Spring 2022 Practicum Final Deliverable

05/13/2022

Zhaokailu Gu, Yang Hu, Dan Li, Rui Lu,
Naijia Wu, Jeffray Tsai, Liam Tay Kearney,
Kimberly Shan

**Quantitative Methods in Social Sciences,
Columbia University**

1. Introduction
2. Our Approach
3. Business Impact
4. Limitation & Improvement
5. Reflections

1.1 Our Team

Quantitative Methods in the Social Sciences (QMSS)

- Master's of Arts program within the Graduate School of Arts and Sciences at Columbia University
- An innovative, flexible, interdisciplinary degree focusing on quantitative research techniques and strategies

Jeffray Tsai
Project Management
Business Application

Zhaokailu (Cece) Gu
Data Collection,
Model Improvement

Naijia (Haylie) Wu
Model Improvement,
Business Application

Yang Hu
Model Improvement,
Business Application

Dan (Jessica) Li
Model Architecture,
Model Improvement

Xia (Kimberly) Shan
Exploratory Data Analysis

Rui Lu
Model Architecture,
Model Improvement

Liam Tay Kearney
Data Collection,
Preprocessing, ETL

1.2 Project Overview

Context

- Flooding has caused tremendous losses and damage in the United States in recent years
- Accurate prediction of flood events enables more effective response, and mitigation of losses
- Adopting cutting-edge **Deep Learning Image Classification Models** is of critical importance

Project Overflow

- Data Collection: Satellite Images (Planet) + Flood event records (NOAA)
- Data Preprocessing: Target input timespan; Balance data structure
- Build Model Structure: Convolutional neural network (CNN) deep learning image classification model
- Model Improvement: Pseudo labeling: Increase input data size; Confounder control: Increase input quality; Model fine tuning
- Business Application: Measure economic impact; Region expansion

Outcome & Takeaway

- Our final CNN model achieves an accuracy of **81.06%**
- We apply the model to to a **vehicle flood loss assessment** to gauge potential mitigated losses
- The model has potential application to **regions which have not experienced significant historical flooding** (and thus have limited image data available) but may experience increased flooding in future years due to climate change related threats.

1.3 Region Selection

Region Choice: Miami-Dade County

Data availability

Flood possibility

Business application

Three geographies shortlisted

New York, NY

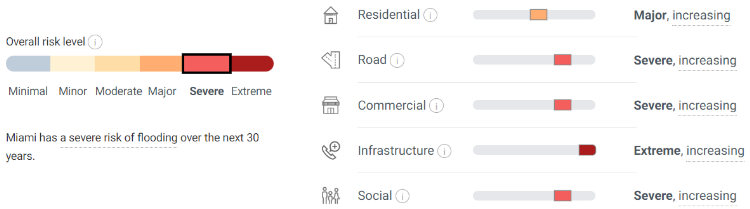
Miami, FL

Houston, TX

Immediacy of flooding

Miami, FL

Floodfactor.com: assess past floods, current risks, and future projections



1.3 Region Selection

Excellent market opportunity: Base on current flood system, without our flood model prediction, flood events are estimated to cause $220.9 billion yearly loss for Miami-Dade

Loss Estimation Model

Current Yearly Loss Estimation

Opportunity

Loss Per Flood Event

Urban Damage

Rural Damage

Infrastruct-ure Damage

×

Average Yearly Flood Frequency

Yearly Flood Loss ($)

$4.7 B loss per flood

47 floods per year

$220.9 billion in losses per year

How much loss we can avoid with accurate model prediction?

