

Predicting the Flood Risk of a Region Using Convolutional Neural Networks and Satellite Images

Vedansh Gupta

Received June 29, 2024

Accepted November 01, 2024

Electronic access November 15, 2024

Floods are one of the most dangerous natural phenomena and cause huge societal and economic losses. Global warming has increased the intensity and frequency of flood events. The flood risk of a region must be identified before a flood occurs, which would allow more time and resources to be spent on transporting materials to assist high flood-risk regions. The prevalence of machine learning (ML) has increased in recent years, presenting a promising solution for identifying the flood risk of a particular region easily and quickly. A Convolutional Neural Network (CNN) can be trained on satellite images of regions of known high and low flood risk worldwide to predict the flood risk of an area based on its satellite image. Feature extraction can also be performed on the neural network to identify the important features of a satellite image for predicting flood risk. This could assist experts in making visual flood predictions. Therefore, this research aims to determine whether a CNN can identify factors that cause some areas to be flood-prone and predict the flood risk of a region using satellite imagery. Other research on this subject is localized to determine flood risk in specific regions, but this research's datasets are global. A CNN was trained on satellite images from regions across the globe. The number of satellite images and regions in the training set was varied. Feature extraction was also done on the model. The highest classification accuracy of 93.22% occurred with the training set of 246 images and 38 locations.

Keywords: Convolutional Neural Network, flood-risk prediction, satellite imagery

Introduction

From 1995 to 2015, floods impacted 2.6 billion people¹. Around 89% of these individuals reside in countries with lower to moderate income levels, where floods frequently trigger food insecurity issues^{2,3}. Research indicates a significant increase in the number of people affected by floods, with 58 to 96 million more people being affected by floods between 2000 and 2015, alongside an increase in both the frequency and severity of flood events⁴. Not only are floods dangerous for people, but they can also have a severe impact on the economy. For example, in Pakistan, heavy monsoon rain affected 33 million people and damaged farmland, causing food insecurity⁵. The flood caused 14.9 billion USD in damage and \$12.5 billion in economic losses, highlighting the devastating impact that floods can have⁵. Therefore, it is imperative that solutions for excessive and damaging flooding be found. One such solution comes through using machine learning (ML) to assist in flood prevention efforts⁶. This research aims to determine whether a Convolutional Neural Network (CNN) can identify the factors that cause some areas to be flood-prone and predict the flood risk of a region using satellite imagery.

If a region is determined to be flood-prone by the CNN model, assistance can be given to help residents during a flood before the next one occurs. In addition, more permanent changes

can be made to manage or reduce the flood risk of the region. These changes include routine monitoring, flood forecasting, improving the flood resistance of infrastructure, and retreating from hazardous areas⁷.

Floods are caused by a sharp increase of water in a region, leading to more water present than the ground can absorb. Heavy rainfall, melting snow, temperature, wind direction, duration, and frequencies of storms are all potential causes of floods⁸.

However, this research will focus on the geographic features that can cause a flood, such as whether the region is covered in rock, vegetation, or snow, the conditions of the soil, and the presence of water nearby⁸. These geographic features can affect the ground's ability to retain water. For example, the ground's ability to retain water is enhanced when there is much soil in the region⁹. Plant roots help maintain a porous soil structure and help bind soil particles together, preventing erosion during rainfall⁹. Thus, regions with much vegetation may have a reduced risk of severe flooding.

Satellite images are a good way to classify regions that are flood-prone and not flood-prone since they provide data even on remote regions in poorer countries, where other data such as governmental land surveys or scientific studies may not be available¹⁰. Many datasets of satellite imagery are readily available online, so the cost of using satellite imagery for flood prediction is minimal and less time-consuming¹¹. In addition, satellite

images provide a good overview of the geographic features in a particular region, which as stated before are crucial for determining flood risk. Machine learning using this satellite imagery can be a powerful way to determine the risk of a region for flooding. This reduces the time and resources for determining the risk level of an area and instead allows for more resources to be diverted toward helping flood-prone regions with flooding issues¹².

A neural network is made up of many interconnected neurons that receive input and output to and from each other with the goal of pattern recognition¹³. A Convolutional Neural Network (CNN) is a type of neural network used mostly for image recognition and it breaks down an image into its individual pixels. Each 3 by 3 group of pixels in the image is looked at to determine how well it matches a specific filter. This data is stored in a neuron, as a number from 0 to 1, with one being closer to how well the group of pixels matches the filter. These filters are indicative of certain elements within the image¹⁴. CNNs have multiple layers, and in each layer the filters it checks for become more abstract. For example, in the first layer, the CNN might check for a filter signifying a specific edge. Then in the next layer, the CNN might check if that edge matches the edge of a coastline or a river, which indicates evidence to determine if the region is flood-prone. In addition, feature extraction can be performed to see which filters the CNN uses to classify regions as flood-prone or not flood-prone. These filters are created by the CNN during the training process, so they are the most important features of a satellite image for classifying it as flood-prone or not flood-prone.

