Image Classification of Forest Fires with Neural Networks

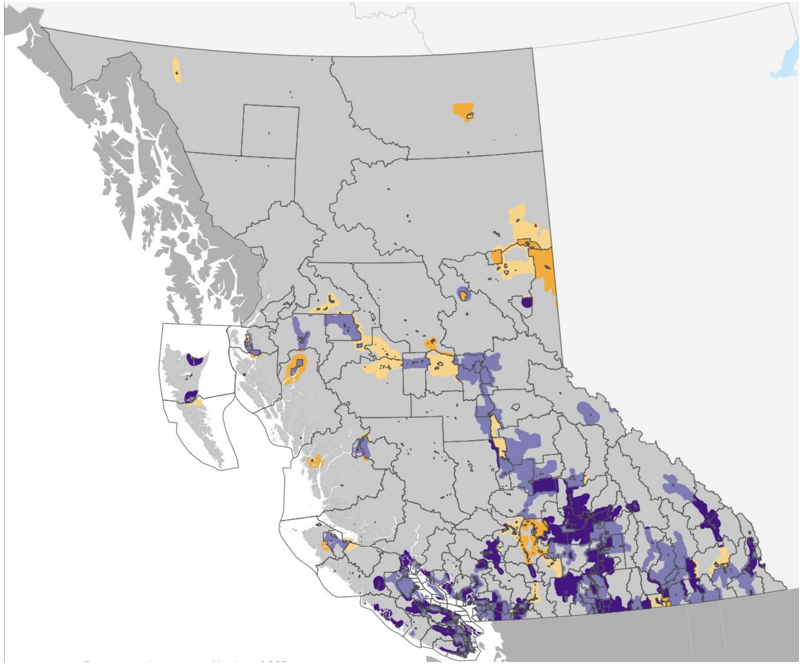
Sprint 3 Update

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:

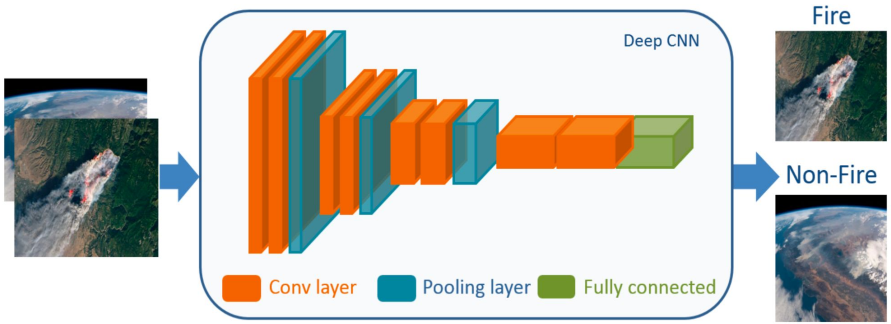
- Air patrols
- Fire warden ground patrols
- Infrared technology
- Computer technology and predictive software
- Lookout towers

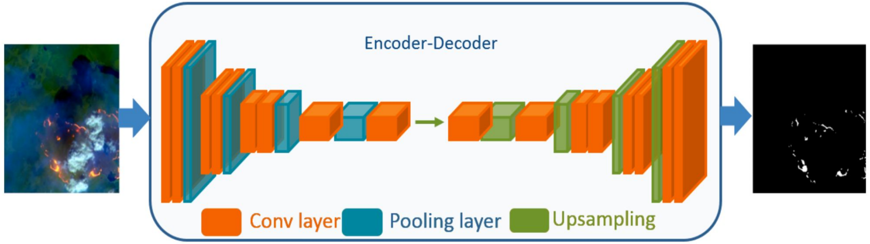


Census Canada gray areas are sparsely populated

Proposed Solutions Using Data Science

- There is an opportunity to use Deep Learning (DL) models for to 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]