Our Model →

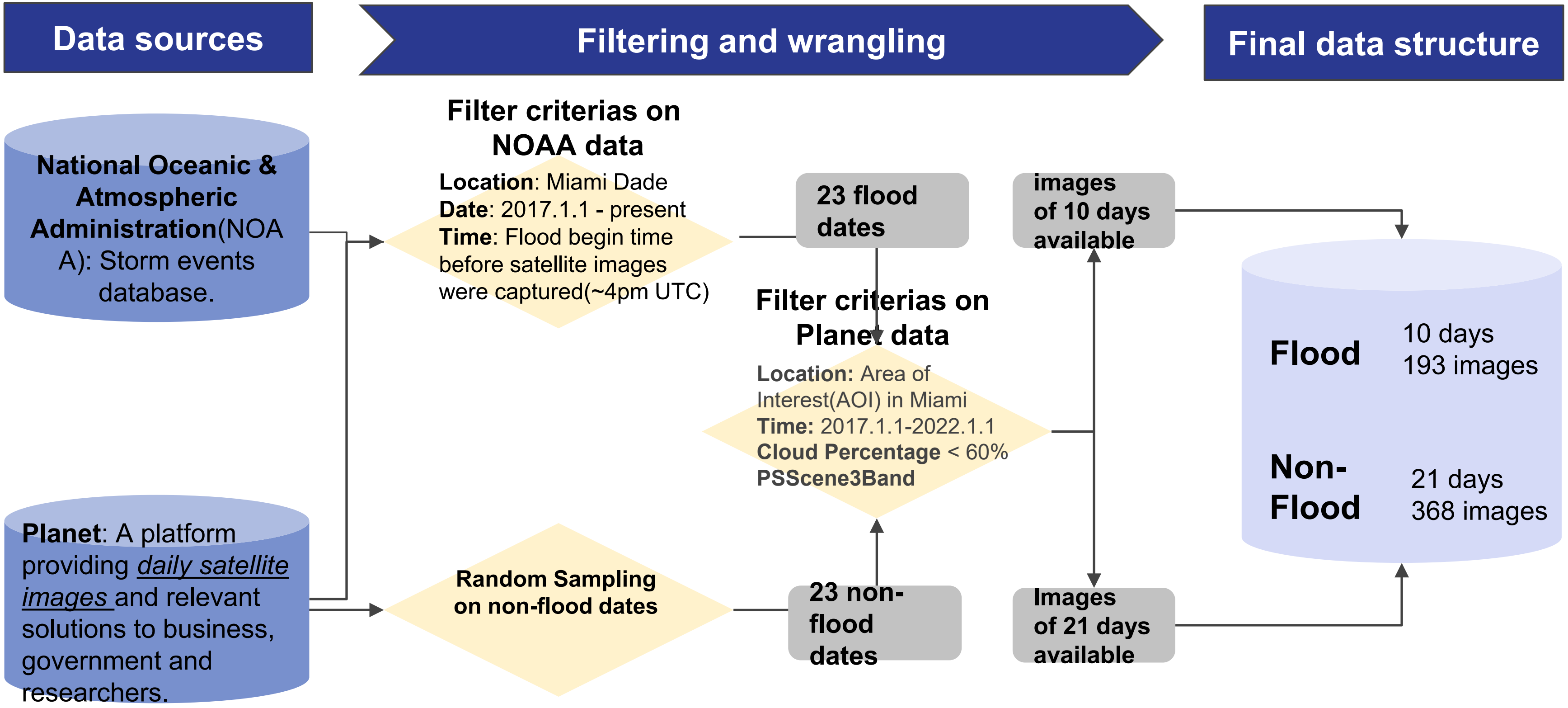
$???

Source: NOAA flood records 2017-2022

1. Introduction
2. **Our Approach**
3. Business Impact
4. Limitation & Improvement
5. Reflections

2.1.0 Our Approach - Data

Data Overview

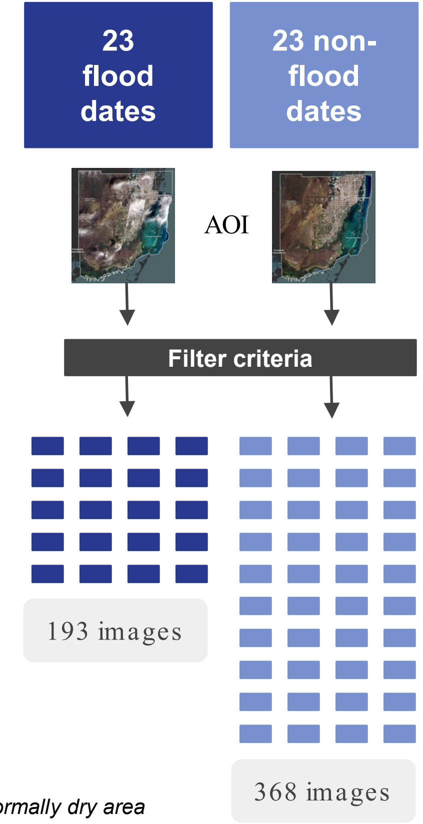


2.1.1 Our Approach - Data

A brief review of data source for historical flooding events - NOAA

- National Oceanic and Atmospheric Administration Storm Events Database (source data from National Weather Service)
- Timestamps of all **flood events** in Miami-Dade county with precise start and end times; we choose events from **January 2017 onward**

ID	County	Type	Begin Date	End Date
844788	miami-dade	Flood	08-JUL-19 13:30:00	08-JUL-19 15:30:00
856225	miami-dade	Flood	11-OCT-19 17:00:00	11-OCT-19 19:00:00
984803	miami-dade	Flood	17-SEP-21 17:45:00	17-SEP-21 19:45:00
849890	miami-dade	Flood	14-AUG-19 13:00:00	14-AUG-19 15:00:00
886879	miami-dade	Flood	17-MAY-20 15:00:00	17-MAY-20 18:00:00
896929	miami-dade	Flood	26-MAY-20 18:30:00	26-MAY-20 21:30:00
869837	miami-dade	Flood	23-DEC-19 02:15:00	23-DEC-19 07:15:00
843153	miami-dade	Flood	24-JUN-19 14:50:00	24-JUN-19 16:00:00
930128	miami-dade	Flood	09-NOV-20 20:00:00	13-NOV-20 19:00:00
892252	miami-dade	Flood	26-MAY-20 16:30:00	28-MAY-20 21:00:00



