Image Classification of Forest Fires with Neural Networks

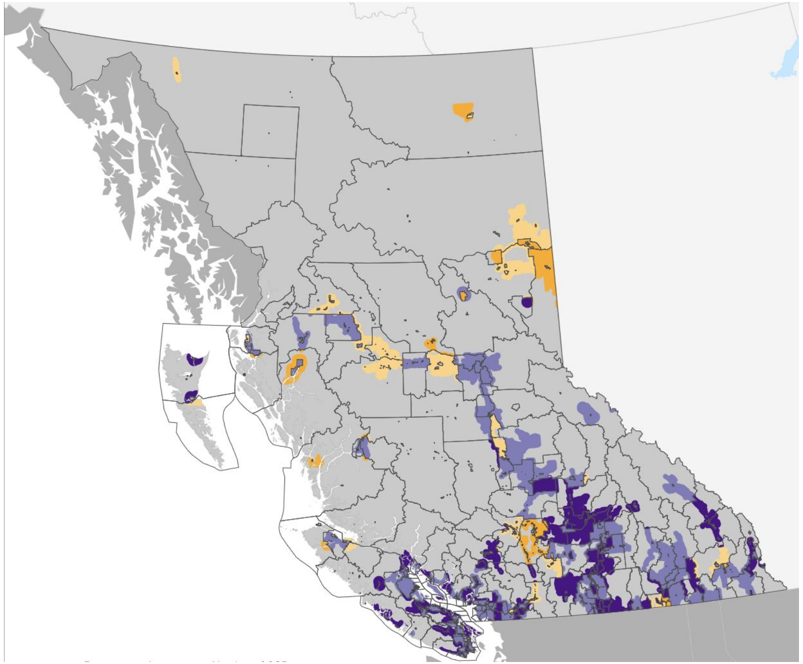
Sprint 2 Update

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:

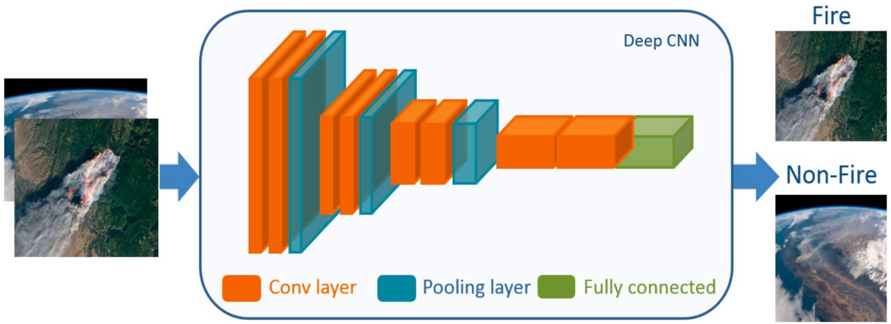
- Air patrols
- Fire warden ground patrols
- Infrared technology
- Computer technology and predictive software
- Lookout towers

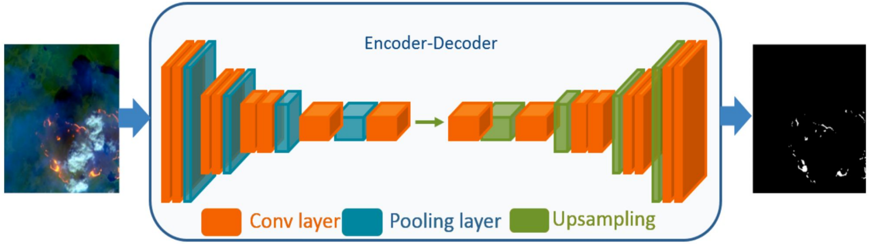


Census Canada gray areas are sparsely populated

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]





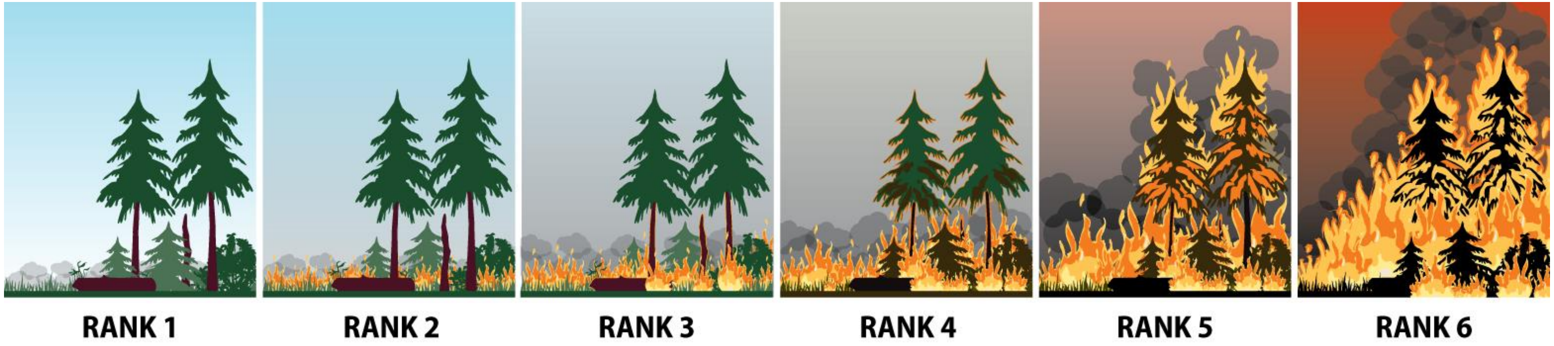
[1] https://doi.org/10.1155/2022/5358359

