Horizon SAGE

Exploring an Adaptive Greedy Approach to Skyline Detection using Neural Networks

Motivation

Mountain identification apps do a poor job of aligning generated mountain lines with the visible horizon.

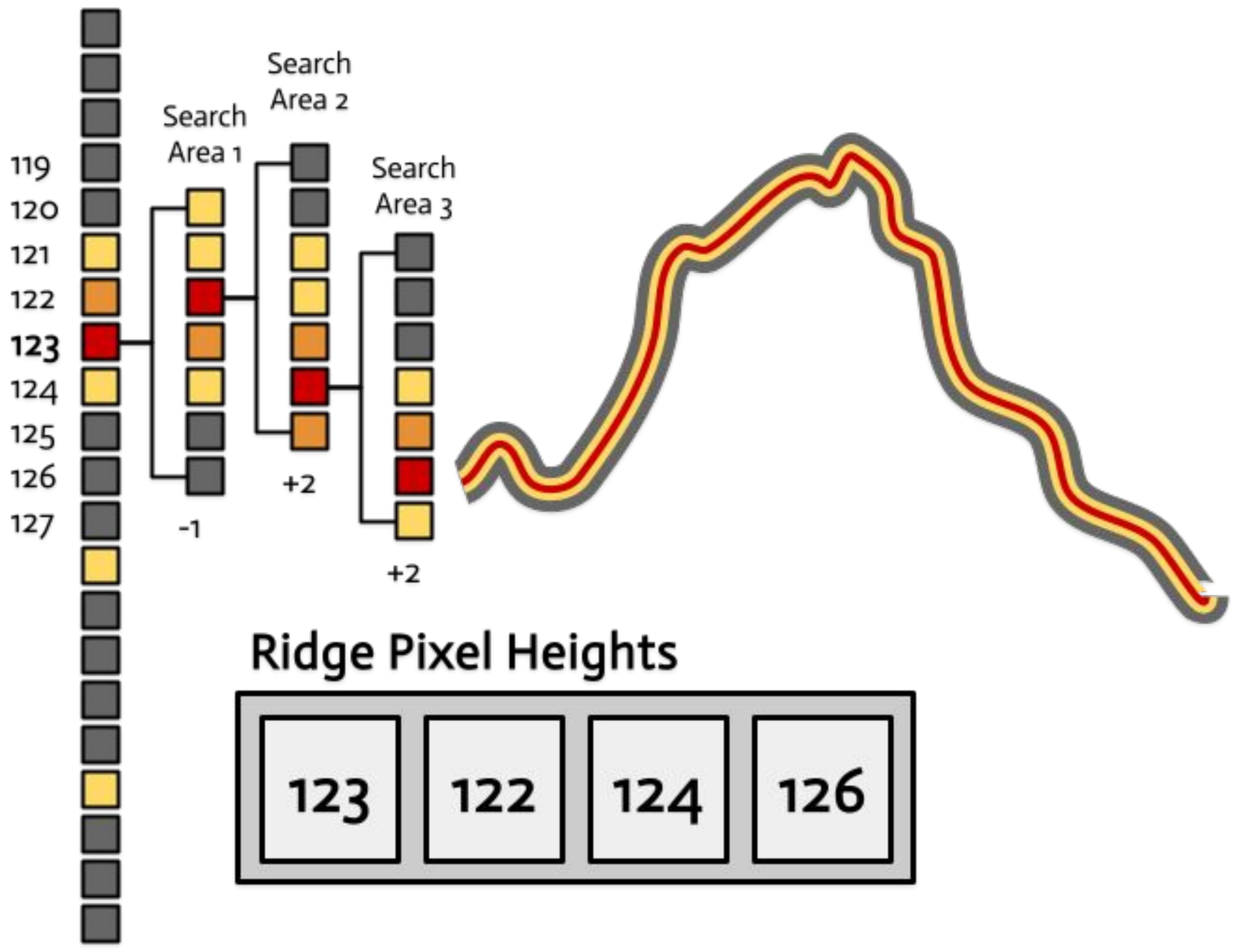
Implementing horizon detection has been explored as a viable means of improving in-app alignment.[1]