Flood event: *"Any high flow, overflow, or inundation by water which causes damage. In general, this would mean the inundation of a normally dry area caused by an increased water level in an established watercourse, or ponding of water, that poses a threat to life or property."*

2.1.2 Our Approach - Data

A brief review of data source for imagery - Planet

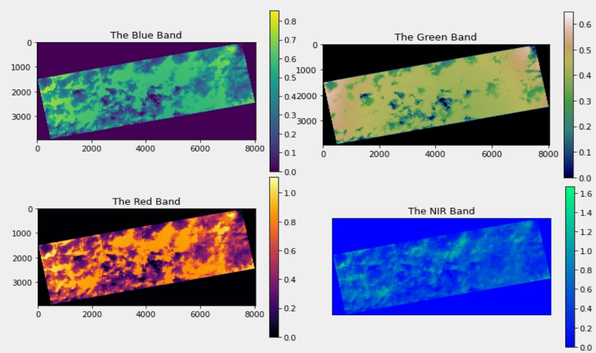
Introduction

Planet Platform Introduction:
- Provides highest frequency satellite data commercially available.

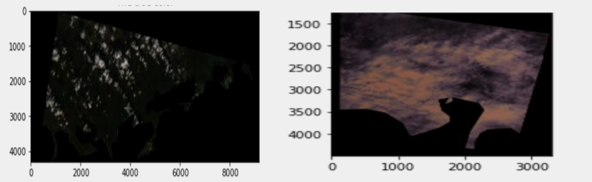
Filter
- Location: Miami, FL
- Time: 2017.1.1-2022.1.1
- Cloud Percentage < 60%
- PSScene3Band

Data Types:
- GeoTIFF, XML (meta), JSON (meta)

EDA



- BGRN, 4 Bands



Wrangling Methods

Discovery

Visualize Pixels, Analytic Bands

Structuring

Color Composite Image

Validating

Data is ready to be analyzed

2.2.0 Our Approach - Model Architecture

Model pipeline and improvement process

Model Selection	Model Improvement	Fine Tuning	Confounder Control
MVP with CNN	Pseudo Labeling	AI Model Share	Input Image Selection
Pre-trained Model	Image Augmentation	Supporting Team Participation	
	Pseudo Labelling and Image Augmentation		

More balanced data

Hyperparameter improvement

Model Optimization

Accuracy	66.67%	69.72%	70.52%	81.06%
Recall	78.38%	70.27%	62.10%	62.5%

2.2.1 Our Approach - Model Architecture

KPMG QMSS

Model Evaluation - Pre-trained Models

ResNet

- **ResNet** has two advantages:
  - Deep layers to capture image patterns
  - ***Skip connection*** to add the output from an earlier layer to a later layer to improve model performance

VGG

- **VGG** has two advantages:
  - A reward-winning model that trained based on large amount of data
  - Trained images of fixed size of 224*224 and have RGB channels (similar to our data)

MobileNet

- **MobileNet** advantage:
  - Enable to build and deploy neural networks in low compute environment

2.2.2 Our Approach - Model Architecture

KPMG QMSS

Model Selection - Pre-trained Models

Pre-trained Model	Training Accuracy	Validation Accuracy	Recall	Precision
ResNet	57.14%	58.97%	27.03%	45.45%
VGG	73.05%	62.26%	78.38%	55.77%
MobileNet	92.53%	50.00%	100%	48.05%

Overall Poor Model Performance

⇨ Low accuracy & recall & precision

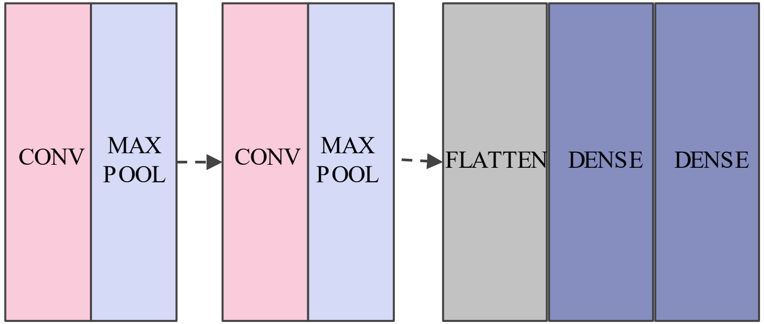
⇨ Low accuracy & Low precision & Long running-time

