WILDFIRE AI

Albin Thorsson, Alicia Toom, Andreas Eliasson,
Markus Antonsen, Mattias Barth

Purpose

Project Idea

The purpose of this project is to automatically monitor and locate wildfires from drone images. Right now you have to manually analyze the images, which is very time consuming.

Our solution is an API where the drone uploads their images and/or videos, and gets information in return. The return information contains whether the image shows signs from a wildfire, and if so, where in the image the fire is located and whether to notify proper authorities.

Work Timeline

Data Analysis

Our initial work consisted of analyze, sort and process the nearly 5000 images we got.

We classified them by “fire” or “not fire” and sorted them into different datasets for the training phases of our model.

Model Selection And Training

We explored and tested several models and algorithms, before settling with a Neural Network.

The training of the model is done mainly in the Cloud, using Google services.

We re-trained and fine-tuned the model several times, to ensure the best accuracy and result.

User Interface

We discussed some different ideas for the UX and UI, and decided that a web based API is the way to go.

The set up for our infrastructure is set and up and running.

Version 1 RnD

Research and developed a Image Classification model, using a Neural Network.

Added initial application structures and functionalities.

Version 2 RnD

Added a Object Detection model to the API and updated the web UI.

Data






The initial dataset contained about 5000 images, which we sorted in FIRE, NON-FIRE and SATELLITE classes, and then resize the images to 250x250 pixels before further usage.

Initial Dataset

FIRE IMAGES

NON FIRE IMAGES

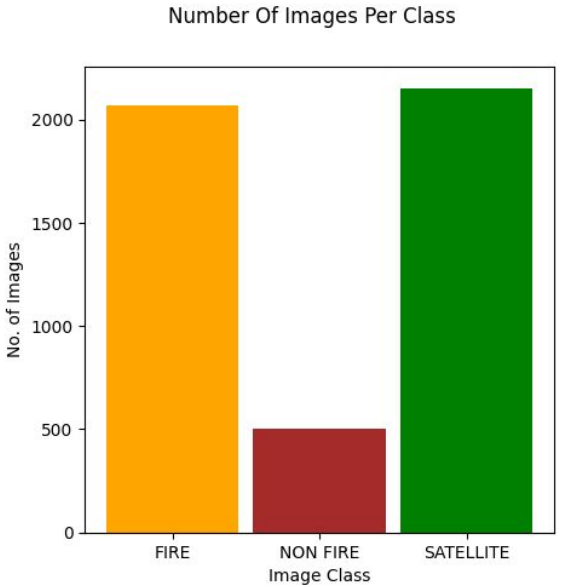
SATELLITE IMAGES

Initial Dataset

After sorting the images we decided not to use the satellite images, at least not for the initial Image Classification model.

But then we discovered that it would result in almost 5 times more fire images than non-fire images.

So we shortened the dataset and used 500 of both fire- and non-fire images.

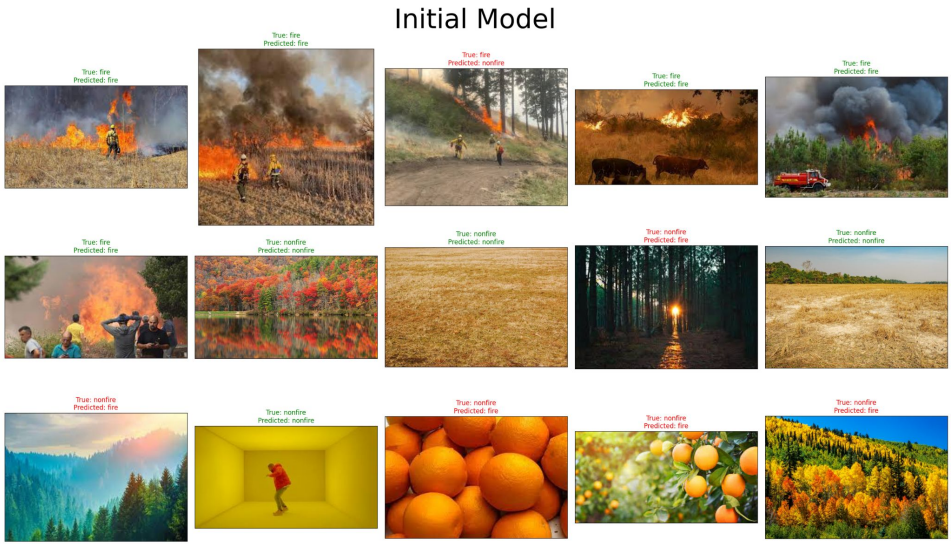


Data Overfitting

Our initial model showed great promise in accuracy and other metrics, but failed when tested it on random images.

We identified that it had been overfitted from our dataset and interpreted many images with orange and yellow colors as “fire”.