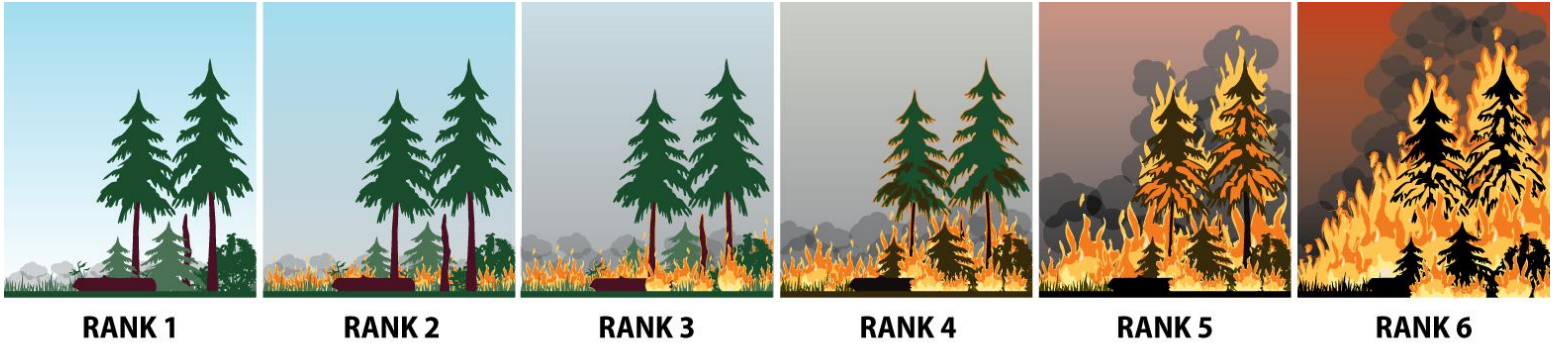
Figures adapted from
https://doi.org/10.3390/fire6050192

[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 and Preprocessing

Forest Fire Dataset (aka DeepFire)

- 1900 JPG images (50% fire/50% non-fire)
- Total size 149 MB
- **All images 250 x 250**
- Article compares VGG19 with non-DL models [1]

The Wildfire Dataset [2]

- 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

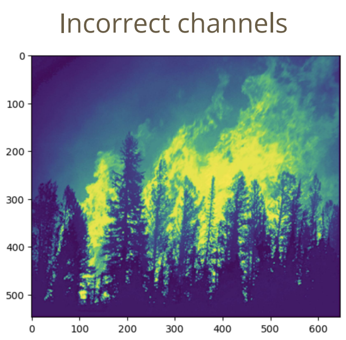
[1] see https://doi.org/10.1155/2022/5358359
[2] *Forests* **2023**, *14*(9), 1697; https://doi.org/10.3390/f14091697

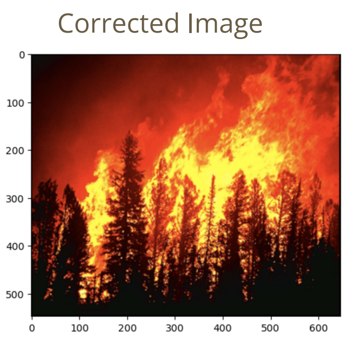
Findings from EDA

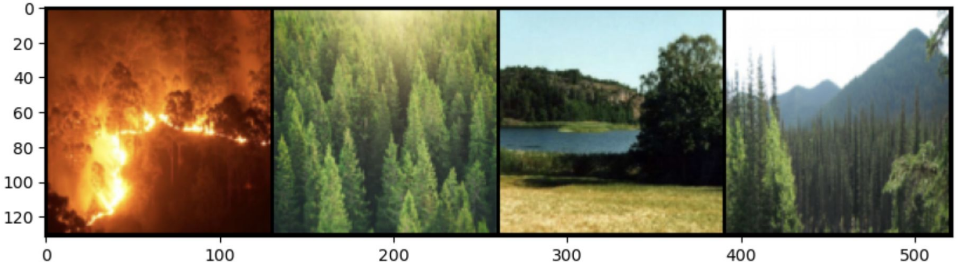
- Datasets are mostly balanced
- No significant quality issues
- Images were not mis-labelled

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]



