Damage Classification Final Presentation

By Alan Cheriyan, Andrew Orlosky, and Pat Marinich

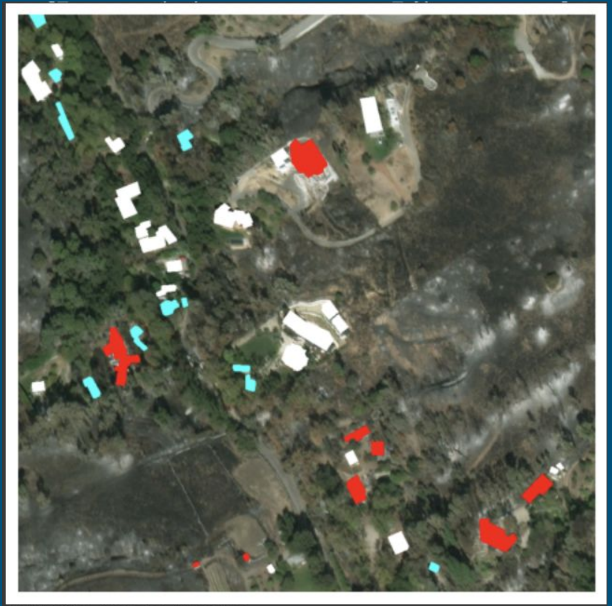
Quick Refresher - Motivation + Objectives

Motivation

- **Big Picture Goal:** Provide the most effective aid post-disaster.
- Capture the **magnitude of damage** in real-time with high accuracy

Objectives

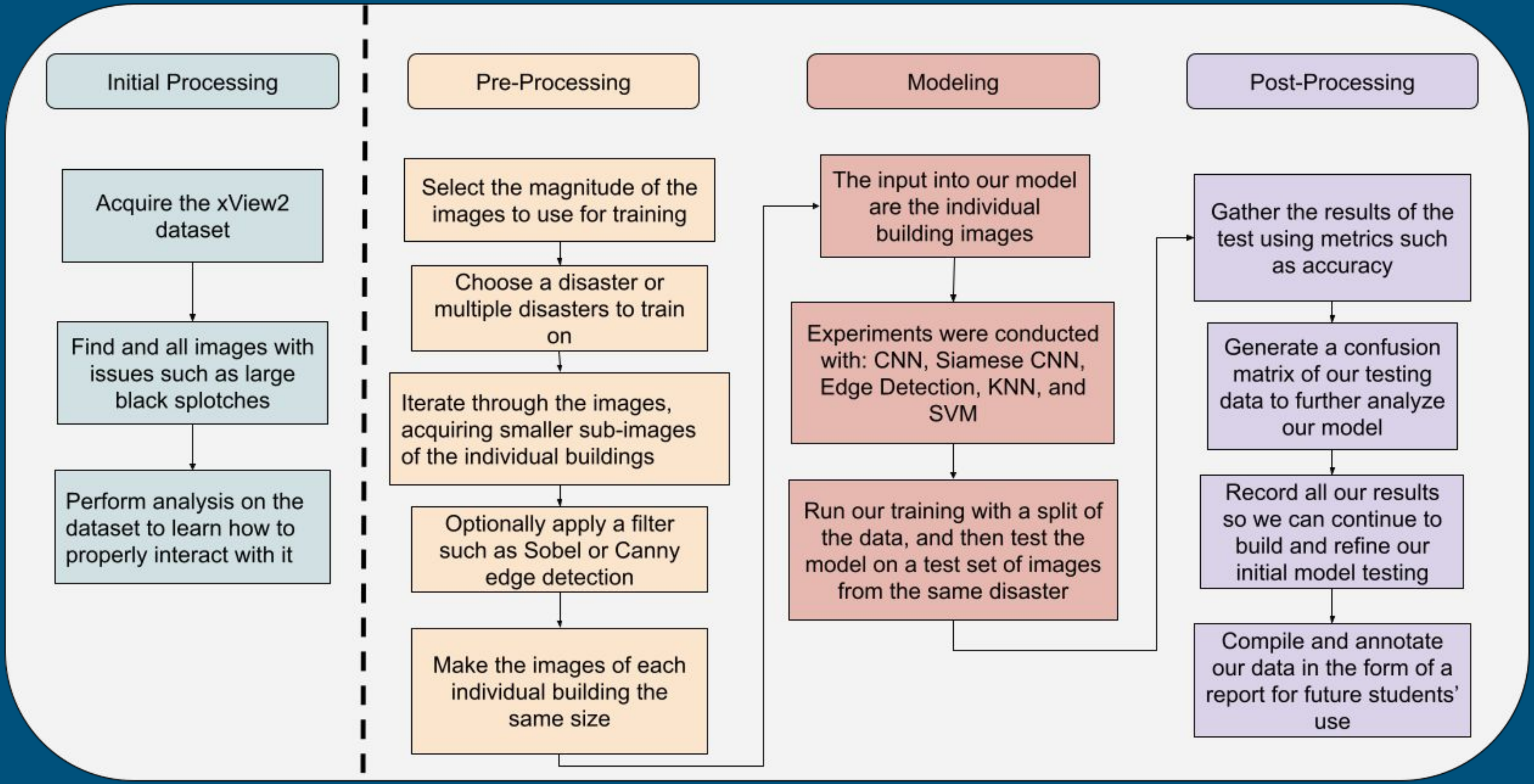
- **Perform Damage Classification**
  - No Damage
  - Minor Damage
  - Major Damage
  - Destroyed
- **Explore Models**
  - Investigate the effectiveness of different models



