Corinne Smith, Nishaad Trivedi
Dr. Nick LaHaye
CPSC 293
21 May 2022

Flood Detection

Motivation

Annually, there is approximately $17 billion in property damage from flooding in the United States. This number is only expected to grow as climate change continues to shift the climate dramatically. This number is only expected to grow. Some estimations point to a 26% rise in the cost of damage over the next three decades.

It is essential to be able to understand the extent of flood damage because, for communities hurt by flooding, it is important to assess the toll of what got impacted. Knowing the status of critical infrastructure is another important factor for disaster response. Federal and NGO agencies can assess what damage is the most significant and then appropriate funds accordingly. They can also point teams to the hardest areas where they will have the largest impact. Further, when planning for disasters in the future, noticing trends of what critical infrastructure was most vulnerable can lead to better usage of prevention funds.

This information can be utilized by NGOs and federal agencies to assess the damage in a specific area at a greater speed if the process is automated. New data could be collected from automated drone gathering. Additionally, insurance companies can use this information as a way to determine quickly if they need to pay out claims.

Currently synthetic aperture radar sensors are utilized to detect flooding regions. However, the sensors are not high enough resolution typically to detect surface level details such as buildings. There are far less of the sensors on satellites compared to optical sensors, so developing a methodology to utilize optical sensors to determine damage would greatly improve the accessibility of flood information.

Data

The data we chose to use consisted of thousands of images of Houston buildings, where each image is centered on a building coordinate. These are satellite images, which were observed by optical sensors. The original data consisted of large pre-processed strips of images, with blacked-out sections and cloud cover. The paper's authors then cleaned the data to have no images with clouds or blacked-out sections.

Then, they referenced data provided by volunteers from Tomnod identifying damaged buildings from Hurricane Harvey in 2017. Their coordinates were used to center the images, which hover around the 128x128 dimension, with slight variation. The final training, validation, and test sets contain equal numbers of buildings before and after flood damage. This is how the labels of 'damage' and 'no_damage' were applied: the authors matched the damaged building images to images of the same buildings taken before the flooding. The resulting training set consisted of 10,000 equally balanced between damage and no damage images. There were in addition 2,000 validation images and 2,000 test images.

Figures 1 and 2:

Damage

No Damage

Original Paper

The original paper that this data set was created for was *Building Damage Annotation on Post-Hurricane Satellite Imagery Based on Convolutional Neural Networks* by Quoc Dung Cao and Youngjun Choe (2020). They tested out many variations of CNNs and data methods to create a final model with a validation accuracy of 98.06% and a test accuracy of 97.29%.

Chosen Methodology

We also chose to utilize a convolutional neural net because it is a robust methodology that utilizes a feedforward network to extract the key features of an image. Current state of the art solutions are similar to CNNs, like Mask R-CNNs and U-Nets. CNNs are one of the most effective methodologies of image recognition while

also balancing the computational weight of the model on a computer. There is no use in having an effective recognition model if it takes too long to run because then it would be too late to act upon the information.

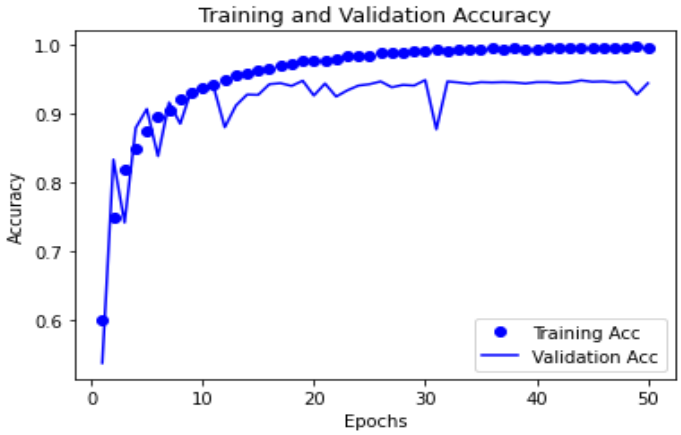
The key parts of the CNN model are the convolutional layers which utilize a filter to scan across the image to hopefully identify key traits that can then be used for whatever calculation is about to happen. The outputs from the filter are then added to a MaxPooling layer which then identifies which areas of the model require the most attention. Dropout is another useful part of the model because it enables nodes which are not doing much to get turned off, assisting in the generalization of the model and helping to prevent overfitting.