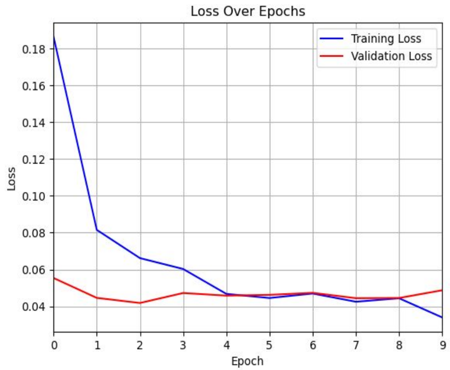


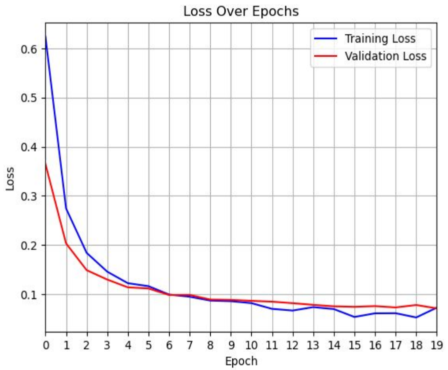


Baseline Models and Evaluation Metrics

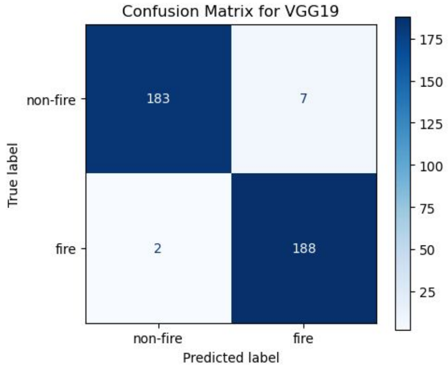
Performed Transfer Learning with VGG19 and ResNet18 on DeepFire dataset:

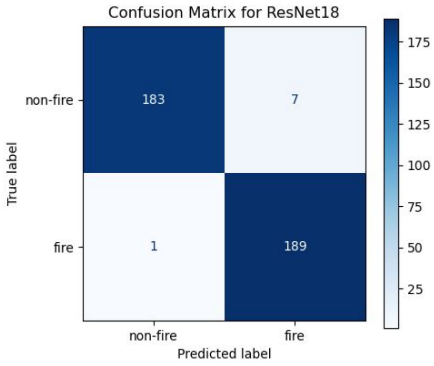
- Both models outperform the results from DeepFire paper
- They both generalize very well
- Since we cannot afford to miss any detection, recall score should be prioritized





	VGG19 DeepFire paper [1]	VGG19	ResNet18
total parameters		139,578,434	11,177,538
trainable parameters		8194	**1026**
model size (MB)		560	**45**
training time		15 min	**12 min**
accuracy	0.9500	0.9763	0.9789
precision	0.9572	0.9641	0.9643
recall	0.9421	0.9895	0.9947
F1 score	0.9496	0.9766	0.9793
AUC score	0.9889	0.9763	0.9789





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

False Positives and False Negatives

FP - VGG19

true: 0
predicted: 1

true: 0
predicted: 1

FN - VGG19

true: 1
predicted: 0

true: 1
predicted: 0

FP - Both Models

true: 0
predicted: 1

true: 0
predicted: 1

true: 0
predicted: 1

true: 0
predicted: 1

FP - ResNet18

true: 0
predicted: 1

true: 0
predicted: 1

FN - ResNet18

true: 1
predicted: 0

This foreshadows that models won’t perform well with smoke, sunsets, and fall-like colours

Prediction with the Wildfire Dataset

Using the Wildfire Dataset (410 test images), both models don’t maintain their performance metrics

Smoke and fire

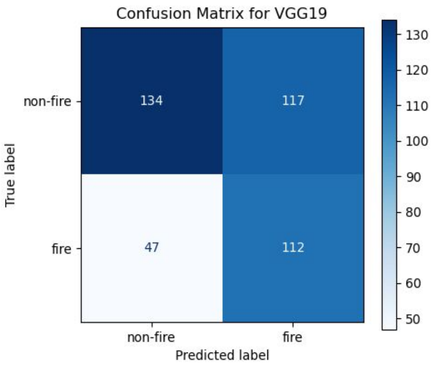
Smoke from fires

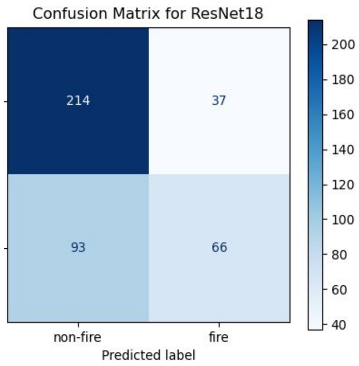
Fire confounding elements

Forested areas without confounding elements

Smoke confounding elements

predictions





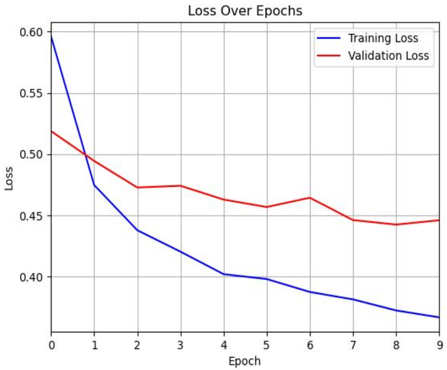
	VGG19	ResNet18
accuracy	0.6000	0.6829
precision	0.4891	0.6408
recall	0.7044	0.4151
F1 score	0.5773	0.5038