Therefore, this research aims to address the question: Can a Convolutional Neural Network identify the factors that can cause some areas to be flood-prone and other areas not to be flood-prone using satellite imagery? Furthermore, can it predict the flood risk of a new area based on its satellite image?

Literature Review

Previous studies have shown that satellite images and machine learning can be used to demonstrate the extent of flooding in certain regions^{12,15,16}. One study by Tanim et al. combined satellite imagery with road closure reports to predict the extent of flooding in urban areas at various times¹². This study utilized Random Forest (RF), Support Vector Machine (SVM), and the Maximum Likelihood Classifier (MLC) machine learning models to make the predictions. It also created a new unsupervised machine learning framework based on the change detection (CD) approach. The accuracy of the models in classifying water and non-water pixels is 0.69, 0.87, 0.83, and 0.87 for the RF, SVM, MLC, and CD models respectively. The SVM and CD methods had the highest accuracy overall. In addition, the study found that flood pixels were mixed up with pixels from the pavement in many instances. This and similar mixup issues are one that

many models to predict flood risk can face, and one which the model in this study faced.

The second study by Portalés-Julià et al. used satellite imagery to determine the extent of flooding for major recent flood events in Pakistan and Australia from Sentinel-2 and Landsat-8/9 optical imagery data¹⁵. The model in the study demonstrates an end-to-end flood mapping system to produce flood extent maps. The data pipeline uses a new cloud-aware flood segmentation model that produces independent cloud and water masks, meaning it works on images covered by semitransparent clouds. Satellite images sometimes have clouds that block the ground, which can be a major problem for research attempting to use satellite images. A machine-learning model that can work around this issue will be useful in increasing the number of usable satellite images in a training set. Both of these studies are important because they showcase using satellite data for flood mapping, which is similar to the research in this paper. However, neither of these studies was about predicting the flood risk of a region using machine learning or CNNs.

A third study by Panahi et al. used the two deep learning networks for a spatially explicit prediction of flash flood probability for the Golestan Province in Iran¹⁶. The two networks used were convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to investigate the relationship between floods and the factors that cause them. Just like for the research in this paper, Google Earth Engine satellite images were part of the training set. The study found that the models can predict flood probability well in the varied terrain and climate of the Golestan Province, and the resulting probability maps can assist in creating mitigation plans for future floods. It also found that the CNN performed marginally better than the RNN, with a 0.78 accuracy to a 0.77 accuracy, but this is not a significant difference. The study found that regions with low elevation, gentle slopes or flat areas, near rivers, and extensive human activity were at higher risk for flooding.

All three of these studies were localized to specific regions. However, the research done in this study builds on the research from the other studies by including datasets from regions all over the globe. This is done to create a more general model that can predict flood risk anywhere.

Methodology

A CNN was created and trained on satellite images from the Google Earth database. The code from a tutorial was modified to create the CNN¹⁷. The CNN was made using the Keras API and the TensorFlow software library. There are nine layers used in total - the order in which they are used is shown in Table 1. Table 1 also shows the output shape of each layer as well as the number of trainable parameters within each layer. For output shape, the first value, "None," refers to the batch size, which is the number of samples the model will look over before

updating its parameters. It is set to “None” here because the batch size was used later in the code when the model was run (but the actual batch size was the standard 32). The next two numbers refer to the size of the output shape in pixels. For example, the first layer outputs a feature map of size 256 by 256 pixels. The last number refers to the number of filters produced by each layer. The number of trainable parameters metric refers to the number of parameters the model needs to adjust during the training process for that specific layer.

Layer (type)	Output Shape	Number of Trainable Parameters
Conv2D	(None, 256, 256, 16)	448
MaxPooling2D	(None, 128, 128, 16)	0
Conv2D	(None, 126, 126, 32)	4,640
MaxPooling2D	(None, 63, 63, 32)	0
Conv2D	(None, 61, 61, 16)	4,624
MaxPooling2D	(None, 30, 30, 16)	0
Flatten	(None, 14400)	0
Dense	(None, 256)	3,686,656
Dense	(None, 1)	257

Table 1 Layers used in order in the training of the CNN model as well as their output shape and number of trainable parameters

Table 2 highlights all the key hyper-parameters used in training the model. The filter’s size is 3 pixels by 3 pixels. The model trains over 20 epochs, after which point the validation accuracy ceases the increase, meaning the model has reached peak accuracy. The Adam optimizer was used, which has a learning rate of 0.001. The stride is 1. Valid padding was used to reduce the number of parameters in the model and improve efficiency. The ReLU activation function was used for every layer but the last Dense layer, where a sigmoid activation was used. The Binary Cross-Entropy loss function was used since the CNN model performs binary classification. Additionally, within the Keras API, a sequential model was used since each layer had one input and output tensor.