⇨ Low accuracy & Low precision Overfitting

- **Accuracy**: the number of correct prediction / total predictions → TP/(TP + TN)
- **Precision**: the number of correct positive predictions / total positive predictions → TP/(TP + FP)
- **Recall**: the number of correct positive predictions / total positives → TP/(TP+FN)

2.2.2 Our Approach - Model Architecture

Model Selection - MVP Model

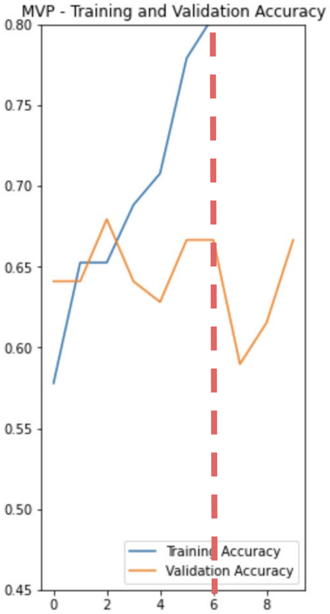


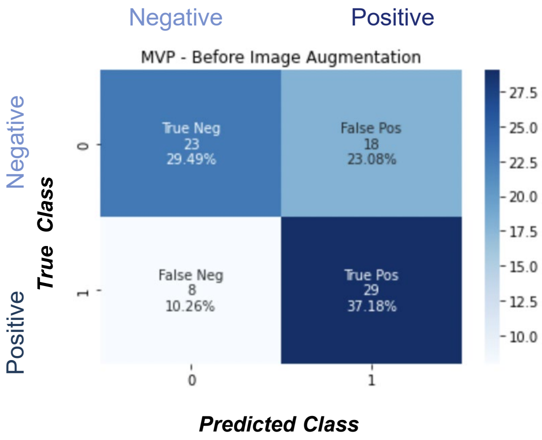
Hyperparameter Choices:

epochs = 6

batch_size = 32

Optimizer: Adam



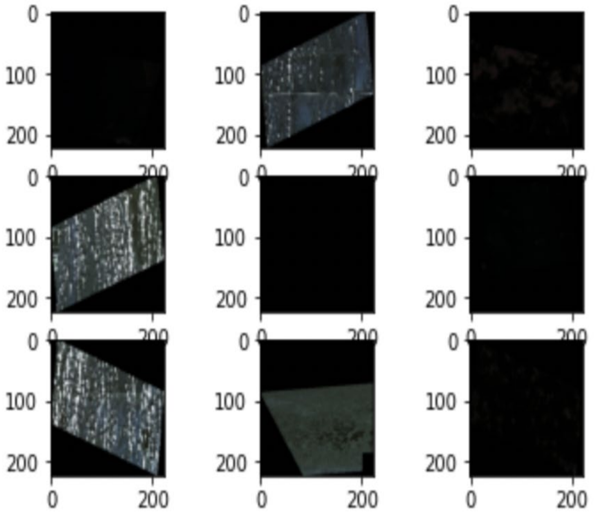


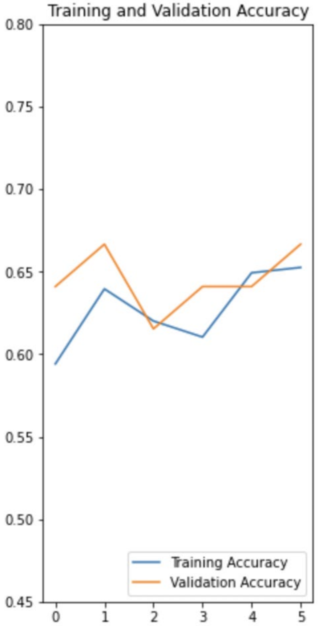
- Accuracy: 77.92% (Validation: 66.67%)
- Recall (validation): 78.38%
- Precision (validation): 61.7%
- The model is sample but overall effective

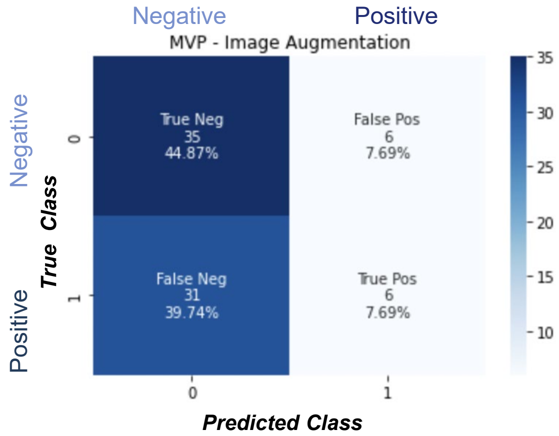
Takeaway:
- Compared to the pre-trained models, our CNN model shows an improvement in performace
- We could improve model performance (reduce overfitting & increase accuracy) using different techniques.

