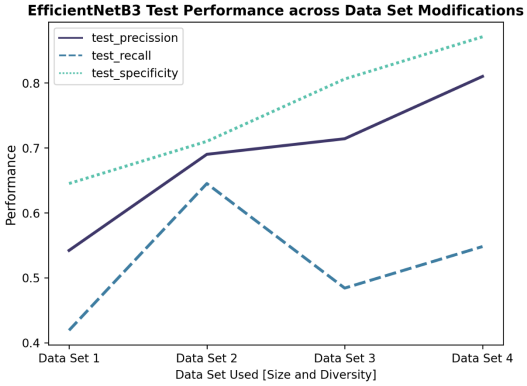


Fig. 5: EfficientNetB3 Test Performance as Training Data Improves from Set 1 to Set 4

2) Fine-Tuning of EfficientNetB3: We experimented with varying the amount of unfrozen higher-level features in EfficientNetB3 training over 15 epochs. We found that further training epochs provide a positive enhancement in training/validation and test performance, although with diminishing returns. But as we increase the levels of unfreezing, overall accuracy and especially recall, or the ability of the model to truly identity landslides, appears to level out.

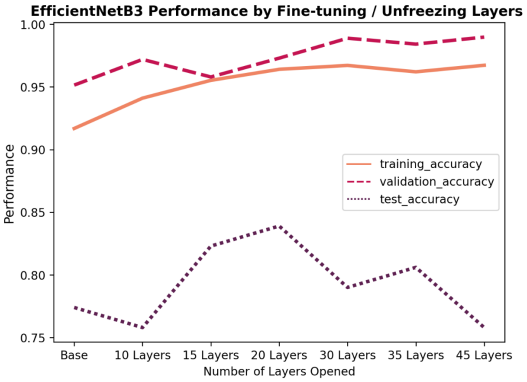


Fig. 6: Training, Validation and Test Accuracy over Layer Opening Scenarios during Model Fine Tuning

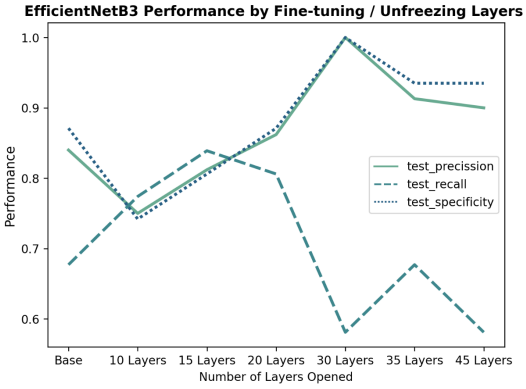


Fig. 7: Test Performance relating to Landslide Class Identification across Layer Opening Scenarios

3) Top Performing EfficientNetB3: Upon evaluating execution results across the experimentation scenarios as described above, we identified a model range where improved inference results were observed for a) identifying landslide out of the test population and b) also providing very good classification of those images where landslides were absent. Our focus was to highlight those models where Recall (e.g., instances where actual landslide images were identified) was high, along with overall good accuracy and specificity.

EfficientNetB3, when Fine-Tuned with Layer unfreezing around 10 to 20 layers, provided optimistic results related to identification of landslides in images, as well as identifying lack of landside activity. We highlight these results. For this model group, we implemented augmentation, dropout, use of the Adam optimizer with a learning rate set at .001 and training over 15 epochs.

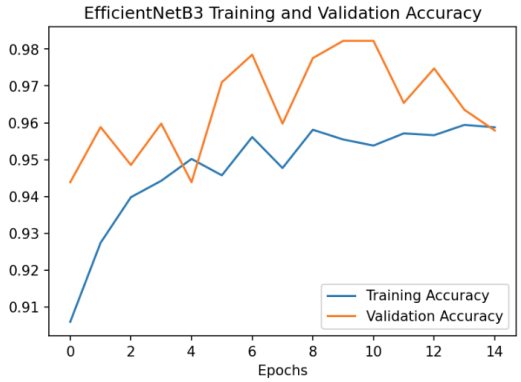


Fig. 8: Training & Validation Accuracy for EfficientNetB3 (w/ 15 Layer Fine-Tuning)

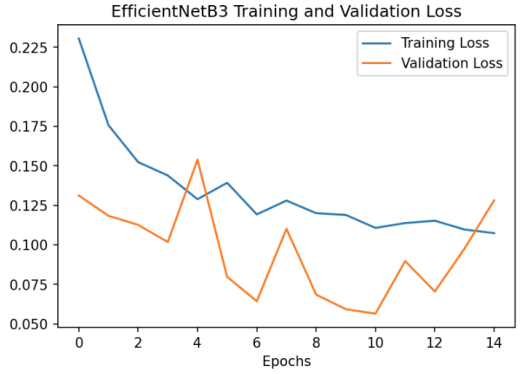


Fig. 9: Loss Results from Training/Validation of EfficientNetB3 (w/ 15 Layer Fine-Tuning)

4) Grad-Cam Analysis: The Grad-CAM visualizations illustrate how the trained model identifies regions of interest in the input satellite imagery to make predictions. These visualizations provide insight into the areas of the image that the model considers most influential for its classification decisions. The first visualization displays the raw activation map generated by Grad-CAM for a specific convolutional layer.