The CNN has the ability to perform binary classification, meaning it can classify images into two distinct categories. This research has the assumption that every region in the world is either flood-prone or not flood-prone. Although this is not the case, it reduces some complexity by focusing on the extreme cases (either the region is flood-prone or not). Binary classifica-

Parameter	Value
Number of Epochs	20
Optimizer	Adam
Filter Size	3 pixels by 3 pixels
Stride	1
Padding	Valid padding
Activation Functions	ReLU, Sigmoid
Loss Function	Binary Cross-Entropy

Table 2 List of key hyper-parameters used

tion was also used since it is the simplest to implement in code. In this case, the two categories for classification are flood-prone and not flood-prone regions. For flood-prone regions, satellite images from Brazil, Bangladesh, Madagascar, the Philippines, the USA, Australia, Italy, Haiti, India, China, Canada, South Africa, Colombia, and the UK were chosen. For non-flood-prone regions, satellite images from Egypt, the USA, Russia, Niger, South Africa, Saudi Arabia, Peru, Mongolia, Australia, Andorra, San Marino, Canada, Khazakstan, and India were chosen. These locations were chosen based on the World Floods Database, which shows where flooding has occurred in the past 15 years¹⁸. Using this dataset, flood-prone locations were chosen if they had frequent and recent flood events. Non-flood-prone locations were chosen if they had not had a flood event in the past 20 years. Also, there are places from every continent, so there is diversity in the dataset, which will help the CNN model make more accurate predictions. Furthermore, there is a mix of more developed and less developed countries in the list, further adding to the diversity within the dataset. Some locations are near the ocean, some are in the mountains, some are in deserts, others are part of rainforests, there are cities and rural areas, etc.

To obtain this data, sample code was used from a GitHub repository that obtained satellite images from Google Earth Engine¹⁹. The images are JPG files of size 1024 pixels by 1024 pixels. The code was revised to get a small number of images from many different locations, specified in the code with their longitude and latitude, at once. The data pipeline was run and the unusable images (e.g. they were covered by clouds or images weren’t complete) were filtered out. Automatic pre-processing methods reduced the image quality or removed more information than it should have from the image, which is why they were not used in this research. Time constraints also prevented experimentation with other potentially more favorable pre-processing methods. This is also the reason why all available satellite images were not used. Each image had to be manually checked to ensure it was usable, so it would be unfeasible to try and include every image. Furthermore, it takes time and computational resources to train the CNN. Although it would increase the performance of the model, the author did not have the computational resources or unlimited time to include all

satellite images in the training set.

The usable satellite images were stored in a folder. At the end of the initial run-through of the data pipeline, there were 246 images from 30 locations. 125 images were from flood-prone regions and 121 images were from non-flood-prone regions, a roughly 50-50 split. This formed the “base case,” which was used as the standard for comparison. There is no technical reason for starting with 246 images, but rather time constraints hindered the addition of more images to the initial dataset.

The original dataset was modified by adding and removing the number of images or adding or removing the number of locations from this original dataset. For the first modification, 20 images (10 flood-prone and 10 not flood-prone) were added or removed each time, for a total of 3 times, keeping the number of locations (30), constant each time. Overall, there were datasets with 186, 206, 226, 246, 266, 286, and 306 images. For the second modification, 4 locations (2 flood-prone and 2 not flood-prone) were added or removed twice, while keeping the total number of images (246) constant. In the end, images from France, the Netherlands, Japan, New Zealand, Somalia, Argentina, the Democratic Republic of the Congo, and Nepal were added. Overall, there were datasets with 22 locations, 26 locations 30 locations, 34 locations, and 38 locations. So overall, the data looked like the following:

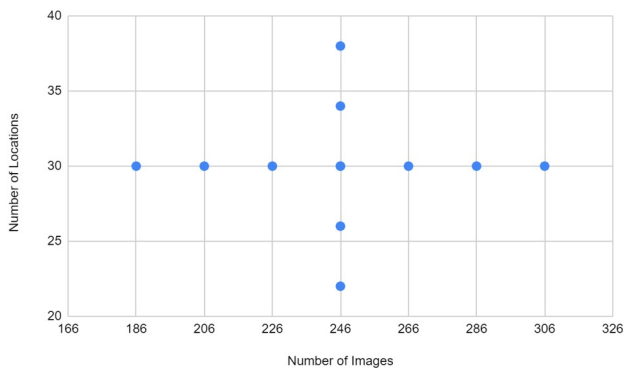


Fig. 1 Each dot represents one data set used for training and validation

All this data had been labeled into two distinct categories (flood-prone region and non-flood-prone region). In addition, 70% of the dataset was used to train the CNN forming the ‘training set’, while the remaining 30% was used to validate the CNN’s performance after it had been trained, forming the ‘validation set’. These 70 to 30 ratios are convention. The data set was shuffled so which images were part of the training and validation sets were randomly determined. Since there were multiple distinct satellite images per region, images from the same region appeared in both the training and validation sets. This ensures that all regions are represented in both the training and validation sets.