Building Damage Counts:

	Damage_Type	Count
0	no-damage	21
1	minor-damage	0
2	major-damage	0
3	destroyed	9
4	un-classified	11

Updated Pipeline

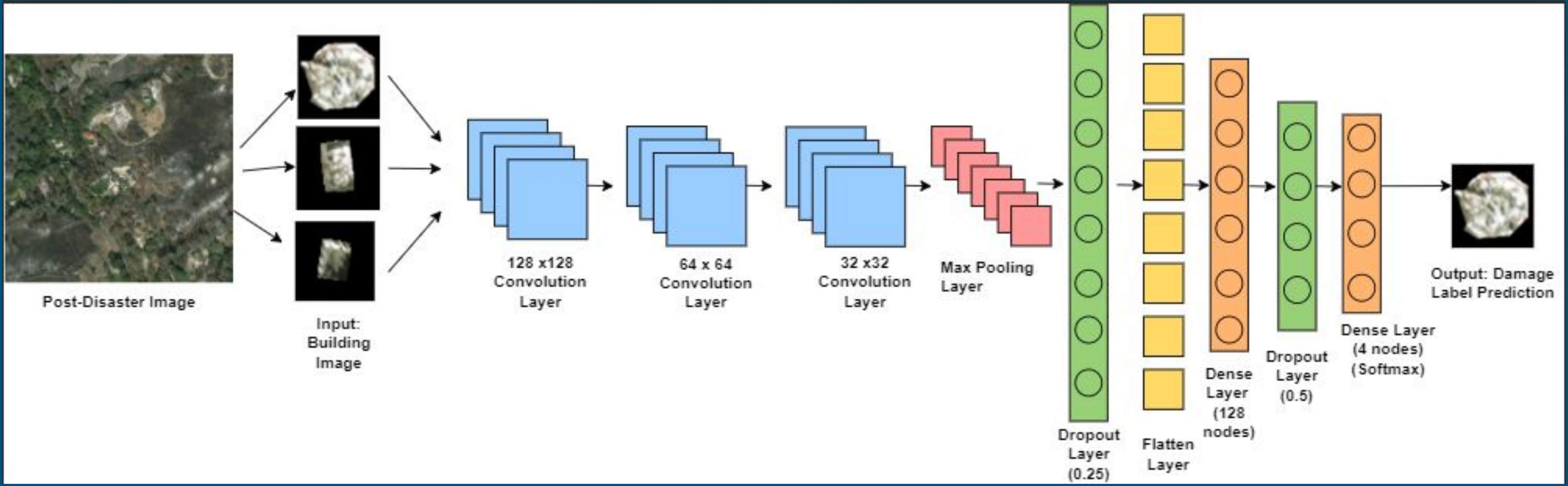


Broad Results - Accuracy

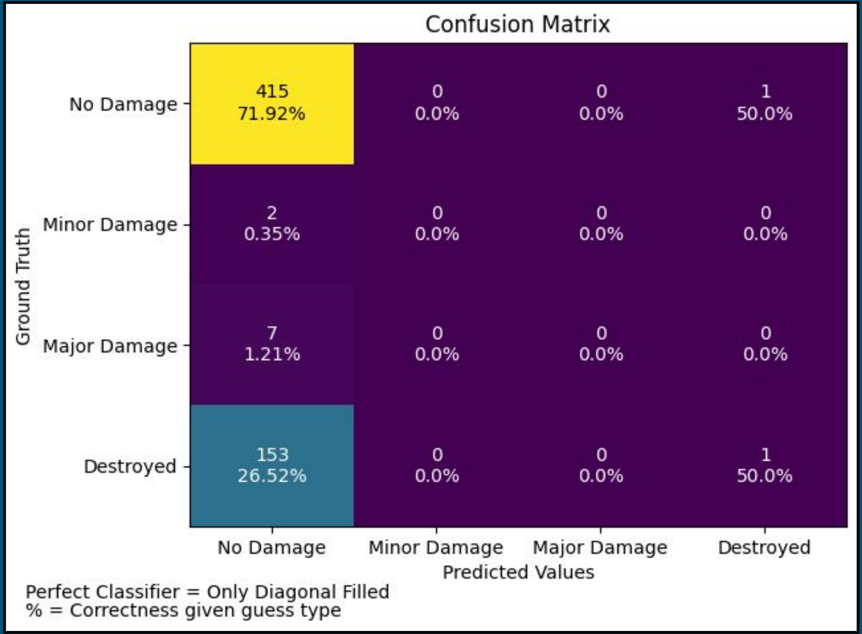
- Disaster type impacted results
- Applying pre-processing filters aided in success
- Skewed categorical data, most data is no-damage