2.2.3 Our Approach - Model Architecture

Model Improvement - Image Augmentation







- Accuracy: 65.26% (validation: 66.67%)
- Recall (validation): 16.22%
- Precision (validation): 50.00%
- Image augmentation is not effective for current data

Takeaway:

- Although the image augmentation increased the variation of our flood datasets and solved the overfitting problem, the accuracy score fell.
- Low recall score implying model predicted flood event as non-flood event.

2.2.4 Our Approach - Model Architecture

Model Improvement - Pseudo Labeling

Step 1

Generate training dataset

flood day images

4 days, 193 images

1 Labeled flood images of flood day

Non-flood day images

193 images

2 Labeled non-flood images of non-flood day

- Manually label the possible flood images from flood date with the flood start time before the image taken time.
- Randomly choose the images from non-flood date as the labeled non-flood images.

Model training & selection

Pre-trained MVP model

Model for pseudo label prediction

Use the labeled images to train the pre-trained MVP models and get a new pseudo label prediction model

Step 2

Pseudo label prediction

2.2.4 Our Approach - Model Architecture

Model Improvement - Pseudo Labeling

Step 2

Prepare unlabeled images to predict

Post 6 days' images of the selected flood events

Rest of images from the selected flood date

Pseudo label prediction

Model for pseudo label prediction

Model retraining

1 Labeled flood images of flood day

2 Labeled non-flood images on non-flood day

3 Pseudo labeled flood and non-flood

Choose unlabeled data because:

- **Flood day labeled non-flood images**: Highly likely to have more flood images that cannot be detected manually in Step 1.
- **Post- flood day images**: flooding is a longer-term event and may last a week or more.

Train the model with labeled data in Step 1 and new pseudo label data in Step 2.

Step 3

Model building

+

Model fine tuning

2.2.4 Our Approach - Model Architecture

Model Improvement - Pseudo Labeling

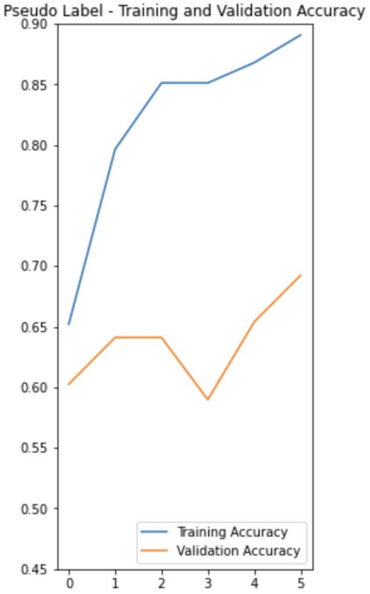
Flood Day Images

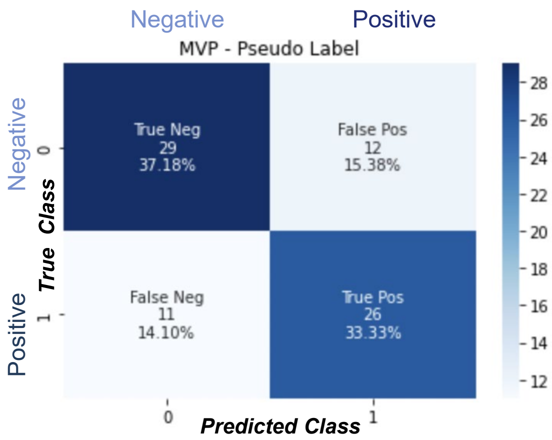
Non-flood images
Bad time / non-flood day

Unlabeled Images
6 days after flood event

Labeled Flood Images
Good Time

- **Good time:** Flood images taken after flood start time.
- **Bad time:** Flood images taken before flood start time.





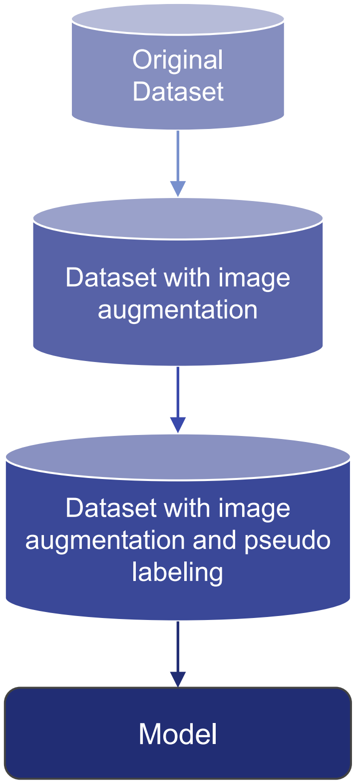
- **Accuracy**: 89.06% (validation 69.32%)
- **Recall** (validation): 70.27%
- **Precision** (validation): 68.42%