The labeled and shuffled data was passed through the CNN,

which analyzed the data to find patterns among flood-prone regions and not flood-prone regions. The CNN used these patterns to predict whether images from a separate testing set are flood-prone or not. The test data set was composed of 59 images and 8 locations that were not featured in the training set. The CNN’s accuracy is measured by calculating the number of correct predictions over the total number of predictions the CNN made. The reason that an additional testing data set was used to determine accuracy rather than simply using the validation set is that the regions in the validation set are featured in the training set. Therefore, the CNN is familiar with the images in the validation set which may increase its validation accuracy. The validation set exists to determine whether the CNN overfitted its model. More explanation on overfitting will be given in the discussion section.

Finally, feature extraction was done to find the patterns that the CNN used to evaluate the data.

This was done via a feature map. A feature map visualizes the weights the CNN has assigned for an input image. This highlights the key features of an image that the CNN looks for in classification. Feature extraction was performed on 3 high and low-flood-risk locations, which were chosen randomly. The high flood-risk locations were Maraetai, New Zealand, Rajkanika, India, and Wuxue, China. The low flood-risk locations were Jaipur, India, Bhotekoshi, Nepal, and San Marino.

Results

Images	Locations	Accuracy
306	30	0.810
286	30	0.741
266	30	0.776
246	30	0.776
226	30	0.759
206	30	0.724
186	30	0.741
246	38	0.932
246	34	0.759
246	26	0.690
246	22	0.759

Table 3 Accuracy of the CNN model on the testing set based on the number of images and locations in the training set

Table 3 shows the results of the analysis. It includes the number of images and locations in the training set as well as the accuracy of that training set. The numbers in the Images column refer to the total number of images in the overall dataset. They include multiple images from the same region as well. For the “base case,” each region had between 6 and 10 satellite images. The number of images per region was increased or

decreased slightly when the total number of images or locations in the dataset was changed. The green highlighted numbers represent the “base case,” as noted earlier, with 246 images and 30 locations. The accuracies may be difficult to interpret from the table, so they are shown in graph form in Figures 2 and 3. In this case, ‘accuracy’ represents how well the CNN classified images from the testing set mentioned earlier. The mean accuracy from the table is 0.742. The median accuracy from the table is 0.759, which corresponds to the dataset of 226 images and 30 locations. The range of the dataset is 0.243. Using the interquartile range method, it can be determined that the accuracy of 0.932 is an outlier. This accuracy occurs in the dataset with 246 images and 38 locations. An explanation for this outlier will be given in the discussion section.

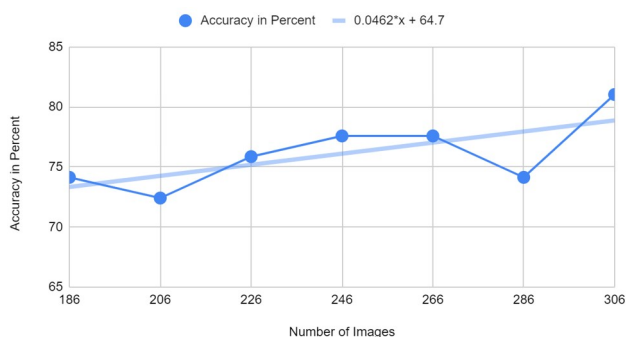


Fig. 2 Accuracy of the CNN model on the testing set in percent vs number of images in the training data set (30 locations each time)

Figure 2 shows the accuracy of the CNN model vs the number of images in the training data set. There are 6 data sets ranging from 186 images to 306 images in total. Each dataset has 30 different locations, however, each image within the data set is unique. As can be seen, there is a positive correlation between the number of images in the data set and the accuracy of the CNN model. The positive slope of the trend line indicates that for every additional image added to the dataset, the accuracy of the model increases by 0.0462%. The y-intercept of the trend line indicates that when 0 images are in the dataset, the accuracy of the model is 64.7%. The accuracy of the model may not have consistently gone up as the number of images in the training set was increased, but overall the trend shows that accuracy increases as the number of images in the training set increases.