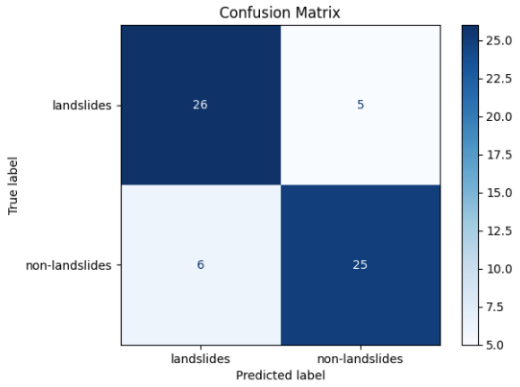
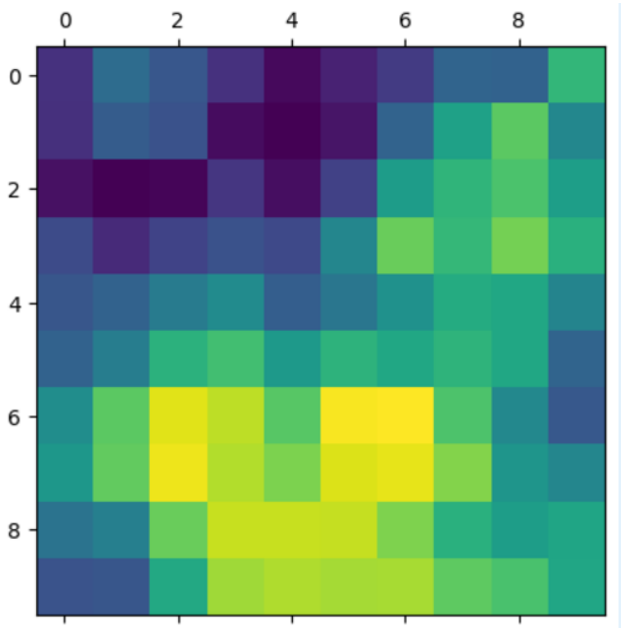


Fig. 10: Confusion Matrix - EfficientNetB3 (w/ 15 Layer Fine-Tuning)

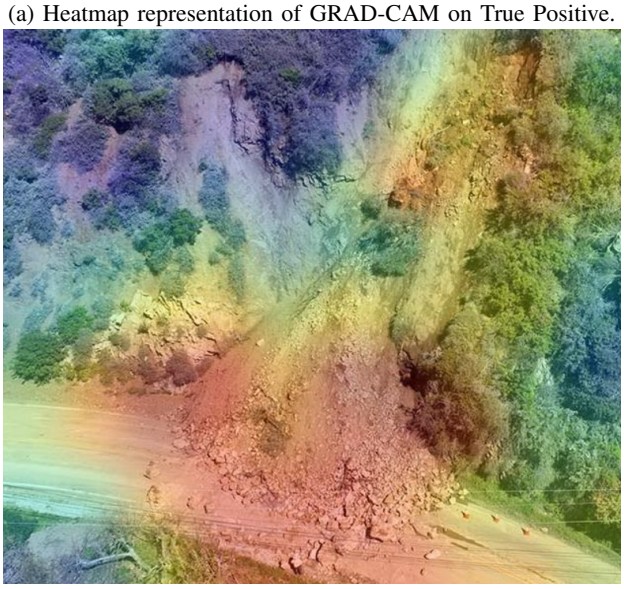
Metric	Value
Training Accuracy	.9552
Validation Accuracy	.9579
Training Loss	.1186
Validation Loss	.1281
Test Accuracy	.8230
Test Precision	.8120
Test Recall	.8390
Test Sensitivity	.8060
Test F1 Score	.8250

TABLE V: Key Metrics for EfficientNetB3 Test Performance (15 Layer Fine-Tuning)

