Satellite-Based Wildfire Perimeter Construction Using FIRMS Detections

Jonathan Kelly, Michael Michelini

Introduction

Wildfires have become an increasingly prominent struggle in California over the last decade, with several fires causing massive devastation to both homeowners and the environment. Effective wildfire management depends heavily on timely response and accurate information regarding current conditions. Currently, there is a lack of real-time automated reporting for wildfire perimeters. This type of information could help first responders and the general public understand the risks an active wildfire may pose to them. To help address this, our project aims to develop an algorithm that estimates wildfire perimeter progression through the lifecycle of a fire. We leverage satellite detections and geospatial analysis techniques to construct these perimeters. This report outlines our algorithm design, data processing pipeline, and evaluation using historical fires. By testing on historical data, we establish a foundation before adapting the system to real-time prediction during active wildfires, where it would be most beneficial.

Related Work

Research has explored using satellite data to estimate fire perimeters. These studies have shown that clustering NASA FIRMS satellite detections can produce reasonable boundary estimates. Briones-Herrera et al. (2020) aggregated FIRMS detections to estimate final burned area at monthly intervals for fires in Mexico. Chen et al. (2022) developed an object-based tracking system that monitors all fire activity across California, including agricultural burns and other non-wildfire events, using alpha shape for polygon construction. Bhuian et al. (2024) compared different polygon construction methods including buffer, concave, convex, and combinations to evaluate their accuracy in delineating fire perimeters.

Our project differs from these approaches by focusing specifically on building a system for individual wildfire tracking and progression. Rather than estimating monthly burned area or monitoring all fire activity statewide, we aim to construct perimeters for individual fire events and track their growth through observation windows. We utilize recent California wildfires and FIRMS detections from 2021 to early 2025, which include new satellite configurations that were not available during these prior studies. This allows us to explore how different methods perform with more frequent detection coverage on the individual fire level.

Data Sources

Our analysis relied on two primary data sources from January 2021 through February 2025 that together describe the progression and final perimeter of historical wildfires.

1. **NASA FIRMS**: Provides near real-time satellite detections of thermal anomalies. Detections are reported as point observations with latitude, longitude, timestamp, and confidence level. We used the VIIRS instrument family which reports at 375-meter resolution, allowing us to trace the finer details of evolving fire perimeters. Between 2021 and 2023, Suomi NPP and NOAA-20 satellites provided detections. In 2024, NASA added NOAA-21 as a third VIIRS satellite to increase

coverage (NASA, n.d.). These detections serve as the foundation for clustering and constructing our fire perimeter polygons.

2. **CAL FIRE Fire Perimeters**: Contains geospatial polygons representing the mapped boundaries of historical California wildfires. Published perimeters reflect the best available mapping compiled from several government agencies including the Bureau of Land Management and United States Forest Service (CAL FIRE, n.d.). Each perimeter serves as the ground truth reference for evaluating our predicted perimeters.

Data Limitations

To evaluate our algorithm for every window of a fire, we need published and verified fire perimeters for the full progression. Currently there are no available sources of data we could find representing this fire perimeter progression for the fires in our dataset. CAL FIRE along with other agencies only post final fire perimeters that do not include any intermediary steps of the fire. Therefore, this prohibits us from evaluating the algorithm at intermediate steps and limits us only to comparison with the final CAL FIRE perimeters.

Additionally, FIRMS detections can be affected by environmental conditions. Cloud cover and smoke can obstruct satellite observations, leading to gaps in detection coverage during certain passes. This may result in missed detections for fast-moving fires during periods of poor visibility, potentially causing the algorithm to underestimate fire spread.

Data Preprocessing Pipeline

Since our goal is to develop a system that could eventually support real-time perimeter tracking on active fires, we designed the preprocessing pipeline to best mimic these conditions. This allows us to make our evaluation more representative of how it might perform with active fires. The pipeline consists of five stages, implemented in data_preprocessing.py.