- Traditional Classifiers performed effectively

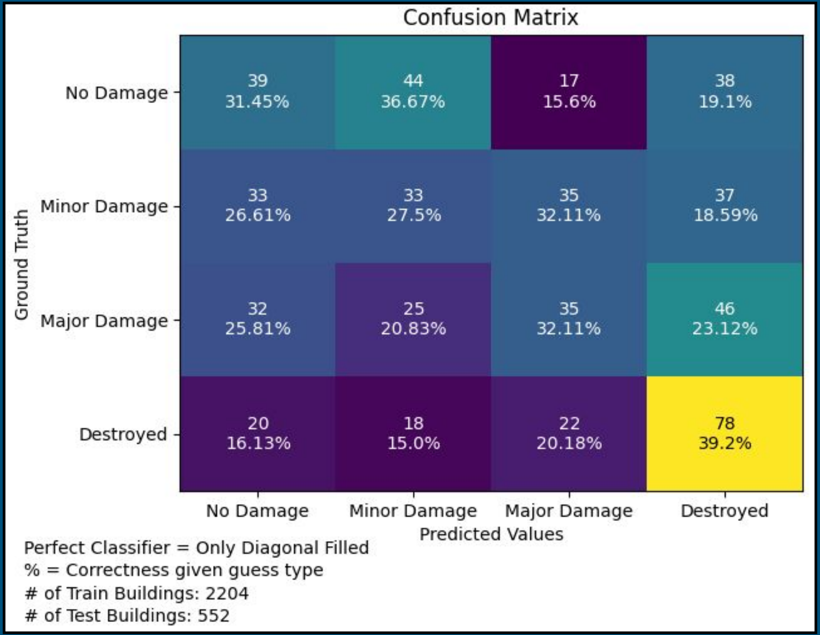
Accuracies	CNN	CNN with Edge Detection	Siamese NN	SVM	KNN Classifier
Hurricane Matthew	55.97%	33.24%	58.30%	54.40%	57.8%
Guatemala Volcano	91.94%	91.79%	32.00%	92.81%	96.89%
Palu Tsunami	83.48%	40.24%	77.5%	62.30%	86.23%
SoCal Fire	84.95%	66.00%	42.33%	82.13%	76.98%