After a process of evaluating our training images, and re-training the model, we got a solid but not perfect result.



New Model

True: fire
Predicted: fire

True: fire
Predicted: fire

True: fire
Predicted: nonfire

True: fire
Predicted: fire

True: fire
Predicted: fire

True: fire
Predicted: fire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

True: nonfire
Predicted: nonfire

Bounding Box-Labeled Dataset

Model

What Algorithm?

We explored several algorithms for both images classification and object detection

Image Classification

- Feedforward Neural Network (FNN)
- Convolutional Neural Network (CNN)
- AutoML - Autogluon
- XGBoost

Object Detection

- Region-based Convolutional Neural Network (RCNN)
- You Only Look Once (YOLO)
- Single-Shot Detector (SSD)

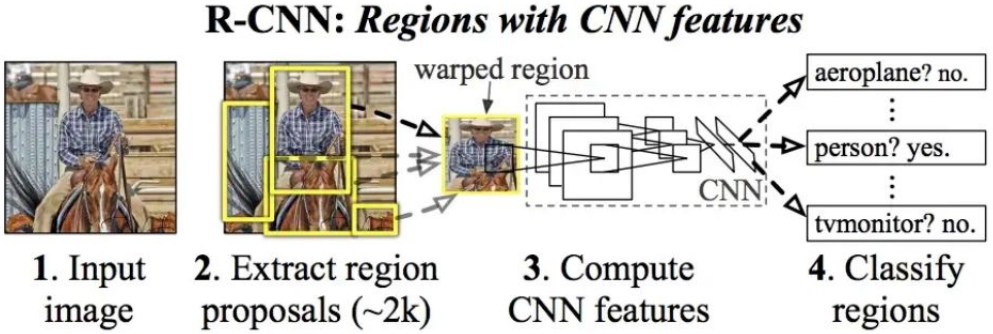
We have decided to use Convolutional Neural Networks for both image classification and object detection.

Convolutional Neural Network

- For the **Image Classification** we used a **Convolutional Neural Network (CNN).**
- A **CNN** is a neural network that is useful in **deep learning and image recognition**. It stands out for its ways to process pixel data, through convolutional layers before output the result.
- It can also **automatically detect features** from input data, which minimise the human effort.
- We decided to go with a CNN for its **straightforward** usage and that it **showed best results** early in our exploring process.

Region Based Convolutional Neural Network

- For the **Object Detection** we used a Region Based Convolutional Neural Network (RCNN).
- It is a type of **object detection algorithm** that is designed to identify objects within **an image or video and draw a bounding box around each object**.
- Object detection algorithms like R-CNN can be trained to detect the presence of wildfires in **satellite or aerial images,** and also images **from the ground**.
- In our case, we use just **images** and from the **ground only** to train the model.



Application Versions

Version 1.0

A simple web-API, where the user upload images and it returns if there's an wildfire on each image or not.