1. **Load and Standardize raw FIRMS and CAL FIRE Datasets**
   Our pipeline starts by loading CAL FIRE's wildfire perimeter dataset along with FIRMS VIIRS detections. We standardize both sources so they share a consistent set of identifiers, coordinates, and timestamps.
2. **Link FIRMS Detections to CAL FIRE Events**
   To link FIRMS detections to individual fire events, we apply a 5 km buffer around each CAL FIRE final perimeter. We then use a spatial join to associate any FIRMS points falling inside this area with that fire event. This wider region better reflects what active fire conditions would be, where no final perimeter exists. In this scenario, the system must look beyond its current predicted boundary to capture new activity.
3. **Cross Fire Filtering**
   The increased buffer can include detections that actually belong to other fires in our dataset. To clean each fire's detection set, we first identify any nearby fires that were active within 10 km during the same time period. For each such pair of overlapping fires, we compare every FIRMS detection to both perimeters and remove it from the current fire if it lies closer to the neighboring fire. The buffer may also capture heat signatures from small unmapped fires or agricultural burns. Since we cannot reliably determine whether these represent spot fires or unrelated activity, we keep them in our dataset.
4. **Observation Windows**
   VIIRS satellites pass over the same area about an hour apart and make two passes per day. This causes FIRMS detections to come in short bursts with long gaps between them. To turn these bursts into clearer observation steps, we group detections into two-hour windows. This grouping

accommodates both the two-satellite years (2021-2023) and the three-satellite years (2024-2025), keeping the observation structure consistent across our dataset. Using these windows gives us a simple, ordered set of observation steps to build perimeters from.

5. **Final Fire Selection Criteria**
   To ensure we only evaluate fires with enough detections and window observations to resemble meaningful perimeter construction and tracking, we apply three simple filters. A fire is kept only if it has:
   - At least four observation windows
   - At least 150 FIRMS detections
   - A mapped area of at least 1 km² in CAL FIRE's final perimeter

After these stages, the pipeline outputs firms_filtered.parquet, containing the cleaned and windowed FIRMS detections, and calfire_filtered.parquet, containing the matched CAL FIRE perimeters. These files serve as the input for our perimeter construction algorithm.

Perimeter Construction Methodology

Our perimeter construction, implemented in perimeter_pipeline.py, converts FIRMS detections into predicted fire perimeters through five main stages: accumulating window detections, clustering detections into groups, merging nearby clusters, filtering low-density outliers, and constructing polygon shapes.

Window Detection Accumulation
Each individual window provides a snapshot of the fire perimeter as it progresses. To approximate the full progression of the fire, we accumulated all detections across windows over time. By taking the accumulation of the current and previous windows we are then able to construct the perimeter for the current time and gradually build the full fire footprint.

Clustering With DBSCAN
To separate valid fire detections from noise, we applied DBSCAN, a density-based clustering algorithm. DBSCAN groups nearby points into clusters based on two parameters:

1. **Eps**: Determines the maximum distance between two points to be considered part of the same cluster
2. **min_samples**: Determines the minimum number of points required to be considered a cluster.

Points not meeting these criteria are labeled as noise and filtered out. By identifying multiple distinct clusters, the algorithm can detect when a fire has separate fronts or disconnected perimeters rather than assuming all detections belong to a single continuous area.

Temporal Merging
DBSCAN often produces multiple separate clusters for what is actually a single fire front. Since satellite passes are limited to twice daily, we miss the continuous spread that connects these areas. To account for this, we assume clusters within two kilometers of each other represent connected fire growth and merge them into a single polygon. Clusters beyond this distance remain separate, representing distinct fire fronts within the same fire event.