CNN Architecture



CNN - Results and Takeaways





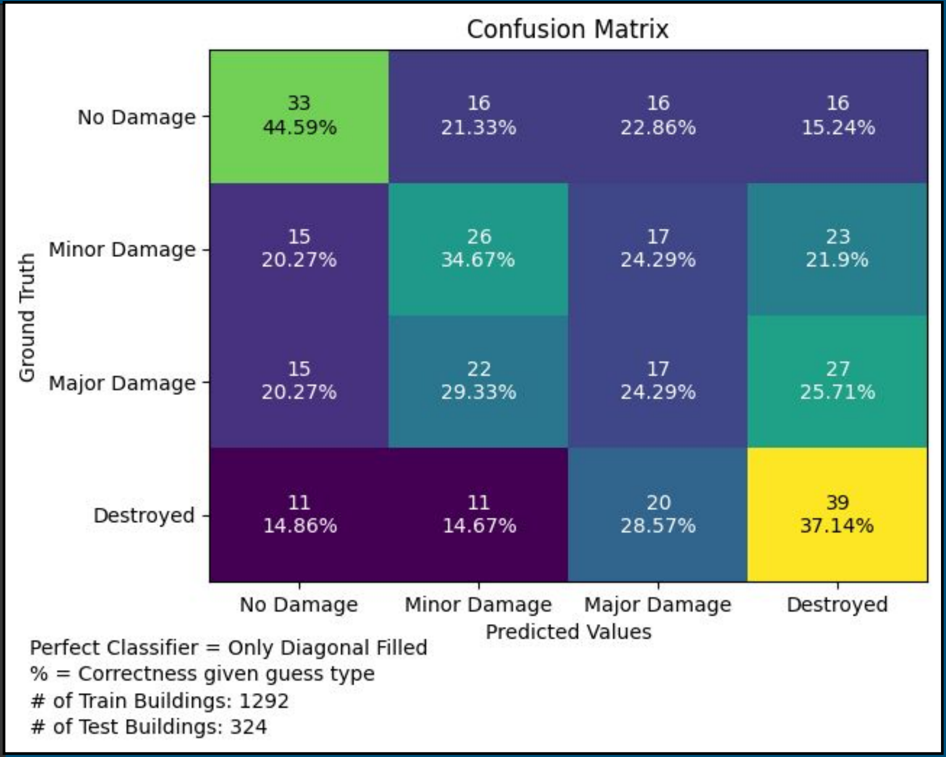
Left: Southern California Fire with larger 7x7 and 5x5 kernel sizes

Right: Hurricane Matthew equal distribution of building damage levels

Although the right model's accuracy is lower, accuracy doesn't always tell the full story

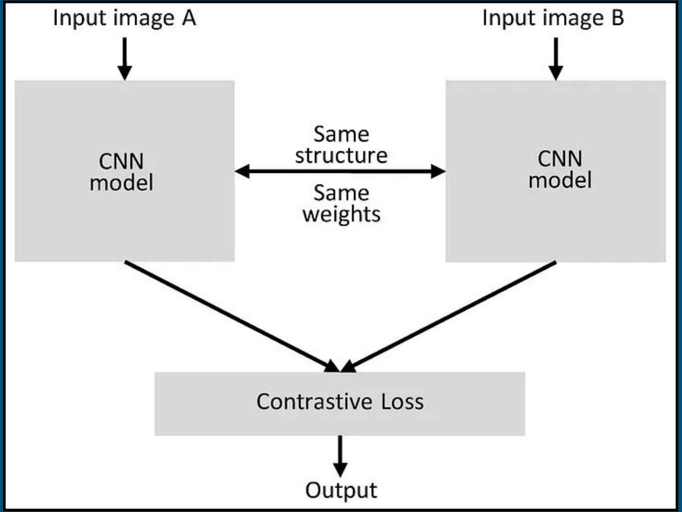
CNN + Edge Detection - Results and Takeaways