Existing skyline detection algorithms are slow because they:

- Classify a large amount of pixels per image
- Reconstruct the ridgeline using complex dynamic programming algorithms

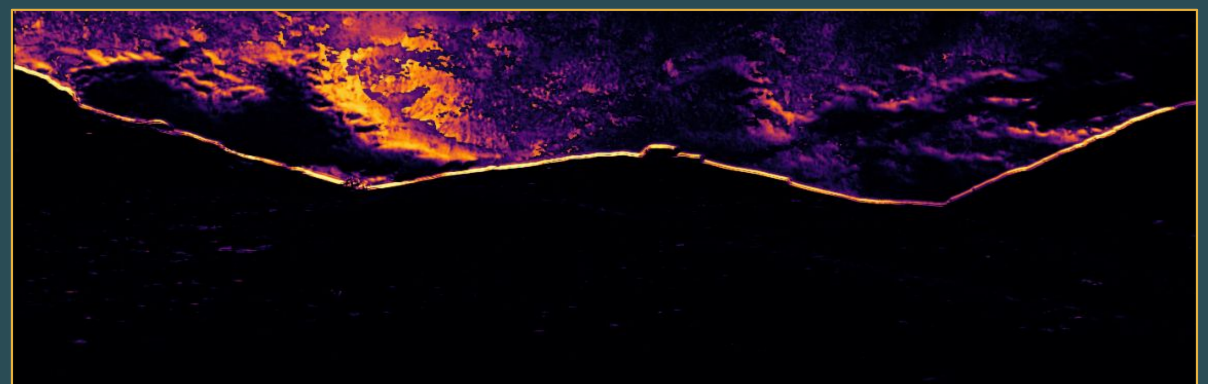
Methods

We classify a small region of pixels in each image column and add the most likely candidate to the reconstructed skyline as we go.

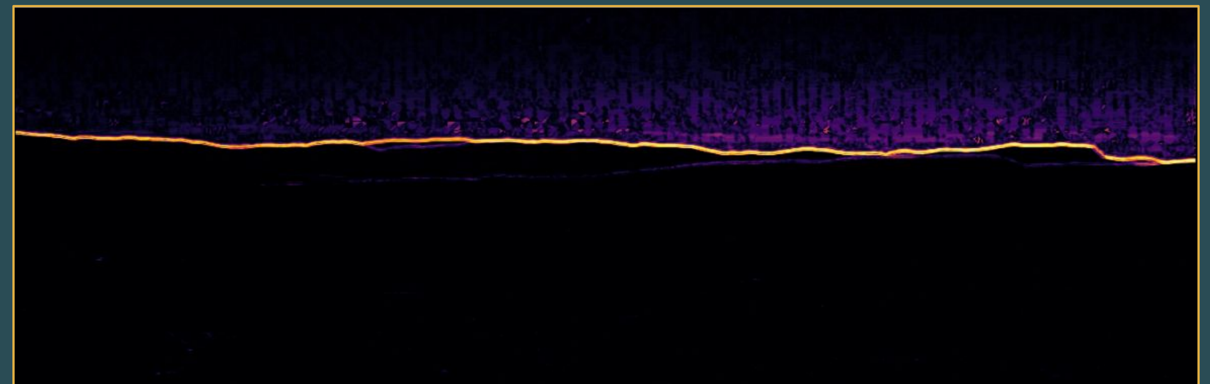


If no pixels in the current search window are likely to be in the ridgeline, we restart the search from the next most probable point. **This avoids spending time classifying unnecessary pixels and removes the need for dynamic programming.**

We developed a **fast** and **accurate adaptive greedy algorithm** that extracts **mountain ridgelines** from images