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

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 and Preprocessing

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 image size
- **Includes confounding elements:**

Preprocessing Steps

Before Modelling:

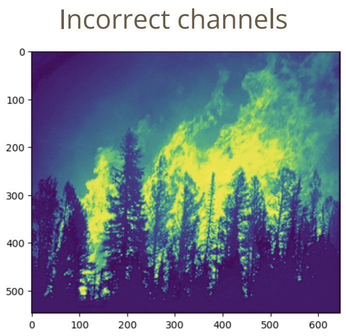
1. Image cleaning (formatting, channels, resizing)
2. Create annotations file (csv)

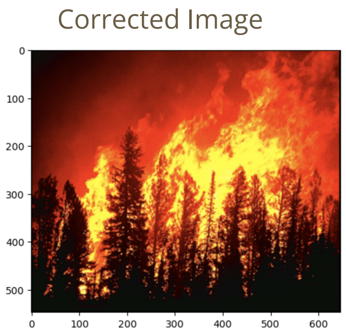
Before Training:

1. Create Dataset (custom PyTorch class)
2. Obtain statistics (mean, std) for train Dataset
3. Define transformations
   a. Resizing
   b. Normalization (from [0, 255] → [-1,1])
4. Create DataLoader
   a. Specify batch size and shuffling

Findings from EDA

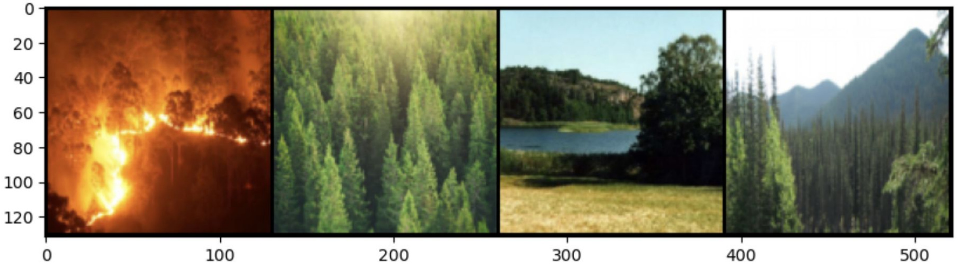
- Datasets are mostly balanced
- No significant quality issues
- Images were not mis-labelled





Important things to note:

- Images must be resized to the same shape to be used by PyTorch
- Datatypes are very important
  - PIL Images
  - `uint8` tensors [0,255]
  - Un-normalized Tensor [0,1]
  - Normalized Tensor [-1,1]



Baseline Models and Evaluation Metrics

LeNet5

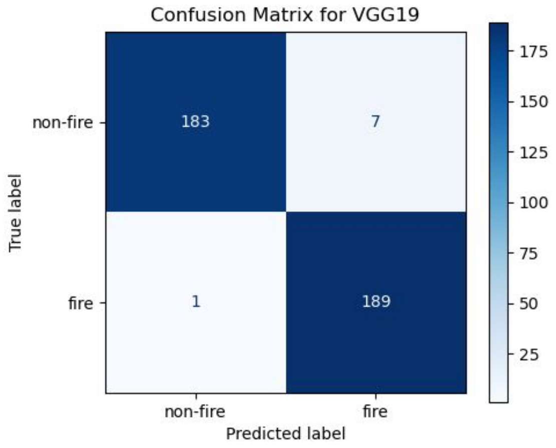
- Very simple CNN originally developed for MNIST Dataset
- Only works for 32 x 32 images

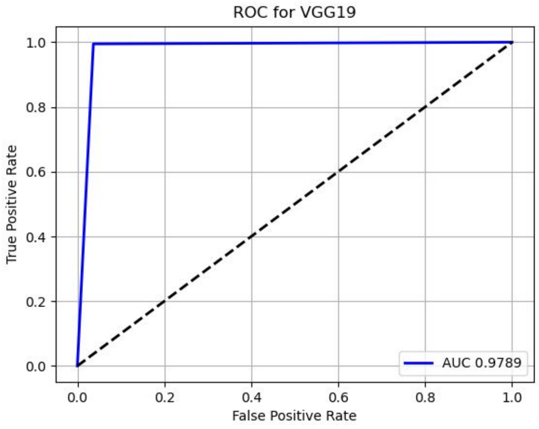
VGG19

- Same model used by Dataset_01 in a [publication]
- Intended for 224 x 224 images
- Suitable for transfer learning

ResNet18

- Uses residual connections to address vanishing gradients
- It has less trainable parameters compared to VGG
- Suitable for transfer learning





	precision	recall	f1-score	support
0	0.99	0.96	0.98	190
1	0.96	0.99	0.98	190

Note these values match well with what has been published

Next Steps

1. Train an image classifier with same-sized fire and non-fire images

   ✔ Train VGG19 `02_fire_dataset` with transfer learning, using the same hyperparameters as authors.

   ✔ Train ResNet18 with `02_fire_dataset` using same hyperparameters as authors.

2. Investigate segmentation of images – CANCELLED out of scope
3. Make a new version of the `03_the_wildfire_dataset` with square images
4. Compare VGG19 and ResNet
   - What is the nature of False Positives and False Negatives?
   - What is the most important metric for real-world deployment of these models?
   - How do models perform doing predictions from images with confounding elements (smoke, sunlight)?
   - Can the models be re-trained with the `03_the_wildfire_dataset` and still perform well on unseen data?