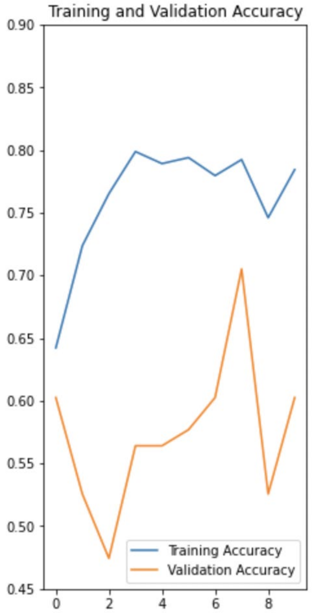
Takeaway:

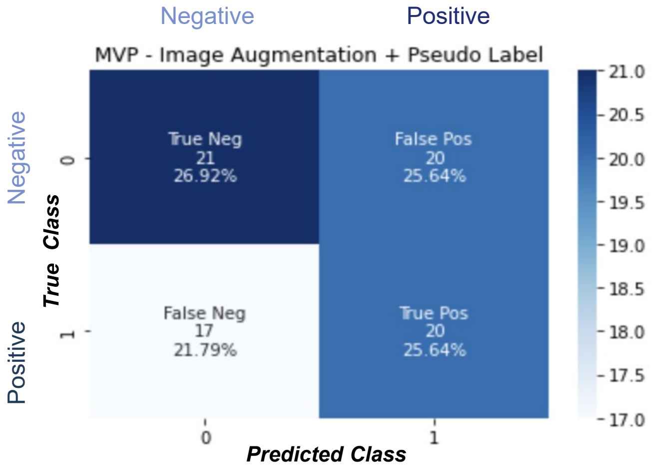
- The pseudo labeling process helped increase the accuracy, but led to overfitting
- The overall recall and precision scores are better

2.2.5 Our Approach - Model Architecture

Model Improvement - Image Augmentation + Pseudo Labeling







- **Accuracy**: 78.43% (Validation: 60.02%)
- **Recall**: 54.05 %
- **Precision**: 50.00 %

Takeaway:

- Current combination method is not effective for current data
- The overfitting problem remained and generated new fluctuation problems

2.2.6 Our Approach - Model Architecture

KPMG QMSS

Model Improvement - Fine Tuning

Tuning Roadmap & Model Structure

Step 1
Tune on each parameter one by one

Step 2
Gridsearch sets of parameters based on previous finding

Convolutional Neural Networks + Pseudo Labeling

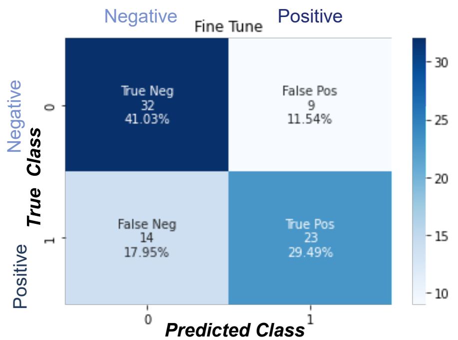
Convolution	**Pooling**	**Full Connection**	**Compile & Fit**
Activation Function ***Relu, tanh***	Pooling Method ***Max***	Activation Function ***Relu, tanh***	Optimizer
#Filters **48**	Strides ***(2,2)***	#Neurons ***256***	Loss Function
Kernel Size	Pooling Size		
Learning Rate ***0.001***			

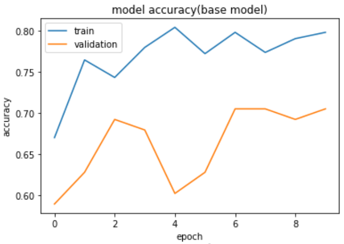
Best set of Parameters

learning_rate = 0.01, filter_number = 48, kernel_number = 1, d_strides = (3,3), activation_fun1 = 'relu', activation_fun2 = 'tanh', d_pool_size = 2, neuron = 128, Epoch = 10, batch_size = 28, optimizer-Adam

AI Model Share Platform Playground:
Request zg2382@columbia.edu to view the playground

Results





Learning Curve

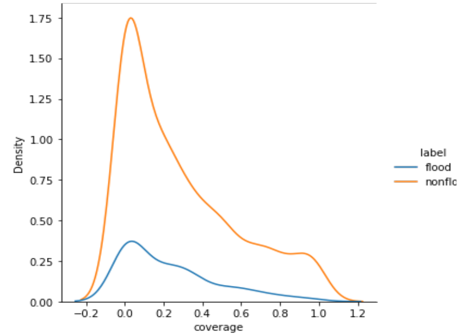
Training Accuracy: 79.79%;
Validation Accuracy: 70.52%
Recall: 62.10%; Precision: 78.04%

