Efficient Wildfire Detection for use in Drones

Nicholas Kroetsch, Matthew French, Charles Light

Background

- When uncontrolled, wildfires can cause loss of life and billions of dollars in damage
- Early detection is key to putting them out before they spread
- How to spot small smoke plumes in remote areas?
- Weather balloons and drones with cameras

Problem

- Drones offer a power-constrained environment for machine learning
- Modern CNN-based computer vision algorithms are very computationally expensive
- How can we decrease the power consumption of fire detection computer vision algorithms without sacrificing accuracy?

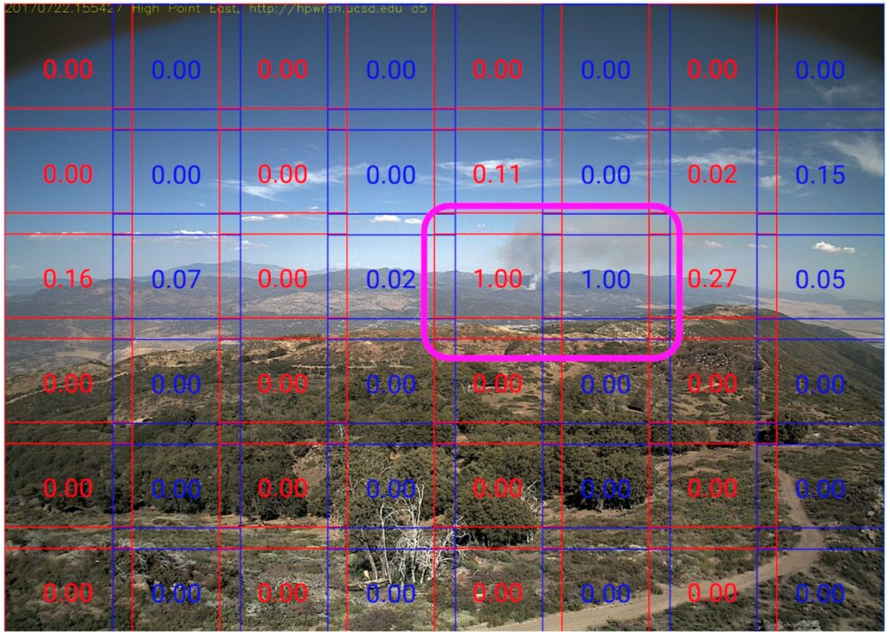
Prior Work

- Wildfire detection with machine learning has been deployed in real systems to detect smoke increasingly well (recent work can notice faraway plumes in only minutes), although prior work in this area has been on more conventional machine learning processing platforms without significant energy constraints.
- Digital foveation and related concepts focus on running expensive machine learning algorithms on only the most important portions of input data. Power optimization in general is an active area of machine learning research.
- The novel contribution involved focuses on the combination of these two research areas.

Lubana and Dick (2018): https://robertdick.org/publications/lubana18nov.pdf
Feng et al. (2019): https://www.sciencedirect.com/science/article/pii/S0167926019301762
Toulouse et al. (2017): https://hal.archives-ouvertes.fr/hal-01560570/file/PrePrintFSJ.pdf
Toreyin et al (2008): https://ieeexplore.ieee.org/document/4632677