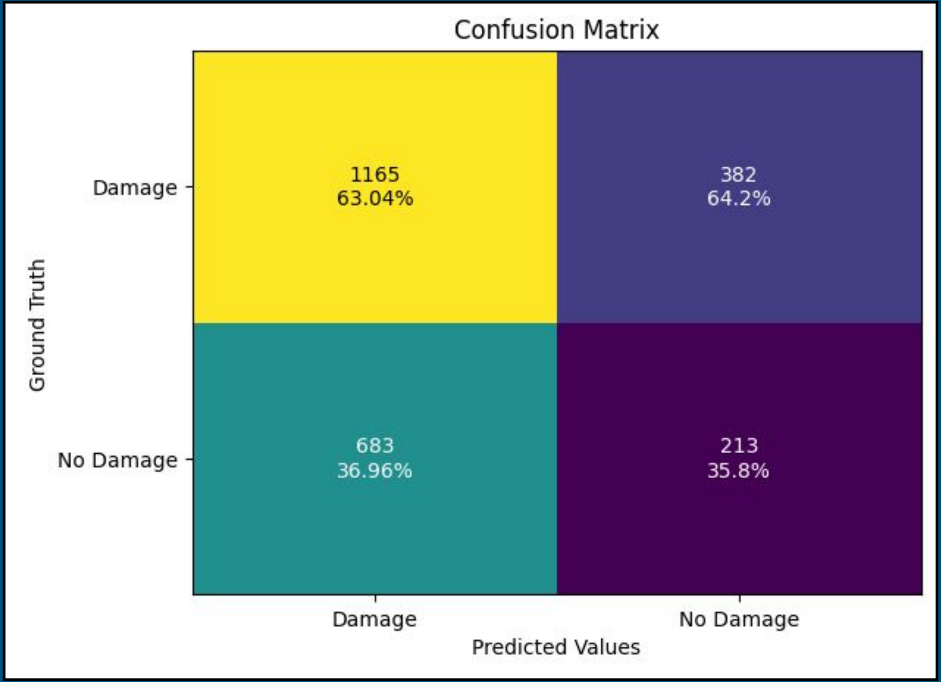
- Lower accuracy but much better guessing pattern
- We found Canny detection more promising than Sobel
- Other filters may be very useful to experiment with in future work



Siamese Network - Results and Takeaways

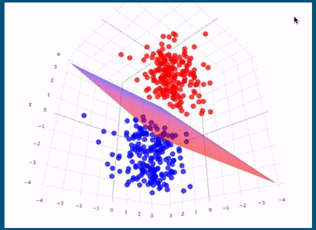
- Binary Classification
  - No-Damage vs Damage
- Complex Model
  - Tuning and adjusting was very difficult and time consuming

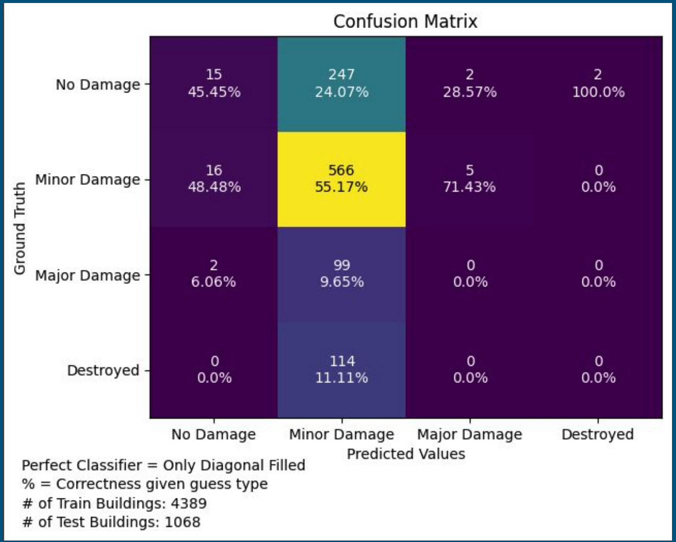




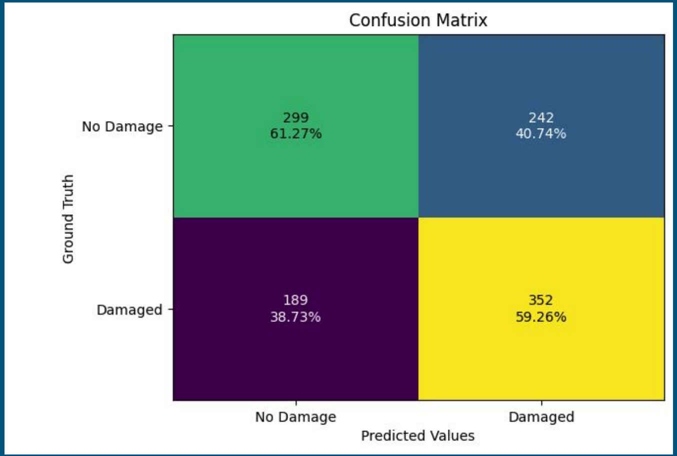
SVM - Results and Takeaways

- PCA
- RBF vs Polynomial Kernels
- Different types of training + testing
    - Baseline
    - Equal Class Split
    - Binary Classification
    - Train on all hurricanes/fires/earthquakes
- Average F1 score of 61%