Density Filter
We also evaluated an optional density filter applied before polygon construction. In certain satellite passes, small clusters of points appeared just outside the actual fire perimeter, likely due to geolocation error from environmental factors. When these are included in the final clustering, they expand our predicted perimeter and overestimate the burned area. To address this, we used the BallTree algorithm which measures the number of points within a specific radius. Points falling under a specified density threshold are filtered out, as areas with fewer accumulated detections give us less confidence in their accuracy. This helps remove isolated edge points and reduce overestimation across fires.

Polygon Construction
Once clusters are identified, each one is converted into a realistic polygon shape. The two methods we explored were selected based on their ability to capture complex perimeter patterns. Other methods such as convex hulls produce mostly smooth perimeters that are not representative of typical fire perimeters.

1. **Concave Hulls**: Creates a polygon from a set of clustered points and allows shapes to follow inward bending concavities and complex patterns by adjusting the concave ratio parameter. Lower values of the ratio reflect tighter formed, complex patterns while larger values create smoother edges.
2. **Alpha Shapes**: Also creates polygons from clustered points, but uses a distance-based parameter instead of a ratio. Smaller alpha values create more detailed shapes while larger values produce smoother edges.

Evaluation Metrics
The final predicted perimeters were evaluated with the following complementary metrics that determine geometric accuracy and classification performance.

1. **F1.25 Score**: A weighted mean of precision and recall with a slight emphasis on recall. We chose this as our primary metric because failing to identify burned area was considered more detrimental than slightly overpredicting fire extent. The beta parameter controls this tradeoff, with values greater than 1 weighting recall higher and values less than 1 weighting precision higher. We used a beta of 1.25 to penalize overestimation while placing more emphasis on identifying burned areas.

$$F1.25 = \frac{(1+1.25^2)*(Precision*Recall)}{(1.25^2)*Precision+Recall}$$

2. **Intersection over Union (IoU)**: Measures geometric similarity between our predicted perimeter and the CAL FIRE perimeter. It is computed by dividing the overlapping area by the total area covered by both perimeters. Higher values indicate better overlap between predicted and actual perimeters.

$$Intersection\ over\ Union\ (IoU) = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

Parameter Tuning
To identify the best perimeter construction approach, we tested both concave hull and alpha shape methods across a range of parameters. Each method was evaluated using the same DBSCAN clustering and density filter settings, along with its own polygon-specific parameters. We restricted tuning to 2024 fires since this was the first full year with the three-satellite configuration, which better reflects how a real-time system would operate. The best parameters for each method were selected based on the mean F1.25 score across all 2024 fires.

Concave hull achieved consistently higher F1.25 scores compared to alpha shape, as shown in Table 1. Additionally, Figure 1 shows a common pattern we observed where alpha shape failed to connect internal concavities of a fire, even with optimal settings. The alpha shape results were too sensitive to its parameter value and did not generalize well across different fire shapes and sizes, while concave hull produced more robust boundaries. This led us to choose concave hull as our preferred construction method.