Firecam

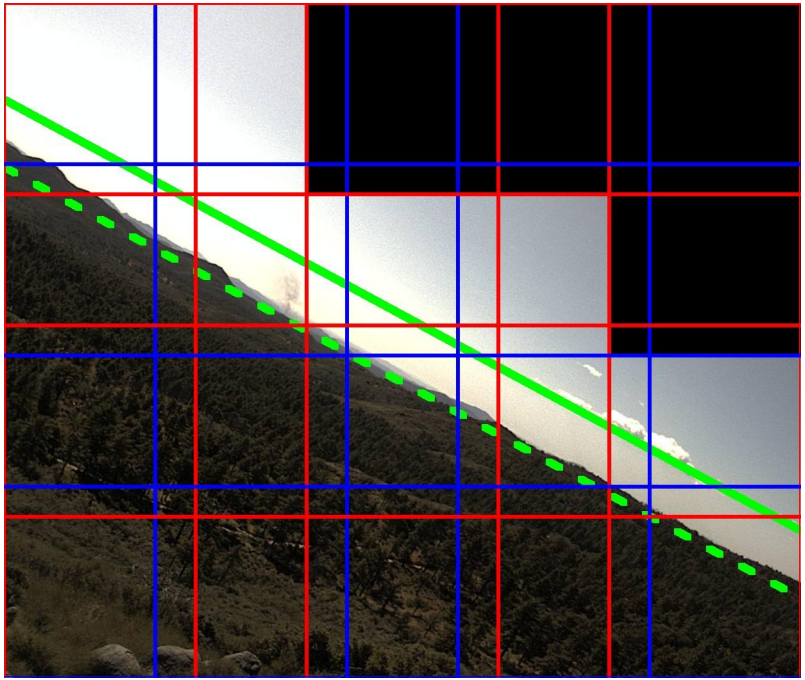
- Prior work by Govil et al. on Firecam forms the basis of our work
  - Provided dataset of wildfire smoke images
  - Provided trained Inceptionv3 CNN model
- Segmentation approach
  - Each image is split into segments
  - Each segment is input to the CNN
  - Segments are classified as smoke or nonsmoke



Govil et al. (2020): https://www.mdpi.com/2072-4292/12/1/166/htm

Horizon Detection Overview

1. Horizon guess separates pixels into two groups (sky + ground)
2. Criterion depends on how alike the two groups are to each other and how different the groups are
3. Horizon corrected up to account for smoke near horizon
4. Segments completely above the horizon are excluded



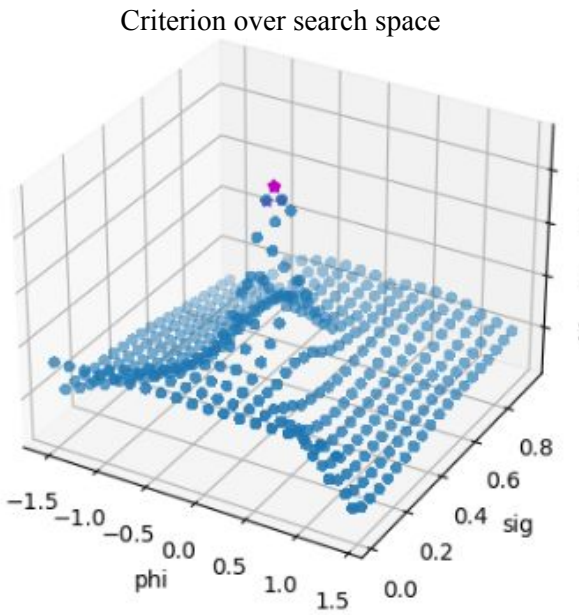
Horizon Detection Details

Criterion

$$J = \frac{1}{|\Sigma_s| + |\Sigma_g| + (\lambda_1^s + \lambda_2^s + \lambda_3^s)^2 + (\lambda_1^g + \lambda_2^g + \lambda_3^g)^2}$$

Parameterization of Horizon

$$\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \quad \sigma \in [0\%, 100\%]$$



Ettinger et al. (2002): http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.6951