Figure 3 shows the accuracy of the CNN model vs the number of locations within the training set. There are 246 images in total for each training set. As more locations were added, there were fewer images per location and vice versa to keep the total number of images constant. As can be seen, there is a positive correlation between the number of locations and the accuracy of the CNN model. The positive slope of the trend line indicates that for every additional image added to the dataset, the accuracy of the model increases by 1.04%. The y-intercept of the trend line indicates that when 0 locations are in the dataset, the accuracy

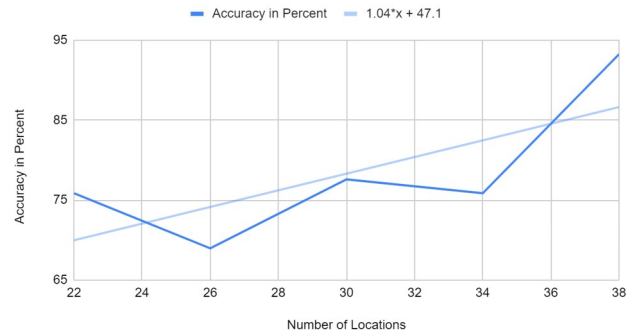


Fig. 3 Accuracy of the CNN model on the testing set in percent vs number of locations in the training data set (246 images each time)

of the model is 47.1%. The accuracy of the model may not have consistently gone up as the number of locations in the training set was increased, but overall the trend shows that accuracy increases as the number of locations increases.

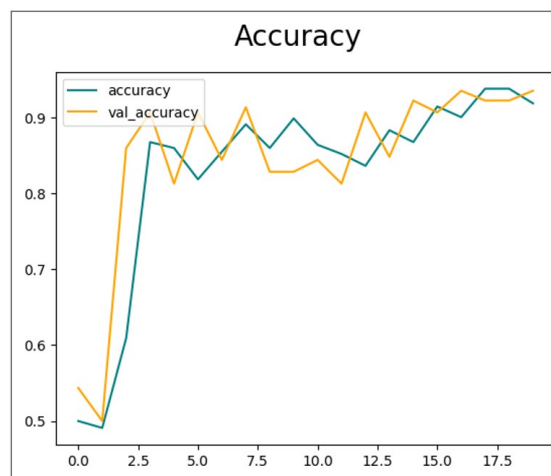


Fig. 4 Accuracy of the CNN Model vs number of epochs for 206 images and 30 locations

As seen in Figure 4, the CNN model trains over 20 epochs. For each epoch, the CNN looks at a batch of images in the training set and recalibrates its weights. At first, the CNN can only make a random 50-50 guess as to whether the region is flood-prone or not. Hence, the accuracy of the model starts at around 50%. The accuracy increases until it stabilizes at a consistent value. When it reaches this value, the model has learned everything it needs to learn from its training set. In this case, the accuracy sharply increases in the first 3 epochs, then stabilizes, and finally increases again right before the end. This indicates that the model initially learns quickly and then it takes several rounds of looking at the training data to fine tune its performance.

Additionally, Figure 4 shows accuracy and validation accuracy vs epoch for the CNN while it is being trained. This graph

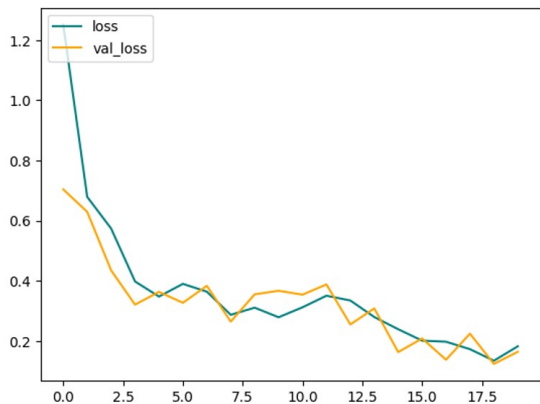


Fig. 5 Loss of the CNN Model vs number of epochs for 206 images and 30 locations

comes from the dataset of 206 images and 30 locations. The term ‘accuracy,’ as it’s used here, is how well the CNN can classify the unlabeled images on which it has already been trained (i.e. images from the training set). Validation accuracy is how well the CNN can classify images from the validation set. The classification accuracy of the CNN was measured based on a testing dataset, which includes unique images and locations from the training or validation datasets. However, Figure 4 reveals that CNN model is not being overfitted. This is because the accuracy is not consistently higher than the validation accuracy. In general, the blue and yellow lines representing accuracy and validation accuracy follow each other, which shows that the model is learning general features that are unique to high and low-flood-risk areas, rather than just memorizing the specifics of the images it trained on. This is because the model was not trained on the validation set, yet the validation accuracy was comparable to the accuracy of the training set.

