# Early Detection of California Wildfires - Project Overview

Project Repository: https://github.com/MThorpester/Wildfire-Detection.git

California once had 660 fire lookout towers/cabs that were manned for decades by personnel monitoring for wildfires. Since the 1990’s most of them have been retired from service as various terrestrial, aerial, and satellite-based technologies that are equipped with visible, IR, or multispectral sensors have become more commonplace. Today, only 38 of those towers remain in operation, mostly in our far northern forests.

This project will focus on wildfire detection using *terrestrial-based optical cameras*. Over the last decade, an increasing number of PTZ (Pan-Tilt-Zoom) fire cameras have been installed across California by the HPWREN and ALERTWildfire consortiums:

* <https://hpwren.ucsd.edu/>
* <http://www.alertwildfire.org/>

These cameras are monitored by firefighters and the general public to 1) detect wildfire ignition, 2) verify & locate reported fires, and 3) monitor wildfire behavior. The cameras have proven highly effective for #2 and #3 - over the last 5 years they have provided critical information for over 500 wildfires. But for people to manually detect fires using these cameras requires that they stare at them for hours on end looking for gradual changes…..a task much better suited to computers!

With this project, we want to create a prototype that automates the early detection of wildfires by scanning real-time webcam images using a Convolutional Neural Network. We plan to leverage these camera networks for historical images to train, test and tune our CNN model. We will then deploy a prototype that uses the CNN model to monitor images in real-time at a single camera location (Deerhorn, Humboldt County. CA) and notifies a human when a potential wildfire is detected.

# Datasets and Resources

|  |  |  |
| --- | --- | --- |
| Source | Link | Description |
| Open Climate Tech | https://github.com/open-climate-tech/firecam/tree/master/datasets/2019a | Smoke (with bounded box file) and non-smoke datasets culled by volunteers from HPWREN images. Smoke images are from fires started in 2019. |
| HPWren Archive | http://hpwren.ucsd.edu/HPWREN-FIgLib/ | Archive of selected fire sequences from HPWren cameras, a few bounded-box files. |
| Monitoring Location | http://www.alertwildfire.org/northcoast/index.html?camera=Axis-DeerHorn2&v=7a7f1c3 | Camera at Deer Horn, Humboldt County |
|  |  |  |

# Data Pre-processing

A significant amount of work involves the preparation of the smoke image datasets to train the neural network. The non-smoke image dataset requires less preparation as the dataset is large (45,000 images) and the images are already sized to 299x299.

## Crop Smoke Images

The 1800 smoke images from the Open Climate Tech 2019 dataset are 3072x2048 pixels in size. The neural net expects images sized at 299x299. Resizing a 3072x2048 pixel image with a tiny plume of smoke down to 299x299 would lose all pixels of smoke. So these 1800 images will need to be cropped to include the smoke plumes (specified by the bounding boxes in the corresponding .csv file) to a size that will not be compromised by resizing to 299x299 (so a maximum of 598x598).

## Augment Smoke Images

Since there are only 1800 smoke images in this dataset (a very small number for a neural net to learn such a complex task from) we will augment these images using several techniques that will boost the dataset to 18,000 images (and also make the image recognition more robust):

* Crop images shifting the relative location of the smoke in each image (for example, one cropped image will have smoke in center, 2nd will have smoke in right lower corner, 3rd will have smoke in upper right hand corner, 4th will have smoke in lower left hand corner, 5th will have smoke in upper left hand corner)
* Horizontally flip each image.

# Design and Train Neural Net

The neural net will be a CNN (Convolutional Neural Net), which are good for image classification tasks. We will start with a simple architecture and expand and tune it as time permits. If we are able to achieve reasonable accuracy (=>75%) and still have time available we may incorporate Transfer Learning and further augment our smoke training dataset:

### Transfer Learning

If time permits, we would like to create a version of our neural net that uses the xCeption neural net as a pre-trained base to see if this enhances our detection accuracy.

### Enriching the Smoke Training Dataset

If time permits, we would like to create additional smoke images from the HPWren archive of fire sequences. Most of these would need to be manually reviewed and cropped, as only a couple of hundred of them have bounding box files. Increasing the number of smoke images in the training dataset will likely increase the detection accuracy of the neural net.

# Deploy Neural Net

The neural net model will be saved, deployed to the server and called via API from our web application.

## Web Application

We will add a Wildfire Detection page to the California Wildfire Dashboard. The page will allow the user to turn on monitoring of the Deerhorn AlertWildfire camera. This will cause an image to be scraped from the webpage every minute and sent to the deployed neural net for prediction. Since these images are 1920 x 1080, the program will need to split them up into overlapping 299 x 299 squares and pass them individually to the neural net. If one or more of the (24 or so) 299x299 squares receives a smoke- positive prediction, then “Smoke Detected” will be displayed to the user with red bounding boxes showing the square(s) that got the positive score.

# Additional Resources

* Tools for creating diagrams of neural network architectures: <https://github.com/ashishpatel26/Tools-to-Design-or-Visualize-Architecture-of-Neural-Network>
* Sample code for extracting & augmenting images based on bounding boxes
  + <https://github.com/open-climate-tech/firecam/blob/master/firecam/image_crop/recrop_min_size.py>
* LIME – tool for explaining classifier
  + <https://github.com/marcotcr/lime>
  + https://medium.com/applied-data-science/a-case-for-interpretable-data-science-using-lime-to-reduce-bias-e44f48a95f75

# Key Project Milestones

## Phase 1 – Data Pre-processing & Neural Net Design

### Tasks

* Create Neural Net prototype
  + Manually create small training & validation dataset
  + Prototype architectures and design parameters
  + Save prototype model for initial deployment
* Create Augmented Smoke Dataset
  + Write program to shift, flip, crop and save 2019 smoke ds images
  + Collect & document additional smoke images from HPWren archive
    - Use above program to extract, augment & save new images

### Deliverables

* V1 CNN w/saved model
* Program that automatically extracts & augments smoke images
* Full smoke image dataset (8,000+ smoke images)

## Phase 2 – Neural Net Training & Web App Deployment

### Tasks

* Train, validate and tune CNN on full dataset
* Create and deploy web page
  + Create new web page
  + Monitor & capture webcam images
  + Send images to CNN for prediction
  + Display results

### Deliverables

* Trained, tuned & deployed neural net model
* Wildfire Detection page - html, css & javascript files
* Python program that passes images to model & gets prediction

## Project Finalization

### Tasks

* Create project presentation
* Clean up and document github repository
* Present project

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| May-June | | | | | | | | | | | | |
| Sun |  | Mon |  | Tue |  | Wed |  | Thu |  | Fri |  | Sat | |
|  |  | 24  **Project Kick-off** |  |  |  | 26 |  |  |  |  |  | 29 | |
|  |  |  |  |  |  |  |  |  |  |  |  |  | |
| 30  **Phase 1 Complete** |  | 31 |  | 1 |  | 2 |  | 3 |  | 4 |  | 5 | |
|  |  |  |  |  |  |  |  |  |  |  |  |  | |
| 6  **Phase 2 Complete** |  | 7  Prepare Project Presentation |  | 8  Clean Up & Document Repo |  | 9  **Project Presentation** |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  | |