Method	F1.25 Mean with s.d.
Concave Hull	0.864 ± 0.095
Alpha Shape	0.847 ± 0.106

Table 1: Concave Hull vs Alpha Shape Performance

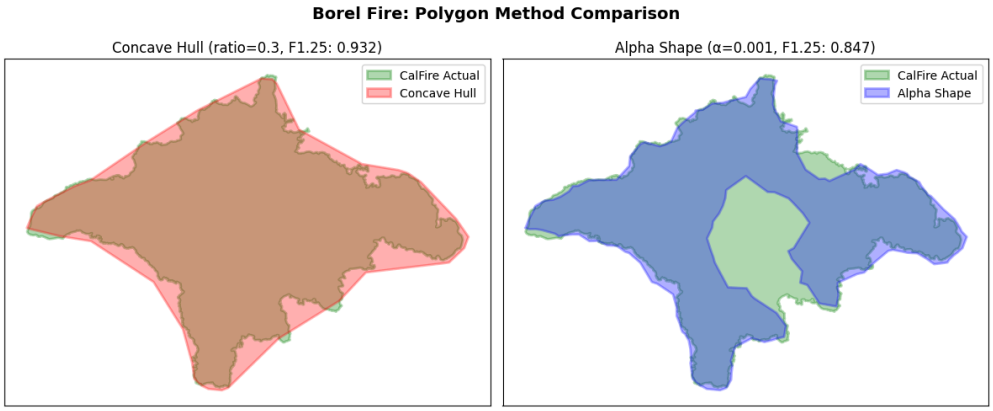

Figure 1: Concave Hull vs Alpha Shape Comparison

Evaluation

We evaluated our algorithm by comparing each fire's final predicted perimeter to the final perimeter mapped by CAL FIRE. As noted earlier, the absence of progression snapshots means this final-to-final comparison is the most consistent basis for assessing performance. Since the FIRMS detection system operated under two different satellite configurations across the fires in our dataset, we report performance separately for the two-satellite years (2021–2023), the three-satellite years (2024–2025), and all years combined. Table 2 reports the mean and standard deviation of F1.25 and IoU scores for each grouping. Figure 2 displays the distribution of F1.25 scores for the two satellite eras, illustrating the range and variability of construction performance.

Era	No. Samples	F1.25 with s.d.	IoU with s.d.
All (2021-2025)	93	0.870 ± 0.069	0.751 ± 0.093
2-Satellite (2021-2023)	60	0.873 ± 0.052	0.754 ± 0.083
3-Satellite (2024-2025)	33	0.864 ± 0.092	0.745 ± 0.109

Table 2: Evaluation Results by Satellite Era

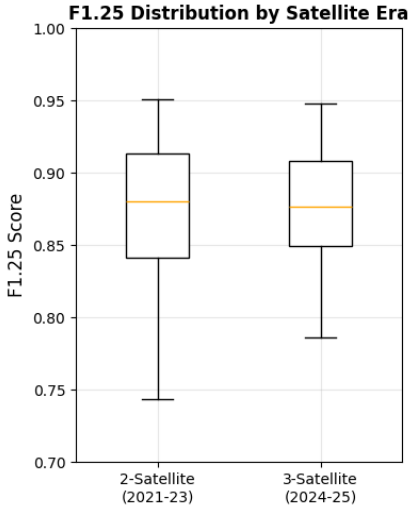

Figure 2: F1.25 Distribution by Satellite Era

Across all 93 fires, our algorithm achieved a mean F1.25 score of 0.870 and a mean IoU of 0.751. An IoU of 0.751 means that, on average, about 75% of the area covered by the true CAL FIRE perimeter overlaps with our reconstructed perimeter. The F1.25 score of 0.870 further reflects strong overall accuracy, balancing the algorithm's ability to identify burned areas (recall) with its ability to avoid over-predicting the fire's extent (precision).

Performance between the two satellite eras is similar. The two-satellite years show slightly higher mean scores (F1.25: 0.873, IoU: 0.754) compared to the three-satellite years (F1.25: 0.864, IoU: 0.745). However, the box plot shows that median performance is nearly identical, indicating that typical behavior is stable across eras. The lower mean in the three-satellite era is driven by a single outlier fire (F1.25: 0.41), which increases variability but does not represent normal performance.

Density Filter Effects

We examined whether the shift from the two-satellite to three-satellite era produced meaningful differences in FIRMS observations. To compare detection rates between eras fairly across different fire sizes, we normalized detections by fire area (km²) and duration (days). Figure 3 shows the distribution of detection rates for both eras, with the three-satellite era showing higher rates due to increased satellite coverage.