Figure 5, which shows the loss of the model vs number of epochs for the dataset with 206 images and 30 locations, highlights how quickly convergence was achieved. Loss quantifies the error of the model, which is the difference in the predicted guess from the actual answer. Convergence, which is when the training process reaches a stable state and the weights and biases are changing little, is achieved when the loss function reaches a minimum. This means the model has reached peak accuracy with the fewest possible errors. In this case, the loss of the model decreases rapidly over the first 3 epochs, then slowly over the next 14. The loss reaches a minimum around 17 epochs, which means the model reaches convergence then.

Figure 6 shows one of the images from the training set from Rajkanika, India. Figure 7 shows the feature extraction done on the image. The lighter-colored areas are the parts of the image that the model is focusing on. There are two layers represented here, and each layer has multiple filters, which each focus on a different part of the image. The lighter parts of each



Fig. 6 Rajkanika, India (Flood-Prone)

filter highlight what the CNN model focuses on to determine the flood risk of a region. Thus, it can be inferred that the model focuses on the greenery, the dirt, and the water in the image since these features are all lighter in color in the filters. Doing feature extraction for other images seems to confirm this. In addition, white reflective features, such as the coastline or infrastructure, are also focused on. The CNN may focus on these kinds of features since desert sand looks white and is reflective. The training data had deserts with reflective sand, so the CNN possibly associated this reflection of light with low flood-risk regions. The model may erroneously classify anything that is white and reflective as desert sand. Desert sand is of course indicative of a non-flood-prone region.

Discussion, Limitations, & Recommendations for Future Work

Overall, the CNN model did a good job at training and classifying satellite images as flood-prone or not flood-prone. The accuracy of the model in every case was always above 50%, which means it did better than just randomly guess. This implies that the model has learned the important things to look for when classifying an image. Thus, coming back to the hypothesis - yes, a Convolutional Neural Network (CNN) be used to identify the factors that can cause some areas to be flood-prone and other areas not to be flood-prone using satellite imagery. It can also predict the flood risk of a new area based on its satellite image since the model had a high classification accuracy on the testing dataset.

Two important conclusions can be made from this research.

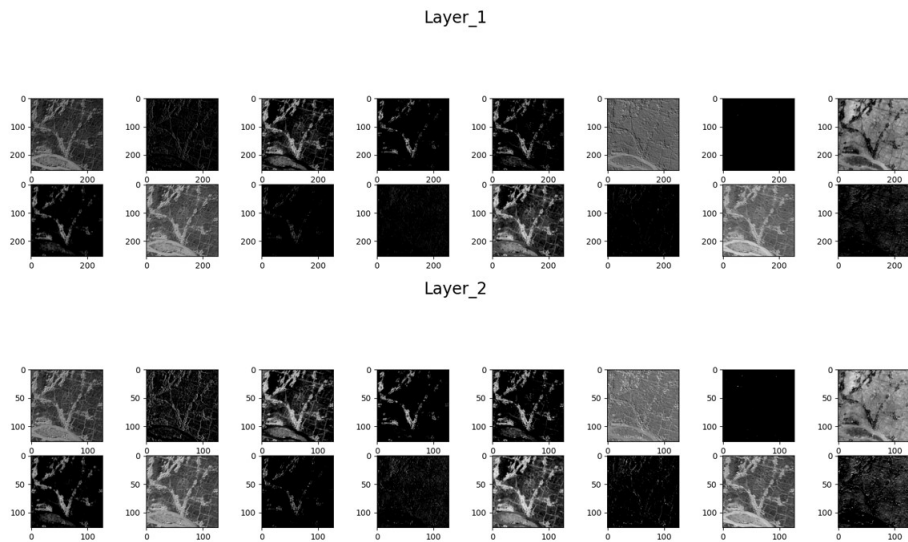


Fig. 7 Feature Extraction for Rajkanika, India

They are that increasing the number of images in the training dataset generally increases the accuracy of the model. Also, increasing the number of locations in the training dataset generally increases the accuracy of the model. Having a larger number of images and locations in the training set increases the diversity of the training set. An increase in diversity means there are more features of an image for the CNN to analyze, making the model more comprehensive. Every region has different factors for being either flood-prone or not, so more diversity in the dataset helps the model learn all these factors, allowing it to make more accurate predictions. The trend lines from Figures 2 and 3 show that for each location added to the training set, the accuracy increases by 1.04%, which is 25 times more than the 0.0462% increase when an image is added. This suggests that adding a new location to the training set increases the diversity of the dataset more than adding new images.