Mainly guessing the majority class: 54.40%



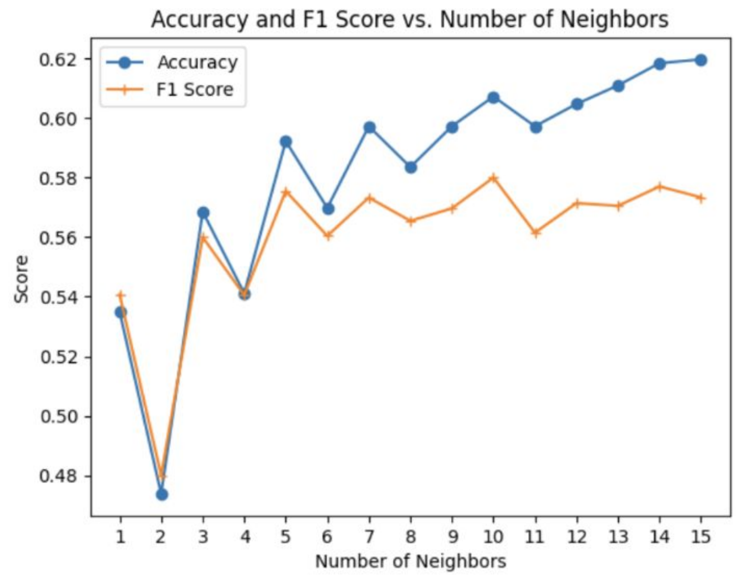
Trained + tested on all hurricanes: 60.17%

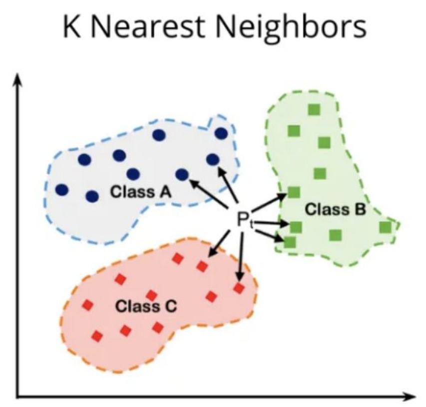
KNN - Results and Takeaways

- Equal Distribution by Damage Label
- Accuracy increases as # neighbors does
- Our F1 score approaches IBM maximum F1 on xView2 dataset

Metric	Score
Localization F_1	0.81
Damage Classification F_1	0.66
Total F_1	0.71

Source: (Alstad, 2020)





(Hurricane Matthew Disaster)

Realistic Constraints

Public Health:
- Disaster Recovery Organizations
- First Responders
- Post-disaster health

Environmental:
- Disaster Types
- Varying Damage
- Location

Global:
- Unequal location distribution
- Developed vs. developing countries

Fairness:
- Proportional Damage Variety
- Infrastructure differences

Economic:
- Disaster preparedness
- Recovery capability
- Damage is relative term by location

Images Sourced from Google Images

Conclusions

- Comparative Analysis of Classification and Traditional Classifiers
    - Accuracy, F1 scores, and Confusion Matrices
- Final Design Choice: KNN!
- Equal Building Distribution was beneficial
- Lessons Learned:
    - Organization
    - Communication
    - Preparation
    - Flexibility
- Further Work

Final Demo

References

General References:

Alstad, C. (2020, April 1). The xView2 AI Challenge. IBM Blog. https://www.ibm.com/blog/the-xview2-ai-challenge/.

Gerard, S., Borne-Pons, P., & Sullivan, J. (2024). A simple, strong baseline for building damage detection on the XBD dataset. *Arxiv*. https://arxiv.org/pdf/2401.17271

Gupta, R., Goodman, B., Patel, N., Hosfelt, R., Sajeev, S., Heim, E. T., Doshi, J., Lucas, K., Choset, H., & Gaston, M. E. (2019). Creating xBD: A Dataset for Assessing Building Damage from Satellite Imagery. *Arxiv*, 10–17. https://doi.org/10.1184/r1/8135576.v1

Hosfelt, R., & Gupta, R. (2019). xView2_baseline [Computer software]. DIUx-xView. https://github.com/DIUx-xView/xView2_baseline