Choose the images you want the model to predict on

CTR + click to select several images

Images

Coordinates

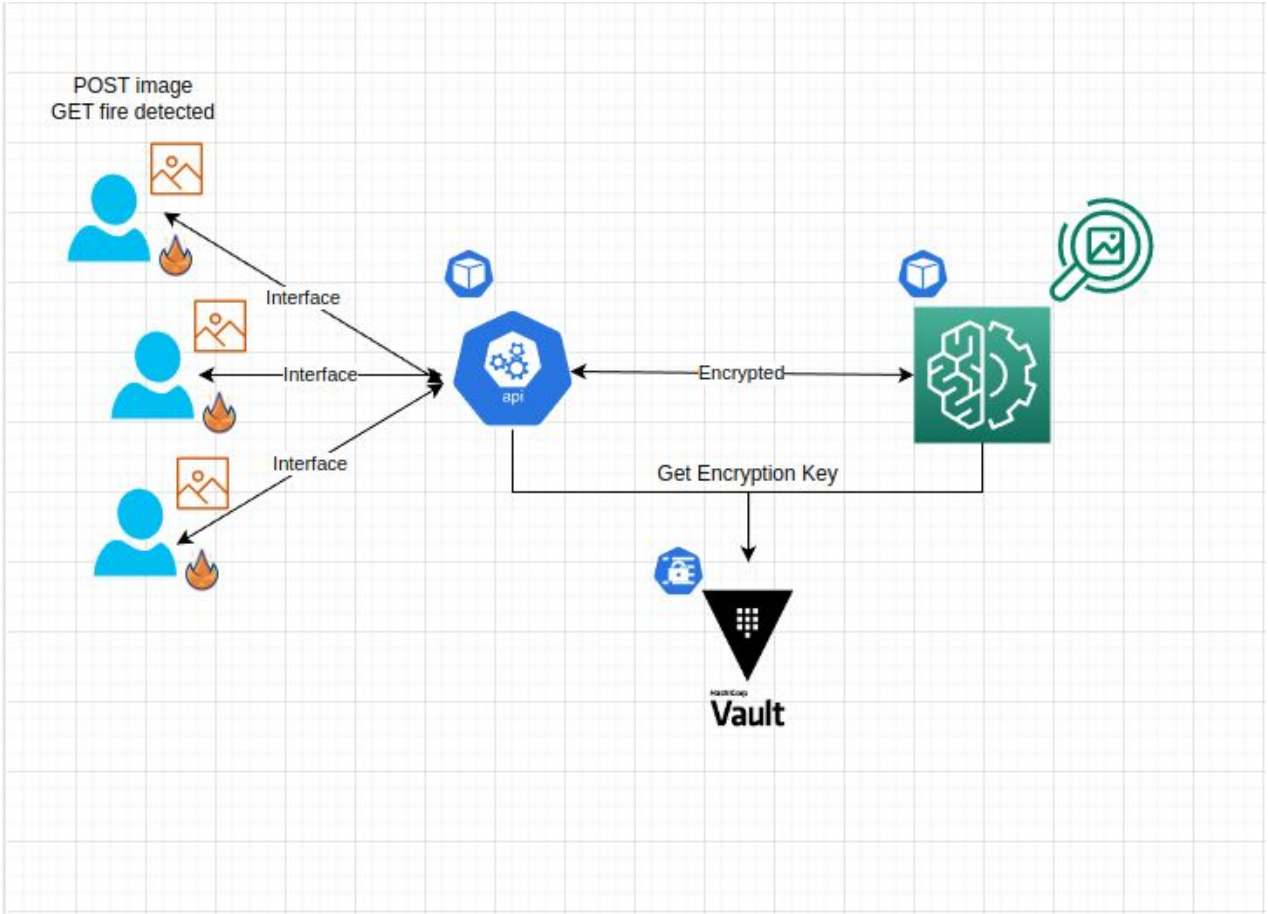
Välj filer | 3 filer | 41 24.2028, 2 10.4418 | Upload

[["Fire detected", "fire-30.jpg"], ["Fire detected", "fire-31.jpg"], ["Fire detected", "fire-32.jpg"]]

Coordinates = ['']

Version 1

System architecture



Current status

Version 2.0

Object Detection

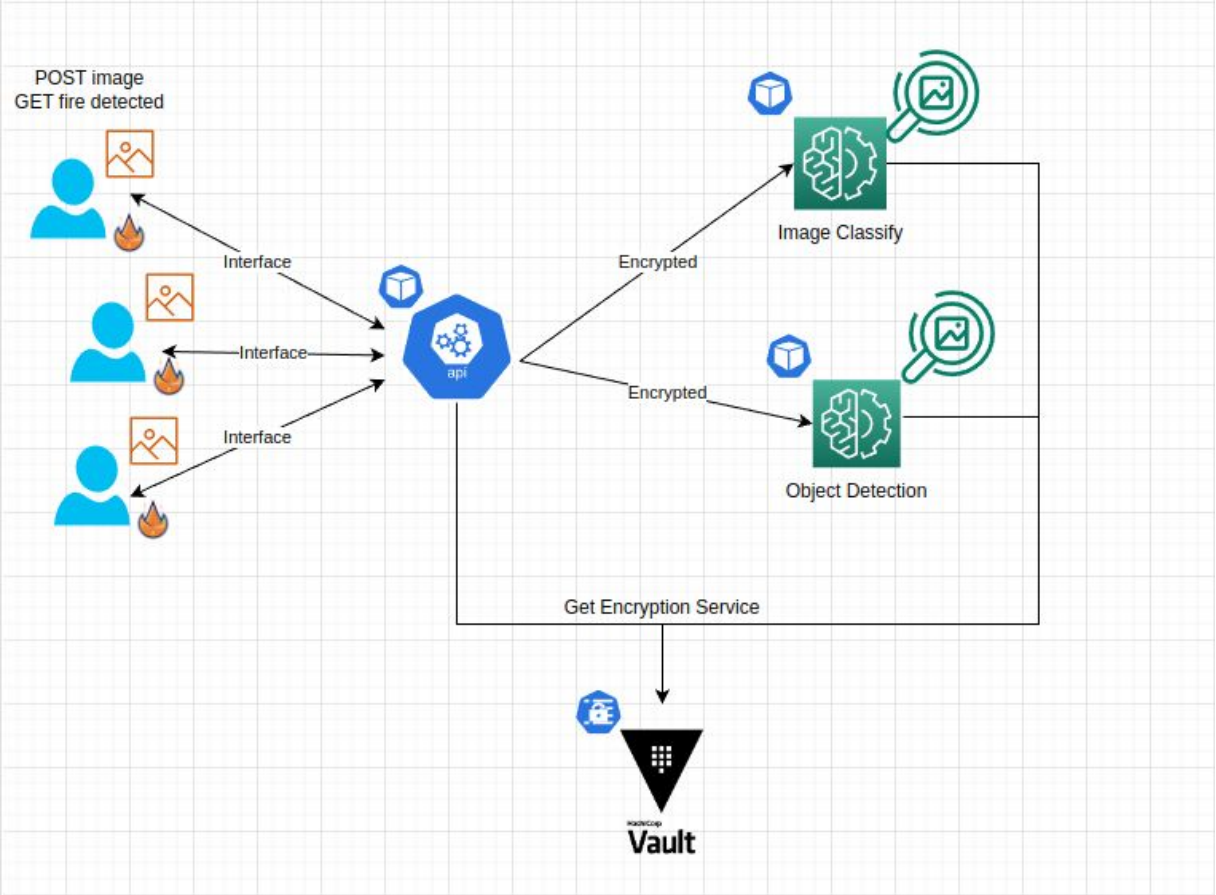
Show more accurately where the fire is located on the images