The final layer is a flatten layer which combines all of the max pooling layers together and then puts it into a feed forward neural net. Because this is a binary classification problem, the final layer needs to be a sigmoid because we are determining whether or not there is damage in the photo.

Model Creation

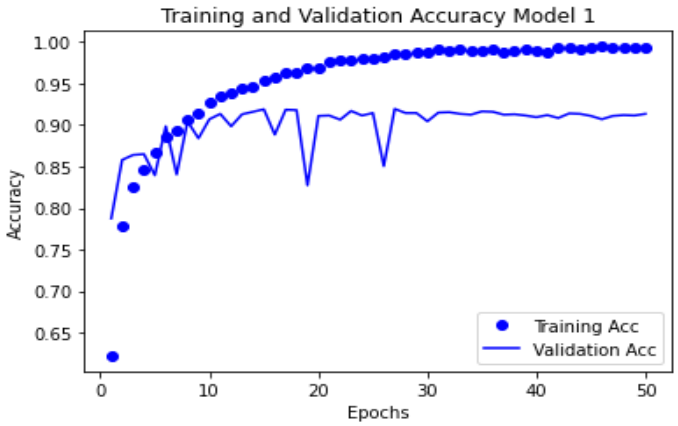
Our goal was to create a model with the highest classification accuracy possible. We made models with and without Dropout and different amounts of layers. First, we established the baseline model with one convolutional layer and one MaxPooling layer to diminish possible overfit. We chose to use the 'ReLu' activation function in our convolutional layer(s) due to its popularity with image classification and set the window to be 3x3, also common for image classification. This model had a 99.6% training accuracy and 94.45% validation accuracy.

Figure 3: 1 convolutional and 1 pooling layer (3x3 window)



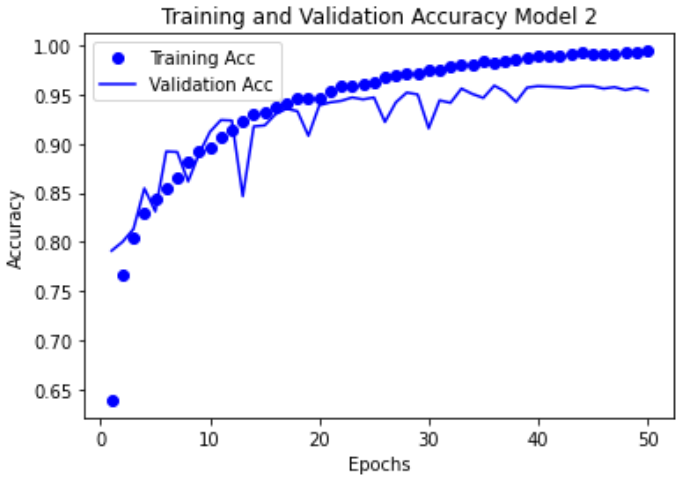
For our next model, we wanted to test out a 5x5 window in our convolutional layer. Increasing the window lowers the computational intensity of the convolutional neural network. The training accuracy was similar to the baseline model, 99.34%. Unfortunately, the validation accuracy bottomed out very quickly at 91.35%, which demonstrated model overfit.

Figure 4: 5x5 window



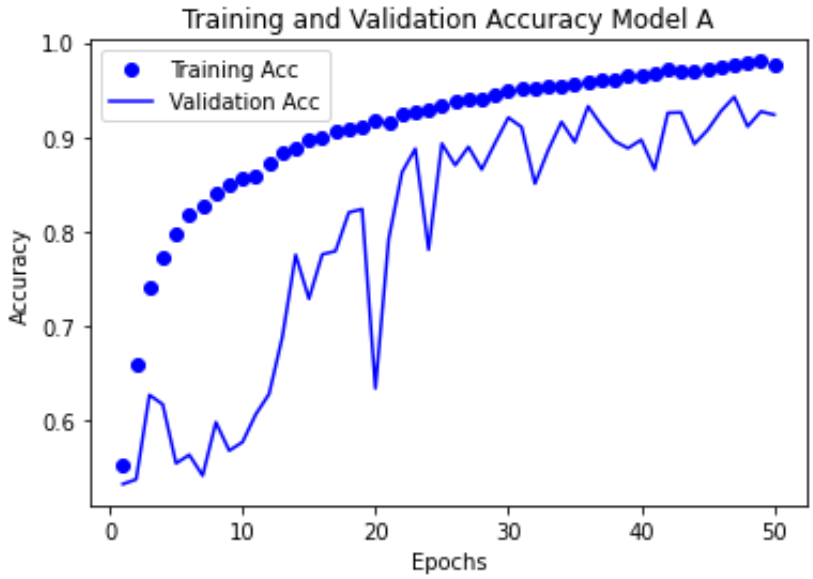
This validation accuracy issue confirmed to us that maintaining a 3x3 window was a better choice for the next models. While we still wanted to minimize complexity, we were curious as to what another convolutional and MaxPooling layer would impact in terms of model accuracy. The training accuracy diminished slightly from the baseline model at 99.43%, but the validation accuracy increased to 95.40%.