This heatmap highlights regions where the gradient values are strongest, indicating features that significantly contribute to the model's predictions (Figure 11a). The lower resolution of this map is due to the spatial downsampling that occurs within convolutional layers through pooling and stride operations. The second visualization enhances interpretability by upscaling the activation map to the dimensions of the original input image and overlaying it onto the satellite imagery (Figure 11b). This composite image shows the spatial relationship between the model's focus and the actual geographical regions, with brighter areas (e.g., yellow and green) corresponding to regions that are critical for the model's decision-making. Conversely, darker regions indicate areas with little influence on the output. For example, in this analysis, bright regions may correlate with geological features or terrain indicative of landslides, while darker regions may represent areas with less relevance, such as water bodies or flat terrain. These visualizations confirm that the model is learning spatially meaningful patterns, with its attention aligning to regions expected to have high predictive value for the task. This level of interpretability is crucial for validating the model's behavior, ensuring its focus aligns with domain knowledge, and identifying potential biases or errors. Further evaluation against ground truth data and domain-specific expectations is necessary to fully assess the reliability and utility of the model's focus. Nevertheless, these Grad-CAM outputs demonstrate the potential of the model to detect relevant features, providing both quantitative predictions and qualitative evidence to support its decisions.



(a) Heatmap representation of GRAD-CAM on True Positive.



(b) Example True Positive from Fine-Tuned EfficientNetB3.

Fig. 11

An apparent pattern in the false positives and false negatives was a tendency for the network to focus on some of the corners of the images, particularly the top right corner (Figure 12). Further work is needed to understand when and why this occurs.

VI. Conclusion

This study demonstrates the effectiveness of deep learning models, particularly EfficientNetB3, for landslide detection using satellite imagery. By utilizing transfer learning and fine-tuning techniques, we were able to significantly improve the model's performance in both identifying landslides and

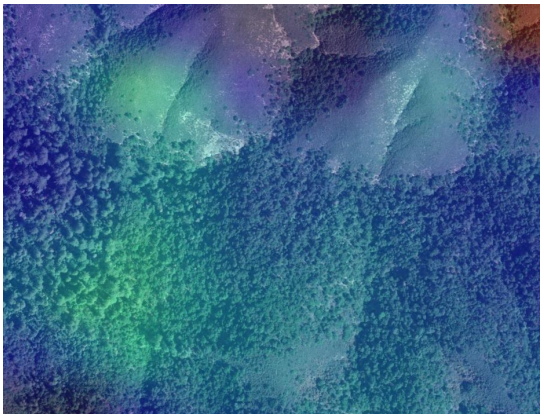


Fig. 12: Example False Positive from Fine-Tuned EfficientNetB3.

distinguishing non-landslide areas. Among the models tested, EfficientNetB3 stood out due to its superior ability to generalize across different datasets and handle complex image features, achieving high accuracy, precision, and recall.

A. Implications

The practical applications of this study are far-reaching, especially in the context of disaster prediction and management. Early and accurate landslide detection can play a crucial role in reducing risks to life and infrastructure by enabling timely warnings for affected areas. By automating the analysis of satellite imagery, this approach can provide real-time monitoring of landslide-prone regions, allowing authorities to issue alerts before disasters strike, thereby minimizing the impact on communities.

Furthermore, the use of Grad-CAM visualizations enhances model interpretability, making it easier to understand why certain regions of an image are classified as landslides. This interpretability is valuable not only for validating the model's predictions but also for providing insights into the environmental factors contributing to landslides, which can be crucial for policymakers, urban planners, and disaster response teams.

B. Limitations

Despite the promising results, some limitations remain in this study. The model's performance is heavily influenced by the diversity and quality of the training data. The inclusion of additional datasets from varied geographical regions and conditions could further enhance its robustness. Furthermore, the need for large, labeled datasets of satellite images for training poses a significant challenge. In some instances, the model struggled with differentiating between landslides and similar-looking terrain features in certain images, highlighting the need for more nuanced data to train the model.

Another limitation is the computational resources required for training and deploying these models. The processing of large satellite images and the high computational cost of training deep learning models like EfficientNetB3 can be prohibitive, especially when scaling the model for real-time use across large regions.

C. Future Development

Future work can build upon the strengths of this study by incorporating additional data sources, such as weather patterns, soil moisture, and seismic activity, which are key factors in landslide occurrences. By integrating these variables, the model could gain a deeper understanding of the temporal and environmental dynamics that influence landslides, leading to more accurate predictions.

Expanding the model to account for temporal changes is another avenue for future development. Landslides often evolve over time, and incorporating temporal data could improve the model's ability to predict imminent landslide risks. Additionally, exploring advanced techniques for model optimization and transfer learning could further enhance the model's efficiency and scalability, making it feasible for large-scale deployment.