Figures 2 and 3 illustrate that the maximum accuracy of the CNN was 93.22%, which occurred with 38 locations and 246 images. As mentioned in the results section, this accuracy was ascertained to be an outlier via the interquartile range method. This could be because there was a high number of locations in the training set. And as determined in the previous paragraph, adding new locations to the training set highly increases the diversity of the training set, which could lead to this high accuracy. In addition, the lowest accuracy of the CNN was around 70%, which occurred with 26 locations and 246 images. Adding locations caused the CNN to have high accuracy and removing locations caused it to have the lowest accuracy, highlighting how the outside influence that the number of locations in the training set has on accuracy.

The maximum number of images and locations used in the training set was 306 and 38 respectively. This is a small amount

in comparison to CNNs used for real-world data analysis, which would have thousands of images and hundreds of locations. However, the model still had decently high accuracy, indicating that few resources are needed to train the CNN. A CNN model used for classifying whether regions are flood-prone or not would eliminate the need for costly governmental surveys or scientific studies, thus saving valuable resources²⁰.

One thing to note is that accuracy is used as the major evaluation measure in this research. This is because there was a similar number of flood-prone and non-flood-prone images in the training set, so the training set was balanced. In addition, a false positive and false negative have equally bad outcomes. Assuming flood-prone means a positive result, a false positive means the region is not flood prone but is mistakenly classified as one. Thus, based on the classification of flood-prone, unnecessary resources would be devoted towards the region. A false negative means the region is flood prone but it is classified as not. This is also a problem because the region would not get the resources it needs to bolster its flood protection services.

Another thing to note is that the CNN model never overfitted the data. Overfitting occurs when the model trains itself too closely to a particular set of data, which decreases its ability to take new data into account. Overfitting occurs when accuracy is consistently higher than validation accuracy. This means that the CNN model is better at classifying images that it has seen before (ones from the training set) than new images it has not seen before (ones from the validation set). For example, imagine that in the training data, every flood-prone image has a palm tree in it. The model may then assume that any satellite image with a palm tree would be flood-prone. In reality, this is an incorrect assumption, but the CNN model has calibrated its weights too closely to the training data to realize this assumption

is wrong. Overfitting is a problem because it means the model has a hard time analyzing new data it hasn't seen before, leading to inaccurate predictions.

This research was limited by time and resources and thus could not be performed on a larger scale. Specifically, time constraints inhibited the ability to add or remove even more images and locations to the data sets. This was because images had to be manually transferred from the data pipeline to the folder where the training data was stored. This allowed the images to be scanned to determine if they were usable (e.g. no clouds or incompletely generated images). However, this process was also time-consuming. As seen in Figures 2 and 3, the accuracy of the CNN model increased as the number of images and locations in the training set increased. It would have been interesting to see if this pattern continued when many more images or locations were added and removed, or if the accuracy eventually leveled off at some point. The accuracy may have leveled off since at some point, adding more data to the training set will not help if there's already enough data for the CNN model to make highly accurate predictions.

Furthermore, different combinations of images and locations could have been tested to see what combination is the most effective at increasing the CNN's accuracy. If a model like this is to be used for practical purposes, it is imperative that it has the highest accuracy possible. In this research, the number of images and locations in the training set was changed while the other parameter was kept constant. But what if both parameters were changed simultaneously? Changing both parameters simultaneously may produce a large effect on the accuracy, or their effects may "cancel out", and the accuracy wouldn't change by much. As seen in Figures 2 and 3, the highest accuracy came when the CNN model had 306 images or 38 locations, with 81% and 93% respectively. A model trained on a dataset with 306 images and 38 locations may have an accuracy higher than 93% due to additive effects or between 81% and 93% due to effects being canceled out.

Another limitation is that satellite images do not capture everything about a particular region. They can get a general view of the area's features, but they cannot capture every individual bush, tree, dirt mound, piece of infrastructure, etc. Also, in this research, only satellite images from one moment in a year for each location were considered. There weren't images from all 12 months of the year. As the year goes by, the flood risk of a location can change, such as through melting snow in the spring and dry spells in the summer. Adding on to this, satellite images also do not take into account other factors that can affect flooding, such as elevation, rainfall, temperature, wind patterns, etc. Future research involving satellite images to predict flood risk should take into account these external and temporal factors and processes to make more accurate flood predictions.

Also, future research needs to take into account imperfections in the CNN model. As stated before when discussing feature

extraction, the model could interpret white lustrous objects as desert sand, which could confuse it into incorrectly classifying a region with many of these kinds of objects as not flood-prone. Identification mistakes like this must be taken into account in future research to avoid erroneous classifications.

Conclusion

As noted before, during feature extraction, the greenery, dirt and sand-covered areas, and the water were all parts considered important by the model. Future research should be done to understand how different kinds, amounts, etc, of plants, dirt, and water affect the flood risk classification of a region. In addition, more detailed research on feature extraction should be done to understand what else the CNN looks for when classifying images.