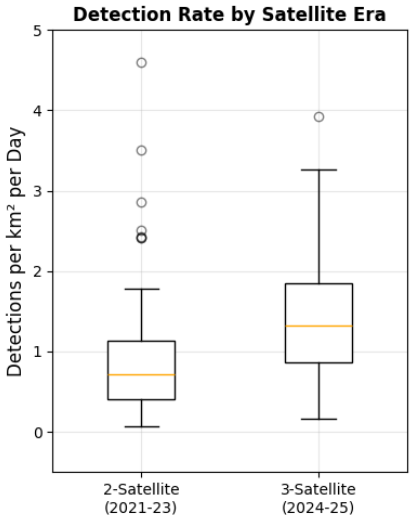

Figure 3: Detection Rate by Satellite Era

Next, to evaluate whether the additional satellite introduced more geolocation noise, we compared the change in our evaluation metrics when applying the density filter across both eras. Figures 4 and 5 show the before and after performance for each era. If the third satellite introduced significant noise, we would expect a larger improvement from the density filter in the three-satellite era.

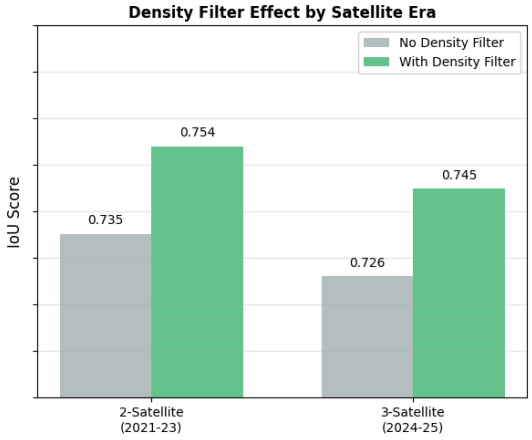

Figure 4: Density Filter Effect on IoU by Satellite Era

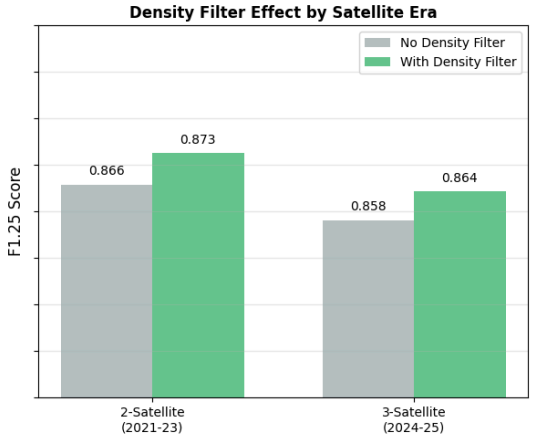

Figure 5: Density Filter Effect on F1.25 by Satellite Era

Key Findings:

- In Figures 4 and 5, we see that across both the two-satellite and three-satellite eras, performance increased as a result of applying the density filter, however, the magnitude of improvement between the two eras was nearly identical indicating that the addition of a third satellite did not introduce any significant noise in detections

- The third satellite likely introduced more overlap in detections for areas that were already well observed

Performance Versus Fire Size and Detection Rate

We also examined the performance of our algorithm against two other key properties, fire size and detection rate. Fire sizes can vary from a few square kilometers to several hundred square kilometers, so it is important that our algorithm generalizes to a broad range of fire sizes. In addition, detection rate can vary greatly by fire considering fire behavior, cloud cover and geolocation error, so it is important for the algorithm to generalize to a broad range of detection points.

