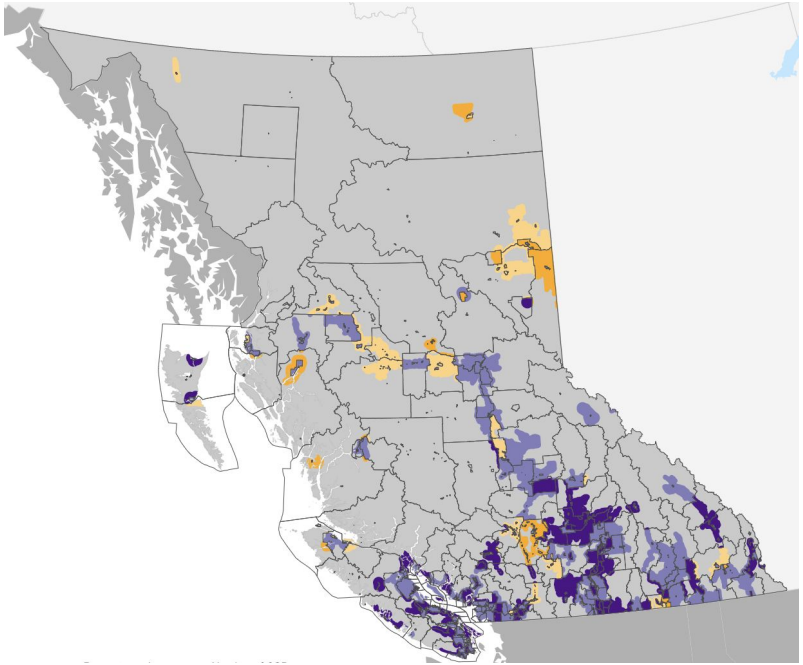
Image Classification of Forest Fires with Neural Networks

Rodrigo Becerra Carrillo

Problem Statement

According to the BC government, about 40% of forest fires are reported by the general public, in addition to other detection strategies such as:

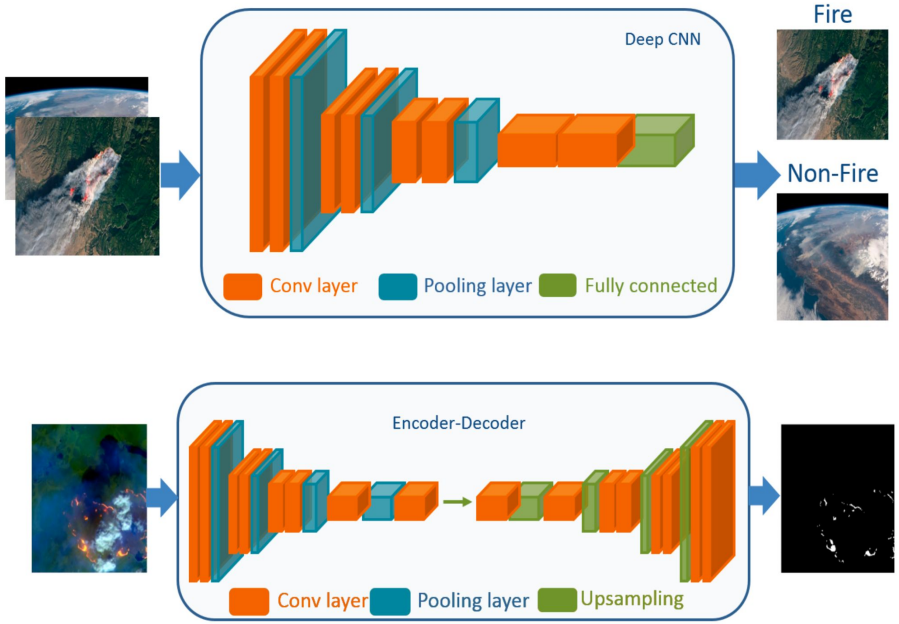
- Air patrols
- Fire warden ground patrols
- Infrared technology
- Computer technology and predictive software
- Lookout towers



Census Canada gray areas are sparsely populated

Proposed Solutions Using Data Science

- There is an opportunity to use Deep Learning (DL) models for early automated detection of fires
- Image classification and segmentation architectures could be used to track and characterize fires
- Reported accuracy scores for DL models with this kind of classification task are 95% [1]