Furthermore, the model's interpretability could be improved by refining the Grad-CAM visualization process to highlight not only the areas of interest but also the specific geological or environmental features associated with landslides. This would provide deeper insights into the predictive patterns of the model, benefiting both model validation and real-world decision-making processes.

By addressing these limitations and expanding the model's capabilities, this research has the potential to become a powerful tool for landslide detection, contributing significantly to disaster prevention and environmental risk management efforts.

VII. Members Contributions

A. Github Reference

https://github.com/eltsvetk/DS6050_Project

B. Contributions

- Harold Haugen – Assessment of the NASA landslide .csv data and build of the Google Earth test set, as well as the final test set from NASA Earth Observatory and other Google Images (see Landslide_SatImage_Tracker.ipynb). Development of the Training/ Validation data and workflow from Dropbox to the team's Juptyer notebooks (see Image_Data_Loading.ipynb notebook). Development of the Baseline CNN model, NASNETLarge, and experimentation across several scenarios using transfer learning with differing layers and different data cases on Efficeint-NetB3. Developed the two model approach for Weight Transfer from a saved .Keras EfficientNetB3 to a new EfficientNetB3 model and capturing results. Contributed to the data, data size and layer size experimentation, methods/modeling and literary review sections fo the report. Helped with writing up the GitHub ReadMe and final presentation of results to the class.
- Daniel Sery – Tested ResNet-50, optimizers, and experimented with image normalization, grayscale, as well as

minor additional experimentation such as dataset size effect on performance in Phase 1. Further development and comparison of ResNet and ENB3 as our lead candidate models. Experimentation with weight transfers. Experimentation with unfreezing approaches based on model architecture (block vs layer). Exploration of deeper models. Development of multi-modal. Revision of report.

- Max Pearton - Creation of the Densenet121 and Mobilenet transfer learning experiment models in Phase 1. Exploration of different datasets on those two models. Development of the EfficientNetB3 model architecture visualization using diagram.net software. Contributed the building of Grad-CAM architecture for EfficientNetB3 in Phase 2. Contributed to the conclusion, abstract, overview, limitations, future development, Grad-CAM analysis, impact of dataset diversity and size, EfficientNetB3 architecture of the final report and helped with final presentation slides.
- Elena Tsvetkova- EDA analysis showing that the larger landslides occur outside of the US. Tested different parameters and convolutional layers to the simple CNN model. Added data augmentation and some hyperparameter testing to Densenet model. Creation of Grad-CAM to the final EFB3 model, before and after tuning. Used confusion matrix results from the test set to understand where the network focused its attention when predicting correctly and incorrectly. Converted the lambda layer in data augmentation code to a subclass of keras layers to allow Grad CAM to work. Contributed to the Intended Experiments and Results section of final report. Helped with final presentation and cleaning up Github repo and adding to ReadME.

REFERENCES

[1] Parsimonious High-Resolution Landslide Susceptibility Modeling at Continental Scales; https://doi.org/10.1029/2024AV001214

[2] USGS Lanslide CSV: Slope-Relief Threshold Landslide Susceptibility Models for the United States and Puerto Rico; https://www.sciencebase.gov/catalog/item/65ccea5bd34ef4b119cb3bac

[3] NASA Cooperative Open Online Landslide Repository (COOLR) Catalog; https://gpm.nasa.gov/landslides/projects.html

[4] Satellite image classification with a convolutional neural network; https:www.sciencedirect.com/science/article/pii/S1877050924006471

[5] A new strategy to map landslides with a generalized convolutional neural network; https://www.nature.com/articles/s41598-021-89015-8

[6] HR-GLDD: A globally distributed high resolution landslide dataset; https://zenodo.org/records/7189381

[7] CAS Landslide Dataset: A Large-Scale and Multisensor Dataset for Deep Learning-Based Landslide Detection; https://zenodo.org/records/10294997

[8] DeepGlobe Land Cover Classification Dataset; https://www.kaggle.com/datasets/balraj98/deepglobe-land-cover-classification-dataset

[9] Wildfire Prediction Dataset (Satellite Images); https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset

[10] Hugging Face SODA-A Satellite Images; https://https://huggingface.co/datasets/satellite-image-deep-learning/SODA-A

[11] NASA Earth Observatory; https://earthobservatory.nasa.gov/topic/landslides

[12] Medium Basic Understanding of Fine-Tuning in Deep Learning, VDOIT Technologies Limited; https://medium.com/@vdoitseo/ /basic-understanding-of-fine-tuning-in-deep-learning-f0f0d5598b77