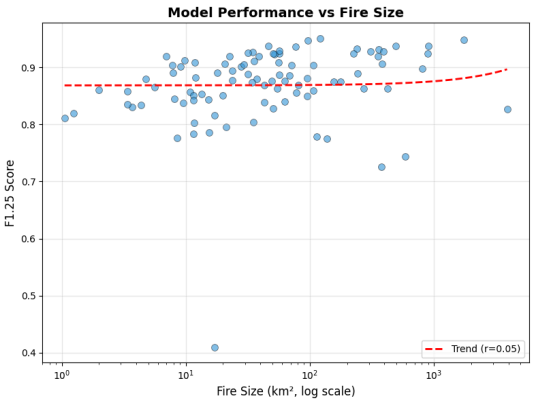

Figure 6: Performance vs Fire Size

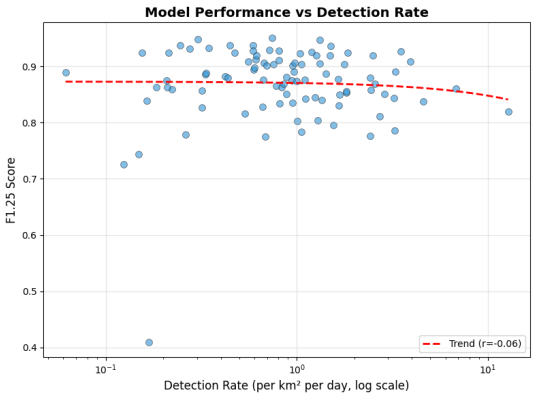

Figure 7: Performance vs Detection Rate

Key Findings:

- In Figure 6 there is a slight positive correlation between performance and fire size with a correlation coefficient of 0.05, indicating that the algorithm generally performs consistent regardless of the size of the fire
- Figure 7, indicates that there is a slight negative correlation between performance and detection rate with a correlation coefficient of -0.06, indicating the algorithm performs consistently regardless of the detection rate of the fire

Prediction Bias Evaluation

To evaluate whether our algorithm consistently over or underpredicts fire size, we computed the relative error between predicted and actual perimeter area for each fire:

$$Relative\ Error\ = \frac{Predicted\ Area - Actual\ Area}{Actual\ Area}$$

A positive value indicates overestimation while a negative value indicates underestimation. Figure 8 shows the distribution of relative error across all fires.

Key Findings:

- As shown in Figure 8 the distribution is right skewed, and we are overpredicting the actual size of the fire around 88% of the time while we are underpredicting around 12% of the time
- This is consistent with our original intent of penalizing missed burned areas at the cost of overpredicting the final area

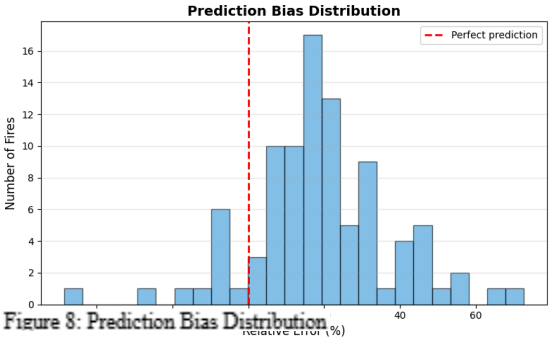

Figure 8: Prediction Bias Distribution

Fire Progression

To assess the quality of our predictions over the lifetime of a fire, we visualized the progression of two fires in our dataset in Figures 9 and 10. Our predictions over time are represented by gradient colored polygons, with the final CAL FIRE perimeter overlaid in black. These visualizations illustrate how the algorithm constructs perimeters at each observation window, showing fire growth over time.

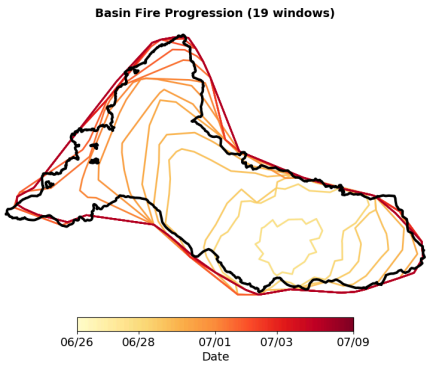

Figure 9: Basin Fire Progression (19 windows)

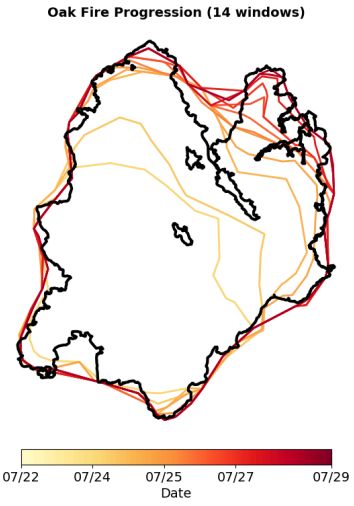

Figure 10: Oak Fire Progression (14 windows)