Figure 5: 2 convolutional, 2 pooling layers (3x3 window)



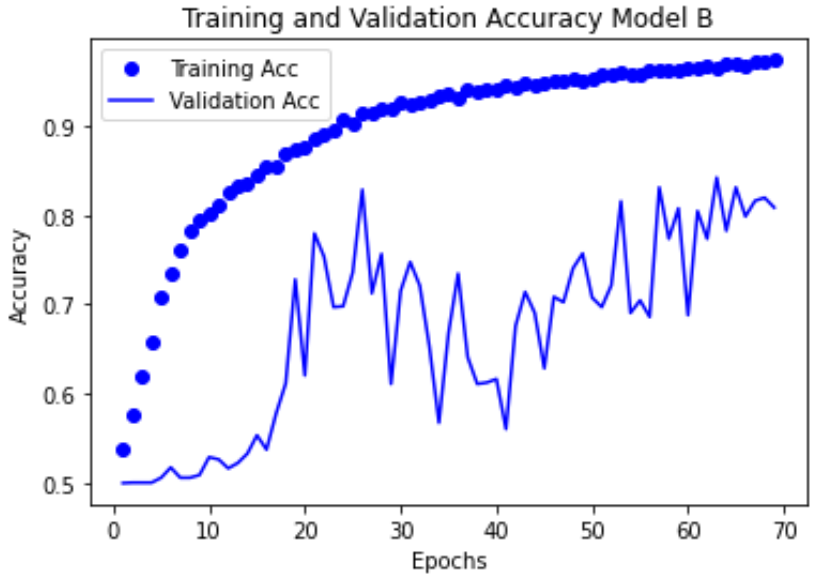
Noticing the overfit present (in all models so far), we decided to add in dropout layers. For our Model A, we added 3 layers of dropout into Model 2, one of 0.25 after each MaxPooling layer and one of 0.5 at the end. This left us with a lower training accuracy of 97.72% and a validation accuracy of 92.35%.

Figure 6:



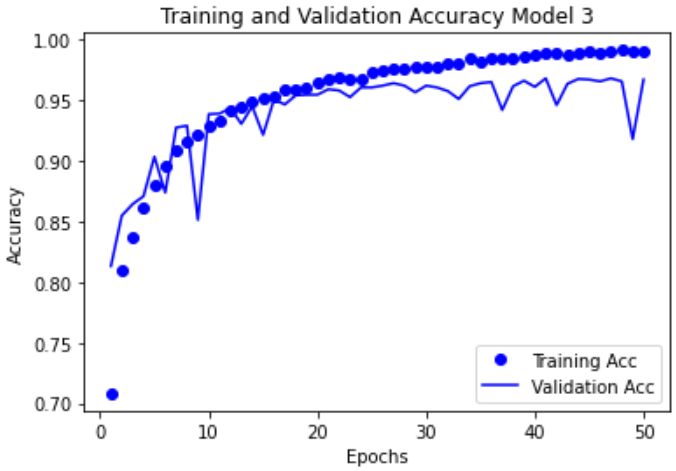
We then tried to up all the Dropout to 0.5, but the training accuracy was 97.36% and validation accuracy plummeted to 80.85%. It had the worst magnitude of overfit over all the models!

Figure 7:



We also wanted to test out one more addition of a convolutional and a MaxPooling layer. With 3 convolutional and 3 MaxPooling layers, the training accuracy was about 99.01%, which was lower than the baseline and Model 2. However, its validation accuracy was 96.65%. Since this model had the highest validation accuracy and least amount of overfit, we used it for testing, receiving an accuracy of 98%.