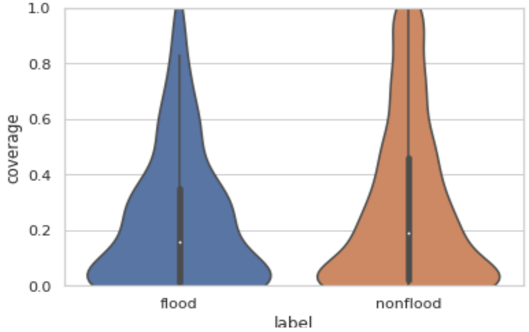
2.2.7 Our Approach - Model Architecture

The cloud coverage distribution difference between flood day images and non-flood day images demonstrates the existence of a cloud confounder problem in input data

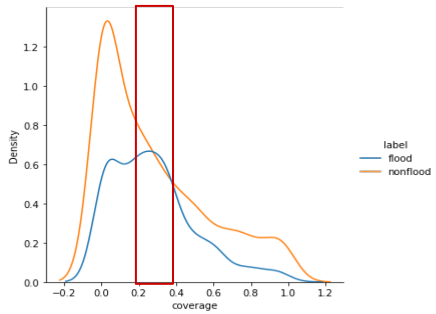
Cloud coverage distribution

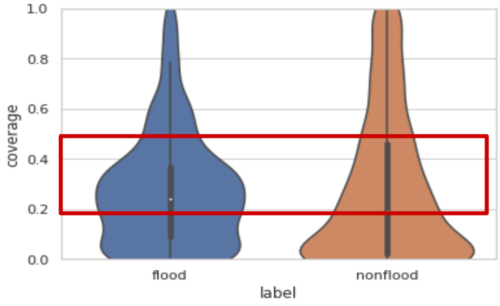
Before Pseudo Label





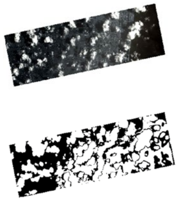
After Pseudo Label





Takeaway

Cloud is unevenly distributed in flood and nonflood datasets



Pseudo label exacerbates uneven distribution of cloud coverage

Cloud is a confounder

2.2.7 Our Approach - Model Architecture

Using only low cloud coverage images as inputs is a good solution for the cloud confounder problem