Discussion

Overall, our algorithm demonstrates strong geometric agreement with official CAL FIRE perimeters, achieving a mean F1.25 score of 0.870 and IoU of 0.751 across 93 fires. By comparing our final predicted perimeters to final CAL FIRE boundaries, we validated that the pipeline from raw FIRMS detections through clustering, filtering, and polygon construction produces realistic perimeter estimates.

Our evaluation showed that the algorithm generalizes well across different conditions. Performance remained consistent between the two-satellite and three-satellite eras, and we found no significant relationship between performance and fire size or detection rate. The density filter helped reduce overestimation caused by geolocation error, with similar improvements observed in both eras.

While our evaluation focused on final-to-final perimeter comparison, visualizing fire progression over time revealed smooth and consistent growth patterns. Perimeters expanded outward gradually without

erratic jumps, suggesting the algorithm produces realistic progression that could be representative of actual fire behavior.

While our algorithm shows strong performance, its use in real-world scenarios carries important considerations. First responders may use live updates for resource allocation, while the public may use them to make evacuation decisions. If users assume these perimeters are exact, they could misjudge the true extent of a fire, placing people and property at risk. To address this, it should be clearly communicated that perimeters are estimates with potential sources of error, including geolocation inaccuracies in FIRMS detections. The algorithm should be used as a supplementary resource, and all decisions should be informed by human judgment.

Having validated the pipeline on historical fires, the next step is translating it into real-time monitoring. FIRMS detections would be pulled from NASA's API as new passes occur, and the algorithm would generate updated perimeters at each observation window. This could feed into a live public dashboard where the public and first responders can track wildfire growth as it happens.

Statement of Work

Jonathan Kelly	Michael Michelini
Compile and clean CAL FIRE perimeter data, data preprocessing pipeline, parameter tuning, visualization creation, report writing	Compile and clean NASA FIRMS data, evaluation pipeline, visualization creation, report writing

References

Bhuian, H., Dastour, H., Ahmed, M. R., & Hassan, Q. K. (2024). Comparison of perimeter delineation methods for remote sensing fire spot data in near/ultra-real-time applications. *Fire*, *7*(7), 226. https://doi.org/10.3390/fire7070226

Briones-Herrera, C. I., Vega-Nieva, D. J., Monjarás-Vega, N. A., Briseño-Reyes, J., López-Serrano, P. M., Corral-Rivas, J. J., Alvarado-Celestino, E., Arellano-Pérez, S., Álvarez-González, J. G., Ruiz-González, A. D., Jolly, W. M., & Parks, S. A. (2020). Near real-time automated early mapping of the perimeter of large forest fires from the aggregation of Viirs and Modis active fires in Mexico. *Remote Sensing*, *12*(12), 2061. https://doi.org/10.3390/rs12122061

California fire perimeters (all). California State Geoportal. (n.d.). https://gis.data.ca.gov/datasets/CALFIRE-Forestry::california-fire-perimeters-all/explore

Chen, Y., Hantson, S., Andela, N., Coffield, S. R., Graff, C. A., Morton, D. C., Ott, L. E., Foufoula-Georgiou, E., Smyth, P., Goulden, M. L., & Randerson, J. T. (2022). California wildfire spread derived using viirs satellite observations and an object-based tracking system. *Scientific Data*, *9*(1). https://doi.org/10.1038/s41597-022-01343-0

Historical fire perimeters. CAL FIRE. (n.d.). https://www.fire.ca.gov/what-we-do/fire-resource-assessment-program/fire-perimeters

NASA. (n.d.). *Firms*. NASA. https://firms.modaps.eosdis.nasa.gov/