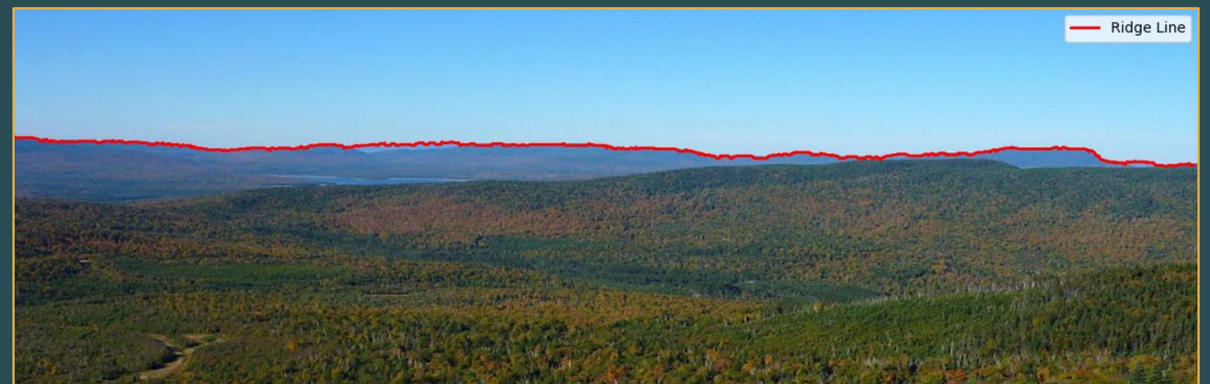


Probability Map



Probability Map

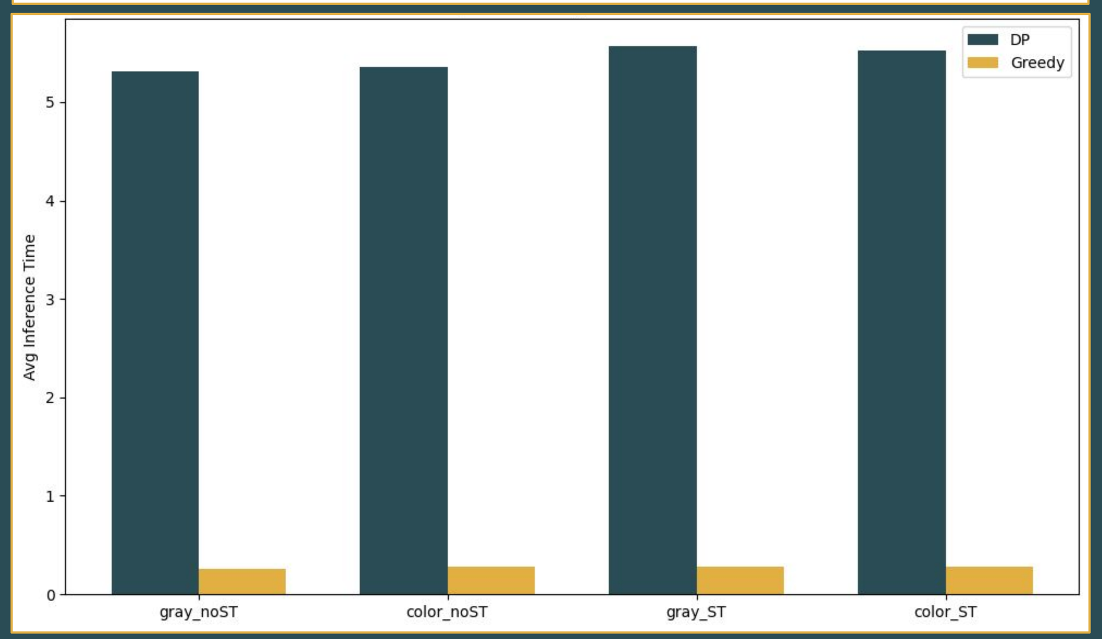
Reconstructed Skyline



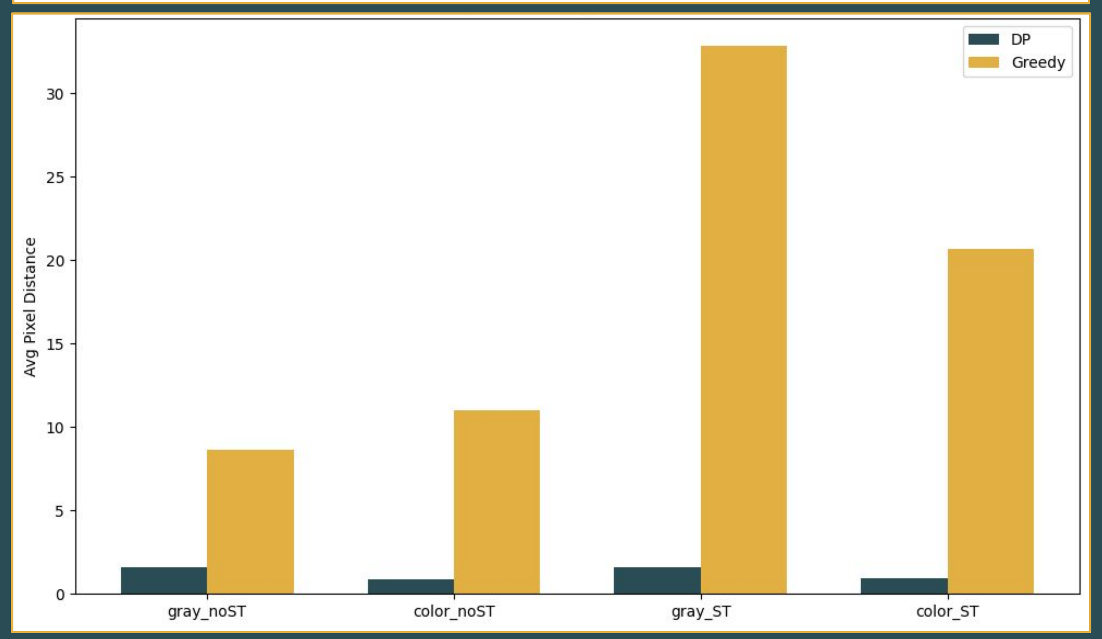
Reconstructed Skyline

Greedy Inference Outpaces Dynamic Programming at a Modest Accuracy Cost

Average Inference Time



Average Pixel Distance

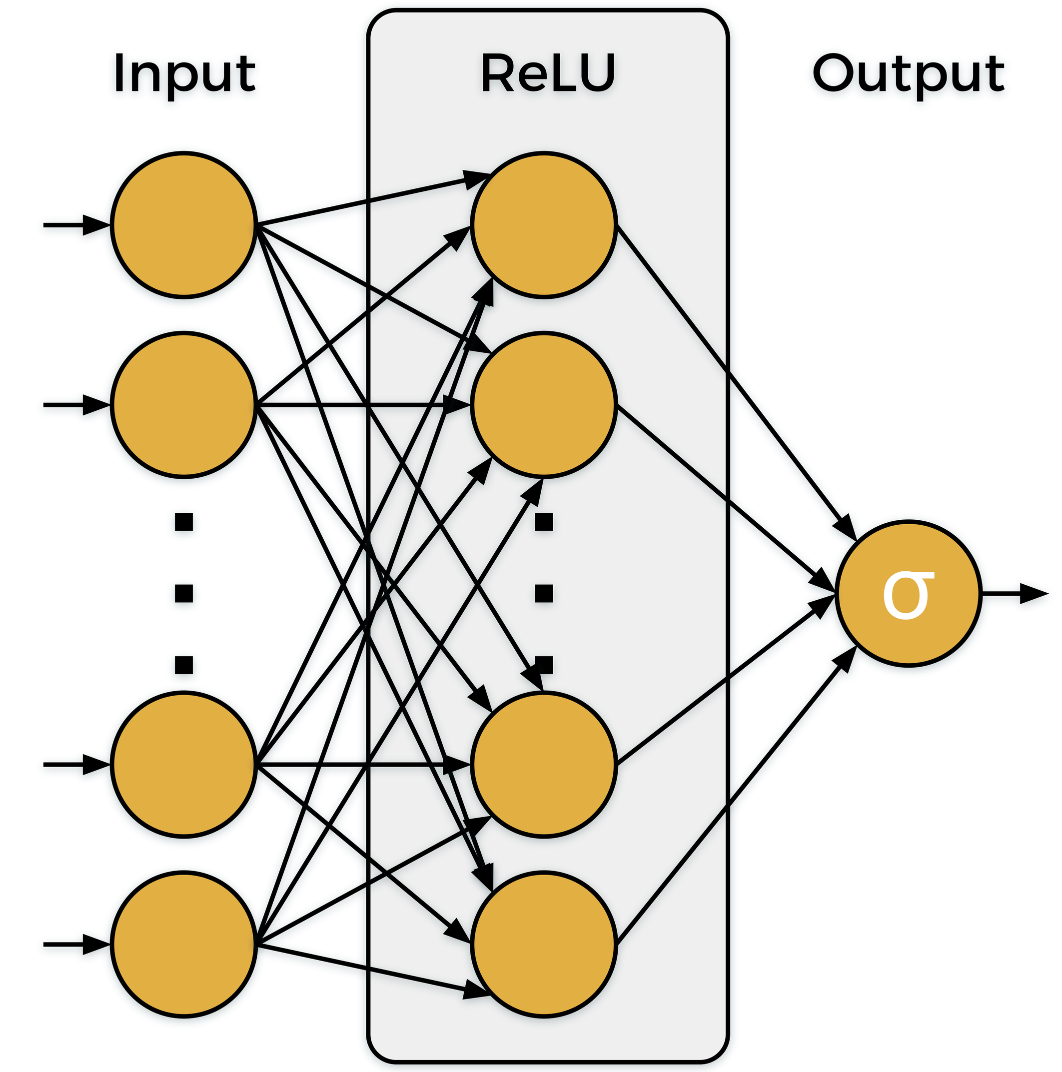


Special thanks to Professor Philip Caplan and Professor Andrea Vaccari for their guidance!

GitHub

[1] Porzi, L., et al. An automatic image-to-DEM alignment approach for annotating mountains pictures on a smartphone. *Machine Vision and Applications* 28, (2017).
[2] Ahmad, T., et al. Resource Efficient Mountainous Skyline Extraction using Shallow Learning. In the *International Joint Conference on Neural Networks* (IJCNN), (2021).
[3] Getreuer, P., et al. BLADE: Filter learning for general purpose computational photography. *2018 IEEE International Conference on Computational Photography* (ICCP), (2018).

We employed classic shallow neural network for binary classification.



This lightweight architecture is expressive enough for skyline pixel classification. Finding the balance between accuracy and speed is crucial for applications on resource-constrained devices like mobile phones.[2]

We explored two feature-based training approaches.

Each pixel is classified using its *7 x 7* pixel neighborhood.

- **Original:** trained on raw pixel intensities
- **ST:** trained on **structure tensor** features and raw pixel intensities

We also compared training on **grayscale** versus **color images**.

Structure tensors summarize the predominant **orientation**, **strength**, and **coherence** of local pixel neighborhoods based on image gradients.[3]

Conclusion

Our algorithm demonstrates high speed and decent accuracy.

Nevertheless, its effectiveness is tightly linked to the accuracy of the underlying classifier. More training data and a deeper model could significantly improve performance. We hope to build on this foundation in the future and eventually develop a fully operational real-time mountain detection application.

Aidan McMillan | 25
Lukka Wolff | 25.5

CS701 Senior Seminar | Philip Caplan | Spring 2025