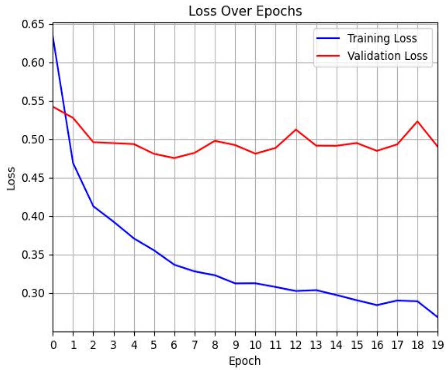
Retraining with the Wildfire Dataset

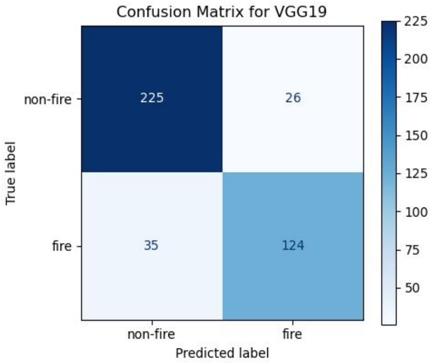
Both models improve their performance after training on the wildfire dataset:

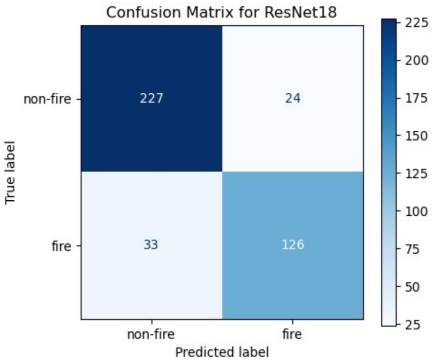
- The performance is on par with what has been reported in the literature [2]
- The models don't generalize as well as before
- Recall score could be further improved
- Additional training strategies are possible (see next slide)

	MobileNetV3 [2]	VGG19	ResNet18
accuracy	0.8405	0.8512	0.861
precision	0.8322	0.8267	0.84
recall	0.7799	0.7799	0.7925
F1 score	0.8049	0.8026	0.8155
AUC score	0.8397	0.8381	0.8484









[2] *Forests* **2023**, *14*(9), 1697; https://doi.org/10.3390/f14091697

Conclusions and Next Steps

- All performance metrics are in close agreement with what has been reported in the literature
- Training with the Wildfire Dataset simulates a more realistic use case
- It’s helpful to have publications to guide your progress
- Real world use-cases need to be incorporated, as these will also influence the complexity of datasets and models

Personal Goals

- Implement Device Selection (CPU, GPU, MPS)
- Implement more scripting and automation
- Explore additional training strategies
  - Multiclass classification (→ need to rebalance dataset)
  - Transfer Learning + Fine-tuning
  - Hierarchical structure classification
- Experiment with Vision Transformers

Plan for Demo Day (streamlit)

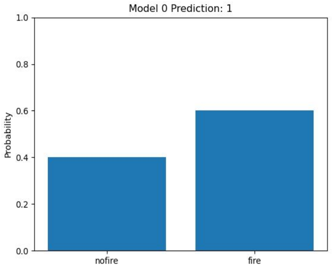
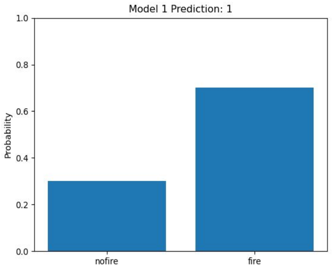
1 Choose Dataset

2 Choose Sample Image

3 Predict

[ ] show code

DANGER

Training Dataset	VGG19	ResNet18
DeepFire		
Wildfire Dataset