Figure 8: 3 convolutional, 3 pooling layers (3x3 window)



Model Performance

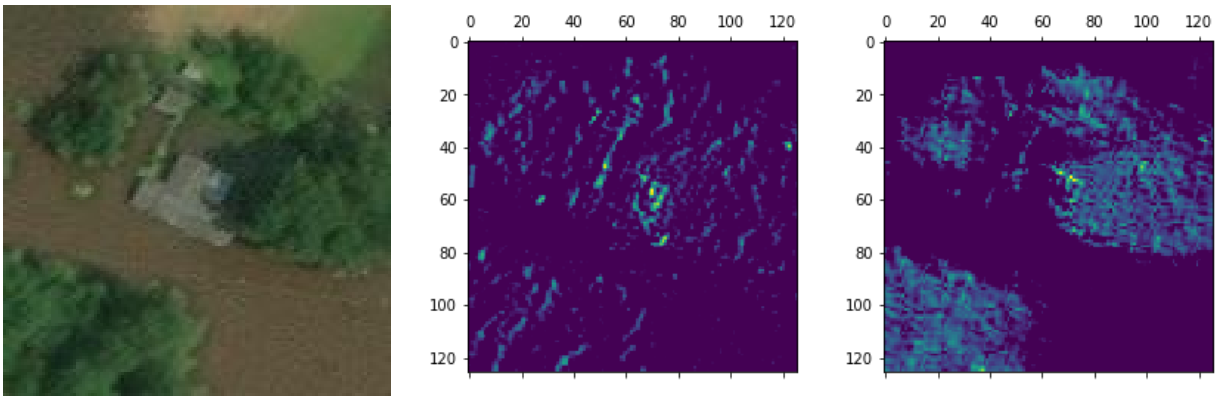
The best model we created had the 6 total layers of convolution and max pooling. We were surprised that adding in dropout to our earlier model created a worse performance or maintained about the same amount of overfit in the model. However, even with the overfit from about a 2-3% difference in training versus testing/validation accuracy, the deepest model performed the best.

Model Visualization

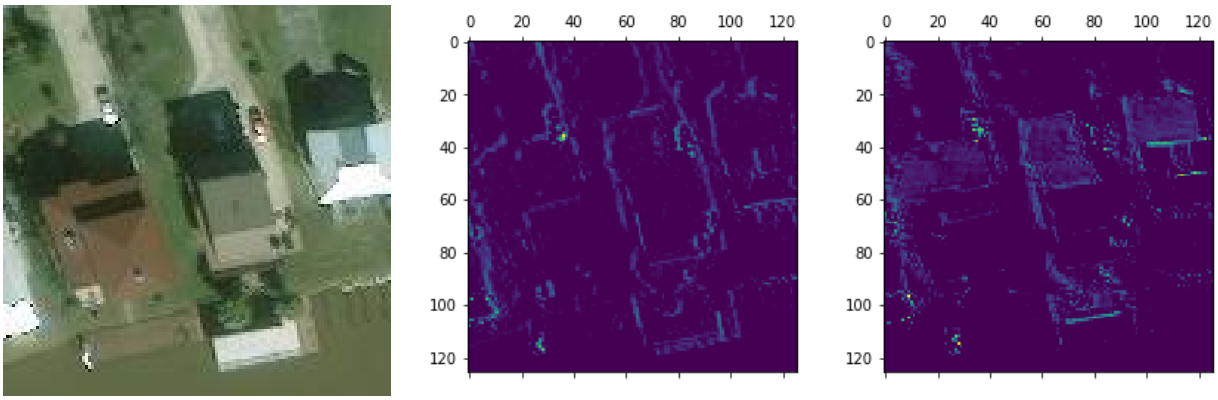
We were able to visualize the intermediate activations, or the feature maps that the CNN uses. Below are two examples of these maps on both a ‘damage’ and ‘no_damage’ image.

Figure 9:

Damage



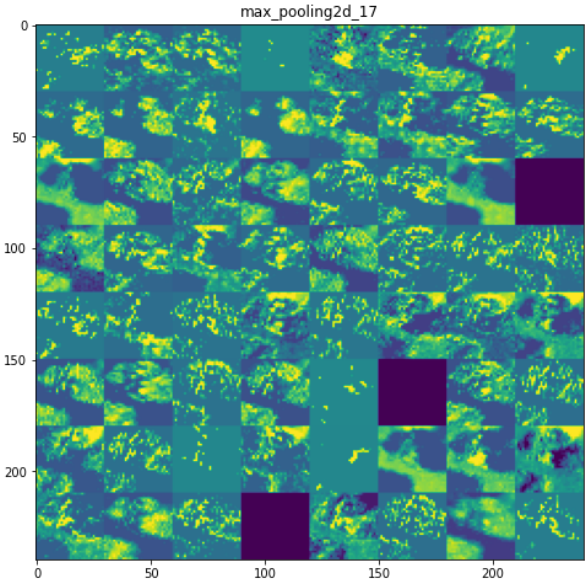
No Damage



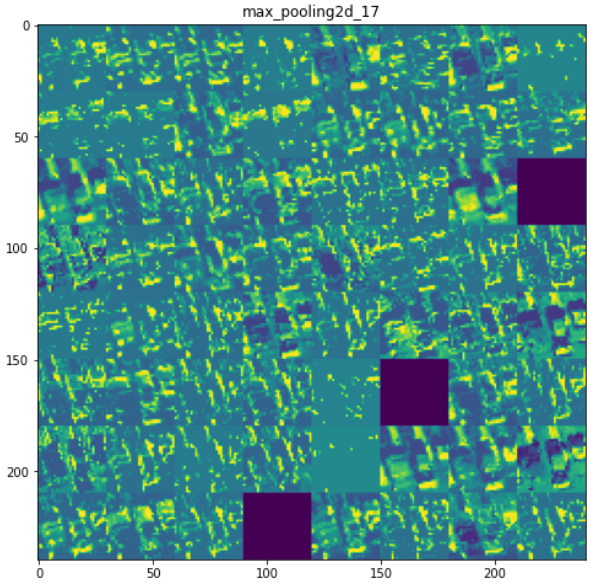
These initial visualizations seem to show some of the tree texture in the ‘damage’ image and the roof borders/gutters/edges of the three buildings in the ‘no damage’ image. Diving in deeper, we can take a look at the set of intermediate activations from the second MaxPooling layer. For the damage image, the model highlights the contrast between the building and the dirt/ground present. The texture of the trees surrounding the building is also visible. With the ‘no damage’ image, the three distinct roofs appear to be the focus of the filters, especially their shadows from the sunlight.

Figure 10:

Damage



No Damage



Conclusion/Reflection

With a convolutional neural network that had a reasonably small number of hidden layers, we were able to create a model with a 98% testing accuracy for hurricane damage. This is very promising, but we still have overfit in the model. To improve this project, we would like to test out more methods of regularization like L2, a method used in the original paper. We would also consider trying to implement data augmentation, as the authors' most successful model utilized that.

In addition to performance increases in the future, there is also a hope to increase the speed of the model in recognizing satellite images. The ideal scenario would be through caching and other methodologies, creating a program which is able to run on a drone, and the drone would then feed back out 'damage' or 'not damage'. Deployment on a drone would enable a much lower cost for image collection in addition to the ability to quickly deploy this technology while in the field. Because drones can be automated, there is the possibility of having several automated drones all fly around collecting information.

One issue with the model at the moment is that it mainly looks at the roofs of buildings to determine whether or not the area is flooded. This tendency will pose an issue if the model were to be applied to other areas of the world where roofs are not the same, such as in rural villages. The same issue arises with the model also picking

up on the landscape and foliage. To combat this in the future, more flooding data similar to this data from around the world should be incorporated into the training set which would help create a more generalized model. This model has promising applications, and with more training it could be very useful for future disaster relief efforts.

References for Paper and Presentation

- https://ieee-dataport.org/open-access/detecting-damaged-buildings-post-hurricane-satellite-imagery-based-customized
- https://arxiv.org/pdf/1807.01688.pdf
- https://www.frontiersin.org/articles/10.3389/frai.2020.534696/full
- https://www.scientificamerican.com/article/new-maps-show-us-flood-damage-rising-26-percent-in-next-30-years/
- https://www.bbc.com/news/av/world-europe-61325769
- https://www.reuters.com/world/asia-pacific/half-million-face-flood-evacuation-sydney-braces-more-heavy-rains-2022-03-02/
- https://www.npr.org/2021/07/25/1020342822/flooding-continues-to-devastate-zhengzhou-city-in-central-china

References for Code

- Deep Learning with Python by Francois Chollet
- https://towardsdatascience.com/convolutional-neural-networks-in-practice-406426c6c19a
- https://www.tensorflow.org/guide/keras/save_and_serialize
- https://stackoverflow.com/questions/66221788/tf-gradients-is-not-supported-when-eager-execution-is-enabled-use-tf-gradientta
- https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/
- https://stats.stackexchange.com/questions/147850/are-pooling-layers-added-before-or-after-dropout-layers