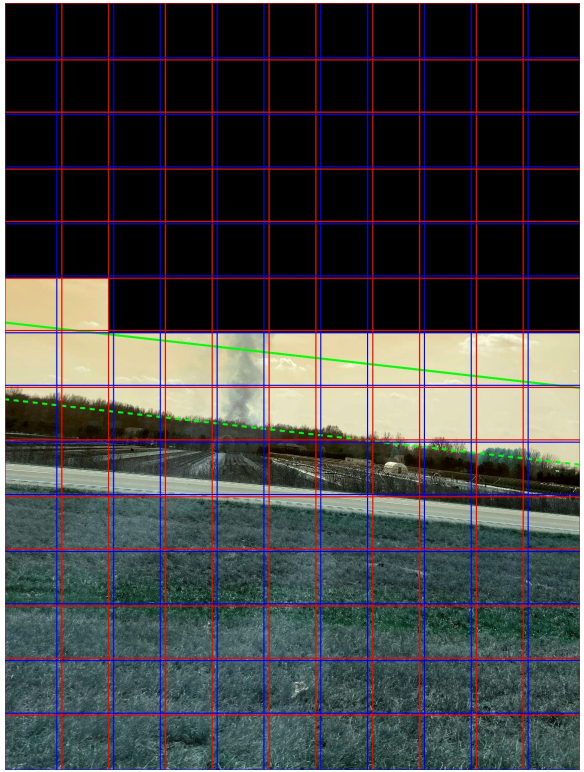
Horizon Detection Results

On 1000 images,

- Without horizon detection:
  - 5002.42 seconds
  - 958/1000 (95.8%) accuracy
- With horizon detection
  - 3135.27 seconds
    - 42.55 seconds on detecting horizon (1.36% of total time)
    - 3092.71 seconds running the model (98.64% of total time)
  - 945/1000 (94.5%) accuracy

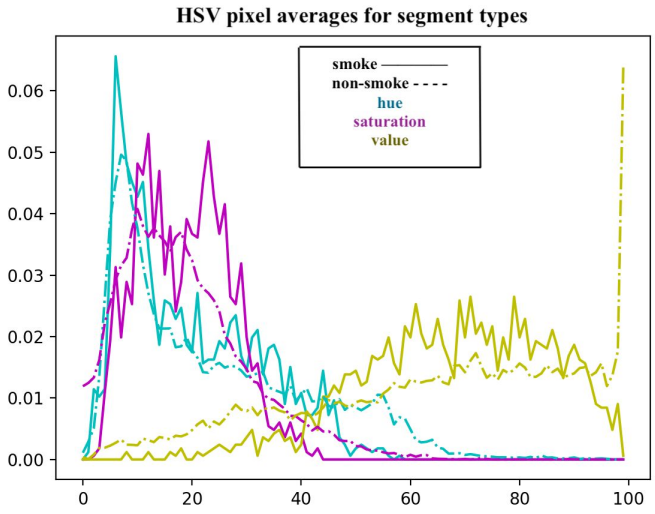
This results in a **37.32% reduction in time spent** (~35% of segments removed) while only **losing 1.3% accuracy** relative to the model.

Bonus



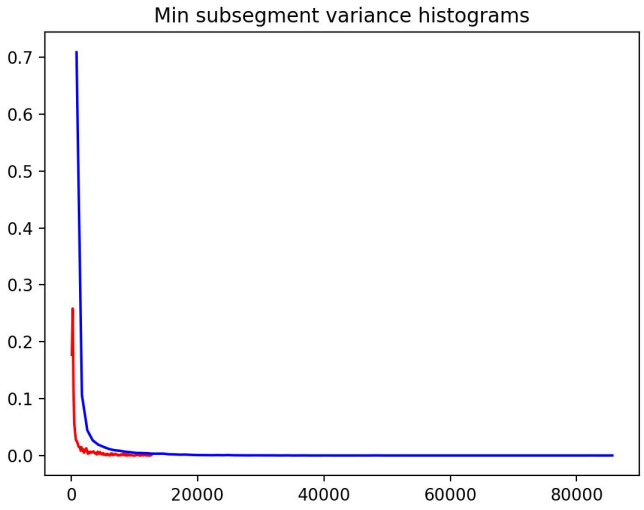
Color Histogram Overview

- We looked into color differences between smoke and non-smoke segments.
- This examination included a viewing of separate histograms for RGB channels, along with average contributions from each channel, although saturation was seen as the most useful feature to test segment types for.
- We measured saturation by variance of the three RGB color channels at each pixel and took the average.



Color Histogram (results)

- Average colors for segment sizes the network runs on differ only slightly by segment type.
- This can’t remove enough segments to be useful (only 2% when trying to avoid all false negatives) due to a skewed distribution in saturation levels.
- We also tried edge detection in the S channel of HSV space, but smoke doesn’t stand out there in comparison to other objects in an image



Conclusions/Future Work

- Horizon detection was successful in reducing the amount of computation required without sacrificing much accuracy
- In this dataset, it is difficult to find a quantitative difference based on color between smoke and non-smoke segments
  - Bigger or closer smoke plumes could make this possible
- Other recent methods for reducing memory/computation of large CNNs such as changing data type and pruning