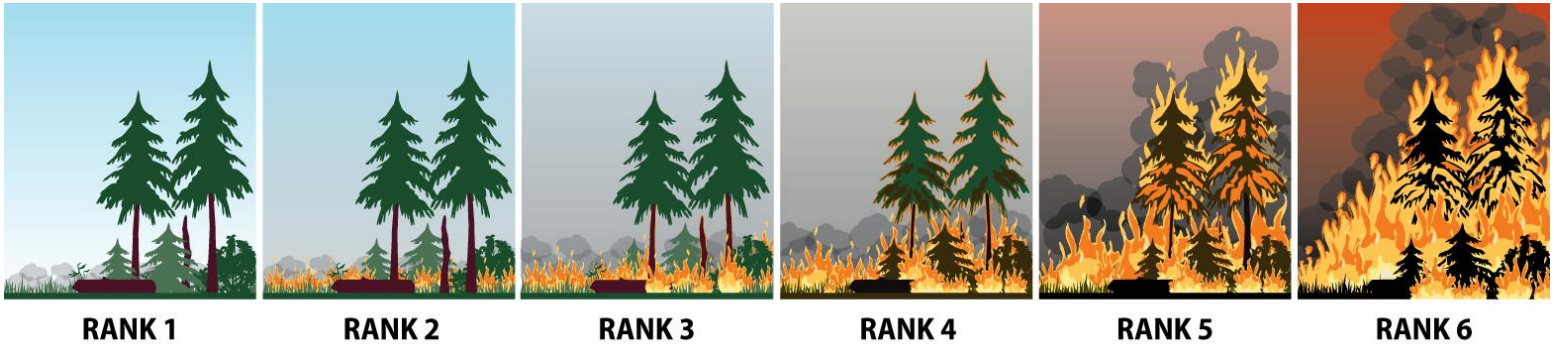
Figures adapted from https://doi.org/10.3390/fire6050192

[1] https://doi.org/10.1155/2022/5358359

Impacts of Proposed Solution

Having an early detection of fires can impact the response time and management before they become too large to control

The BC Wildfire Service uses a ranking scale based on visual indicators to describe fire behaviour



The financial burden on taxpayers could be reduced. In 2023 forest fires incurred an over budget of > $700 M for the provincial government in BC.

Dataset Details

Dataset_01: Fire Dataset

- 999 PNG images (75% fire/25% non-fire)
- Total size 406 MB
- On average 750 x 1180

Dataset_02: Forest Fire Dataset

- 1900 JPG images (50% fire/50% non-fire)
- Total size 149 MB
- **All images 250 x 250**
- Same authors as Dataset_01 used in a publication

Dataset_03: The Wildfire Dataset

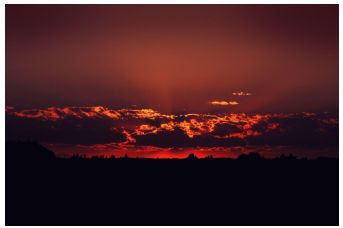
- 2700 PNG and JPG images
- 40% fire/60% non-fire
- Total size 11 GB, variable size
- **Includes confounding elements**

No significant quality issues encountered so far

- Number of channels: [4, 250, 250] or [1, 250, 250] → convert to RGB with PIL
- Oversized images: [3, 256, 256] → use PyTorch `crop` and `save_figure`

train: fire

test: non-fire



test: non-fire

Next Steps

1. Setup development environment for the PyTorch library
   a. Setup `DataLoader` class → **Complete**
   b. Organize Python modules in `src/` directory

2. Train an image classifier with same-sized fire and non-fire images
   a. Preprocess images to ensure they are all the same size → **Complete for Dataset_02**
   b. Investigate accuracy metrics for classification tasks (accuracy, precision, recall, F1 score)
   c. Implement simplest CNN model (i.e. LeNet)
   d. Identify state-of-the-art models that could do transfer learning (i.e. VGG, ResNet)

3. Investigate segmentation of images
4. Investigate resizing of images
   a. Evaluate what strategy to pursue for image resizing