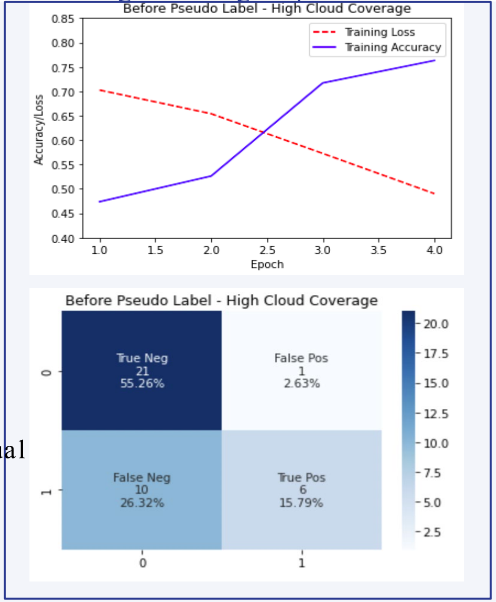
Control & Performance | **Takeaway**

High cloud coverage detriment the accuracy of the model

High cloud gets high false positive

High cloud group

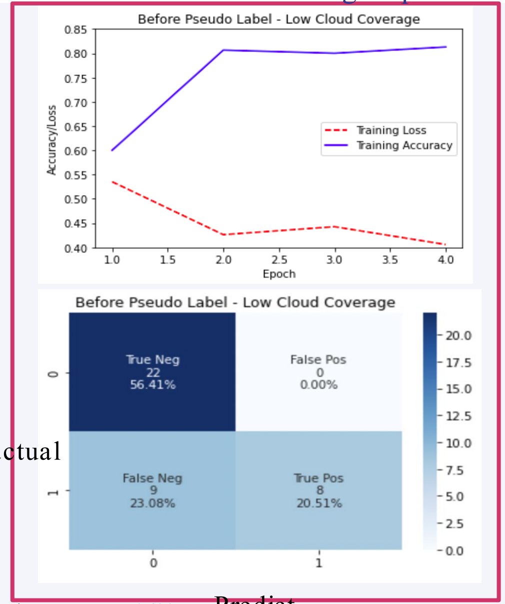
actual



Predict

Low cloud group

actual



Predict

Low cloud group: cloud coverage < 16%
High cloud group: cloud coverage > 16%

By limiting cloud coverage, accuracy increased by 10%

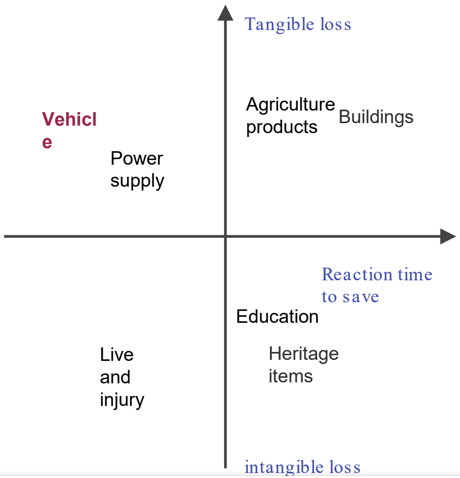
Less likely to predict false flood Precision increased by 14%

1. Introduction
2. Our Approach
3. **Business Impact**
4. Limitation & Improvement
5. Reflections

3.1 Business Impacts

Since our model is short-term prediction up to a daily update frequency, the model is well suited for application to business segments with short flood response time

Total flood damage cost
$220.9 B

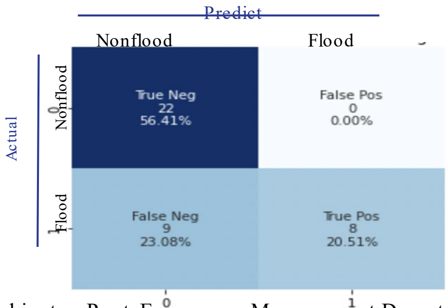


- Focus on the loss that have high monetary values and short reaction time since (few hours to a day) to save flooded loss
- Vehicle is a good segment

Cost Matrix for Vehicle

Actual \ Predict	Nonflood	Flood
Nonflood	**No Alert (Benefit)** 0	**Over Alert (Cost)** Car Flood insurance cost **$148 M**
Flood	**Misalert (Cost)** Vehicle Flood Loss: **$1.5M**	**Right Alert (Benefit)** 0

Confusion Matrix



Weighted average

Flood Vehicle loss saved

	Per flood	Year total
Loss saved	**0.34 M**	**16.27 M**

Business Application

- **Stakeholder: car owners**
- **Application: A software to report flood event daily for the car location and provide the evacuation regions to help the car owner leave the flood region in time to save the loss**

Source: FEMA, florida flood insurance.org, Miami-Dade Government Press Release, Washington Post, Emergency Management Department

3.2 Business Impacts

Applied to other regions

Model Use Case
How about other regions?
Non-flood regions can see an increase in floods by 2050
They potentially have poorer flood detection systems

Why our model?

Pre-Trained Model Advantages
Cost effective
Simple to implement
Can take advantage of high-frequency (daily) satellite data
Threshold-optimized
Model output can be used as an input for flood damage estimation

Insurance pricing	Public sector	Resource allocation

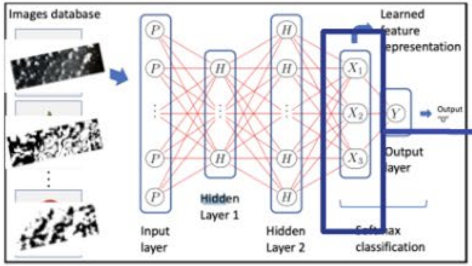
1. Introduction
2. Our Approach
3. Business Impact
4. **Limitation & Improvement**
5. Reflections

Confounding Adjustment: Casualty-aware Learn

Learned Feature → Counterfactual Feature → Train Logistic Regression → Predict

Learned Feature

Input:
Training set image (153,128,128,3)



Output:

- *Learned feature Xj*
- *Training confounder (cloud) Ctr*

Counterfactual Feature

Input:

- *Learned feature Xj*
- *Training confounder (cloud) Ctr*
- *Training set labels Ytr*

Linear regression model:
*Xj = beta(Ytr)*Ytr+beta(Ctr)*Ctr*

$\hat{X}^{*}_{tr}$

Output:

- *Train counterfactual features Xtr**
- *Test counterfactual features Xts**

Train Logistic Regression

Input:

- *Train counterfactual features Xtr**

Logistic regression classifier

$\hat{X}^{*ts}_{j}$

Output:

- *Counterfactual Labels*

Predict

$$\begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ . \\ . \\ . \\ 1 \end{Bmatrix}$$

Prevent neural networks from leveraging spurious associations induced by clouds
Straightforward to implement and computationally efficient

1. Introduction
2. Our Approach
3. Business Impact
4. Limitation & Improvement
5. **Reflections**