Advanced System Architecture

More architectural layers to the application structure and more advanced encryption

Version 2

System architecture



Next step

Version 3.0

Video Analysis

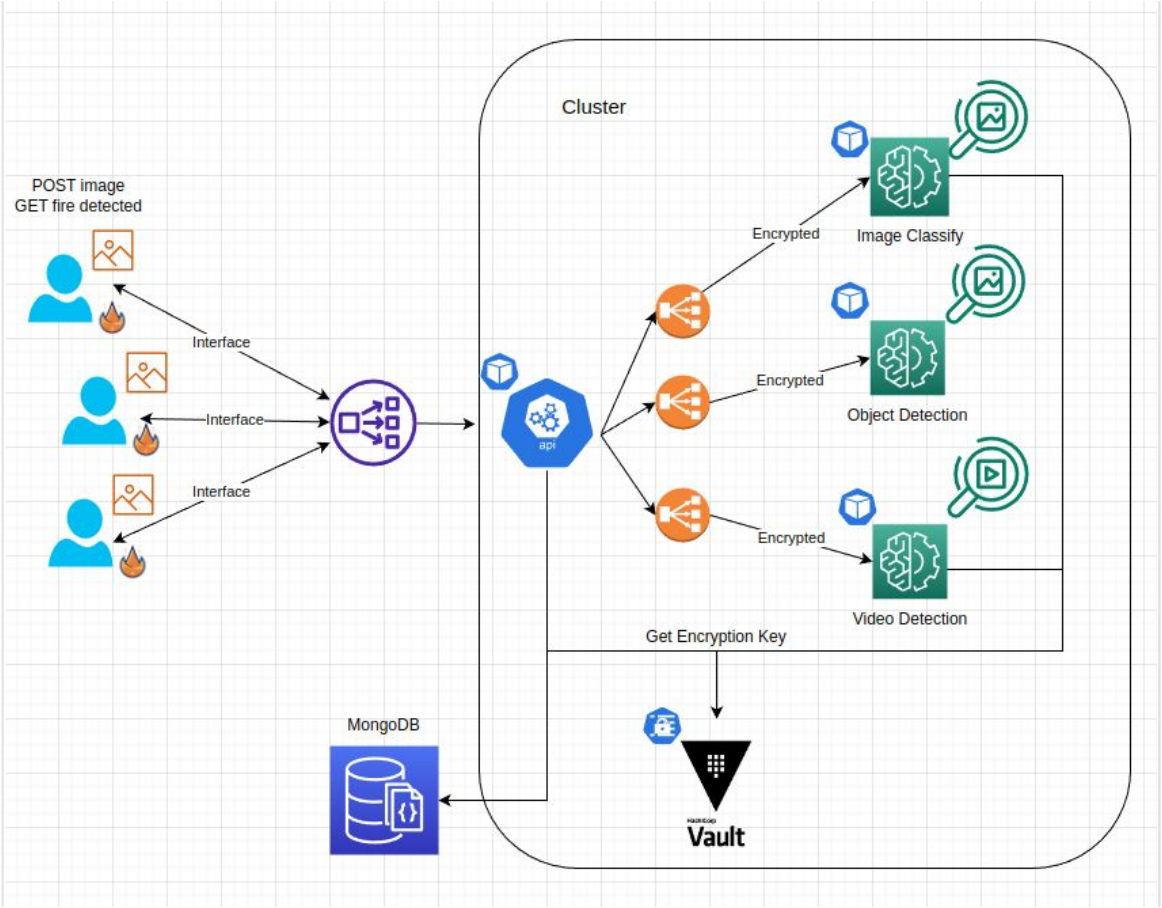
Implement the usage of real-time videos to the system

Database

Adding a database structure to store uploaded images and eventually automatically re-train our models

Future... Think bigger

System architecture



Demo Time