May, S., Dupuis, A., Lagrange, A., De Vieilleville, F., & Fernandez-Martin, C. (2022). BUILDING DAMAGE ASSESSMENT WITH DEEP LEARNING. *˜the œInternational Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences/International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIII-B3-2022*, 1133–1138. https://doi.org/10.5194/isprs-archives-xliii-b3-2022-1133-2022

xView2. (n.d.). Computer vision for building damage assessment. xView2: Assess building damage. Retrieved from https://xview2.org/

References for Preprocessing:

Cropping Concave polygon from Image using Opencv python. (n.d.). Stack Overflow. https://stackoverflow.com/questions/48301186/cropping-concave-polygon-from-image-using-opencv-python

GeeksforGeeks. (2023, January 18). *Draw a filled polygon using the OpenCV function fillPoly()*. GeeksforGeeks. https://www.geeksforgeeks.org/draw-a-filled-polygon-using-the-opencv-function-fillpoly

Lezwon. (2019, December 31). *XView2 Challenge*. Kaggle. https://www.kaggle.com/code/lezwon/xview2-challenge#kln-47

References for Confusion Matrix Design

matplotlib.pyplot.pcolormesh — Matplotlib 3.8.4 documentation. (n.d.). https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.pcolormesh.html

Creating annotated heatmaps — Matplotlib 3.8.4 documentation. (n.d.). https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html#sphx-glr-gallery-images-contours-and-fields-image-annotated-heatmap-py

Placing text boxes — Matplotlib 3.8.4 documentation. (n.d.). https://matplotlib.org/stable/gallery/text_labels_and_annotations/placing_text_boxes.html

References

References for Siamese Modeling

Dutt, A. (2022, August 1). Siamese Networks Introduction and Implementation - towards Data Science. *Medium*. https://towardsdatascience.com/siamese-networks-introduction-and-implementation-2140e3443dee

Keras accuracy does not change. (n.d.). Stack Overflow. https://stackoverflow.com/questions/37213388/keras-accuracy-does-not-change

Rosebrock, A. (2021, April 17). *Building image pairs for siamese networks with Python - PyImageSearch*. PyImageSearch. https://pyimagesearch.com/2020/11/23/building-image-pairs-for-siamese-networks-with-python/

Rosebrock, A. (2024, May 8). *Siamese network with Keras, TensorFlow, and Deep Learning - PyImageSearch*. PyImageSearch. https://pyimagesearch.com/2020/11/30/siamese-networks-with-keras-tensorflow-and-deep-learning/

Sklearn.utils.class_weight.compute_class_weight. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

References for Edge Detection CNN implementation

Image Kernels explained visually. (n.d.). Explained Visually. https://setosa.io/ev/image-kernels/

Team, L., & Team, L. (2024, February 5). *Edge Detection using OpenCV | LearnOpenCV #*. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow With Code, & Tutorials. https://learnopencv.com/edge-detection-using-opencv/

References for SVM and KNN implementations

Martin, E. (2024, March 18). *Multiclass Classification Using Support Vector Machines*. Baeldung. https://www.baeldung.com/cs/svm-multiclass-classification#:~:text=Multiclass%20Classification-,In%20this%20type%2C%20the%20machine%20should%20classify%20an%20instance%20as,dog%20breed%20in%20an%20image

References

References for Non-Original Images:

Figure 3. Illustration of linear SVM Classifier separating the two. . . (n.d.). ResearchGate. https://www.researchgate.net/figure/Illustration-of-linear-SVM-Classifier-separating-the-two-classes-Illustration-of-linear_fig1_359803757

Sachinsoni. (2023, June 11). K Nearest neighbours — Introduction to machine learning algorithms. *Medium*. https://medium.com/@sachinsoni600517/k-nearest-neighbours-introduction-to-machine-learning-algorithms-9dbc9d9fb3b2

Singh, P. (2021, December 11). Siamese Network Keras for Image and Text similarity. *Medium*. https://medium.com/@prabhnoor0212/siamese-network-keras-31a3a8f37d04

University of Maryland engineering seal - Bing. (n.d.). Bing. https://www.bing.com/images/search?view=detailV2&

Vishaltiwari. (n.d.). *GitHub - vishaltiwari/Building-Extraction: Extracting Buildings from Aerial Images*. GitHub. https://github.com/vishaltiwari/Building-Extraction

Thanks

Any Questions?