Going back to the feature extraction done in this study, it seems to be that more greenery and water correspond to a higher chance of the region being flood-prone, and more dirt and sand would mean there is a higher chance of the region not being flood-prone. Plants and trees need lots of water to survive, which can come from floods. Dirt and sand with little plants are characteristic of deserts, which are not flood-prone. There is not enough water in deserts to support much greenery. Therefore, when the CNN model sees much sand with little greenery, it assumes that the region is not flood-prone. Of course, there are regions of the world that don't fit these stereotypes of greenery equals flood-prone and sand equals not flood-prone. The research in this study was broad in scope as it contained locations from all over the world, which can make it harder to generalize features from one location to the next. Many parts of the world have vastly different climate patterns. Thus, future research could only consider locations that are near each other and similar in climate. This would make it possible to generalize more factors affecting flood risk from one region to another, increasing the accuracy of the CNN model.

Ultimately, this research demonstrates that it is possible to use Convolutional Neural Networks to classify satellite images of regions as high or low flood risk. With high CNN classification accuracy, early action can be taken to assist high flood-risk regions with flood prevention measures. Additionally, feature extraction analysis helps researchers identify specific factors that cause a region to be at high and low flood risk. This allows governments, communities, and aid groups to take targeted measures to address the specific factor of an area that causes it to be flood-prone, as determined by feature extraction.

Acknowledgements

Thank you for the guidance of Eleftheria "Ellie" Fassman from Cornell University in the development of this research paper.

References

- 1 U. CRDDD, *The human cost of weather related disasters 1995-2015*, <https://www.cred.be/HCWRD>.
- 2 J. Rentschler, M. Salhab and B. Jafino, *Flood exposure and poverty in 188 countries*.
- 3 C. Reed, W. Anderson, A. Kruczkiewicz, J. Nakamura, D. Gallo, R. Seager and S. McDermid, *The impact of flooding on food security across Africa*.
- 4 B. Tellman, J. Sullivan, C. Kuhn, A. Kettner, C. Doyle, G. Brakenridge, T. Erickson and D. Slayback, *Satellite imaging reveals increased proportion of population exposed to floods*.
- 5 Center for Disaster Philanthropy, <https://disasterphilanthropy.org/disasters/2022-pakistan-floods/>.
- 6 Z. LI, H. LIU, C. LUO and G., *FU*.
- 7 R. Wilby and R. Keenan, *Adapting to flood risk under climate change*.
- 8 R. Goodrich, *Causes and control of major floods*.
- 9 B. Smith, *The role of vegetation in catastrophic floods: A spatial analysis. Bachelor of Environmental Science (Honours)*, <https://ro.uow.edu.au/thsci/65>.
- 10 S. Skakun, N. Kussul, A. Shelestov and O. Kussul, *Flood Hazard and Flood Risk Assessment Using a Time Series of Satellite Images: A Case Study in Namibia*.
- 11 M. Burke, A. Driscoll, D. Lobell and S. Ermon, *Using satellite imagery to understand and promote sustainable development*.
- 12 A. Tanim, C. McRae, H. Tavakol-Davani and E. Goharian, *Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning*, <https://doi.org/10.3390/w14071140>.
- 13 P. Picton, *What is a Neural Network?*, https://doi.org/10.1007/978-1-349-13530-1_1.
- 14 K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, arXiv preprint arXiv:1511.08458.
- 15 E. Portalés-Julià, G. Mateo-García, C. Purcell and L. Gómez-Chova, *Global flood extent segmentation in optical satellite images*, <https://doi.org/10.1038/s41598-023-47595-7>.
- 16 M. Panahi, A. Jaafari, A. Shirzadi, H. Shahabi, O. Rahmati, E. Omidvar, S. Lee and D. Bui, *Deep learning neural networks for spatially explicit prediction of flash flood probability*, <https://doi.org/10.1016/j.gsf.2020.09.007>.
- 17 N. Renotte, *Build a Deep CNN Image Classifier with ANY Images*, <https://www.youtube.com/watch?v=jztwpsIzEGc>.
- 18 *Global Flood Database*, <https://global-flood-database.cloudtostreet.ai/>.
- 19 U. Mall, *Change Event Dataset for Discovery from Spatio-temporal Remote Sensing Imagery*, <https://github.com/utkarshmall113/satellite-change-events>.
- 20 E. Nemni, J. Bullock, S. Belabbes and L. Bromley, *Fully Convolutional Neural Network for Rapid Flood Segmentation in Synthetic Aperture Radar Imagery*, <https://doi.org/10.3390/rs12162532>.