|  |  |  |
| --- | --- | --- |
| 🔥 Wildfire Risk Detection | | |
|  |  |  |
| WIFIRE Satellite Image | | |

# 📘 Project Overview

This project performs wildfire risk analysis using a single satellite image from the WIFIRE project. The image contains terrain features encoded in RGB channels:

* Red → Altitude
* Green → Slope
* Blue → Aspect (direction of slope)

Using simple threshold-based rules, we classify each pixel as either fire-prone or not fire-prone, and visualize the result as a binary risk mask. This project serves as a basic introduction to image-based analysis using NumPy, without requiring advanced machine learning or deep learning libraries.

# 🎯 Objective

* Load and analyze satellite image data (RGB terrain features).
* Perform Exploratory Data Analysis (EDA) on terrain variables.
* Build a basic ML classifier using rule-based logic.
* Generate and visualize a fire risk mask.
* Save the output for further ML use (e.g., model training, GIS tools).

# 🗃️ Dataset

Source: WIFIRE (Wildfire Integrated System)

Image Description: A 3-channel RGB image (sd-3layers.jpg)

Each pixel encodes:

* Red (0–255): Altitude
* Green (0–255): Slope
* Blue (0–255): Aspect (mapped to degrees 0–360)

# 🧪 Techniques Used

|  |  |
| --- | --- |
| Step | Description |
| Image Loading | Used PIL and NumPy to extract pixel-level terrain data |
| EDA | Visualized histograms of Altitude, Slope, and Aspect |
| Classification | Applied a **manual threshold rule** to define fire-prone pixels |
| Mask Creation | Created a **binary mask** (1 = fire-prone, 0 = safe) |
| Visualization | Plotted results using matplotlib |
| Saving Outputs | Exported binary mask as a .png image for downstream use |

# 🛠️ Project Workflow

1. **Load the Image**

from PIL import Image

import numpy as np

image = Image.open('sd-3layers.jpg')

image\_np = np.array(image)

1. Extract Terrain Features

altitude = image\_np[:, :, 0] # Red channel

slope = image\_np[:, :, 1] # Green channel

aspect = image\_np[:, :, 2] # Blue channel

1. Convert Aspect to Degrees

aspect\_deg = (aspect.astype(np.float32) / 255.0) \* 360.0

1. Define Fire-Risk Rules

high\_altitude = altitude > 180

steep\_slope = slope > 160

south\_facing = (aspect\_deg >= 90) & (aspect\_deg <= 180)

fire\_risk\_mask = (high\_altitude & steep\_slope & south\_facing).astype(np.uint8)

1. Visualize the Result

import matplotlib.pyplot as plt

plt.imshow(fire\_risk\_mask, cmap='Reds')

plt.title("Fire Risk Zones")

plt.axis('off')

plt.show()

1. Save the Output

mask\_img = Image.fromarray(fire\_risk\_mask \* 255)

mask\_img.save('fire\_risk\_mask.png')

# 📊 Exploratory Data Analysis (EDA)

We explored the image data with histograms to understand:

* Altitude distribution (elevation zones)
* Slope steepness
* Aspect (direction terrain faces)

This helped inform the thresholds used for classification.

# ✅ Output

|  |  |
| --- | --- |
| File | Description |
| fire\_risk\_mask.png | Binary image: 1 = fire-risk, 0 = safe |

# 📈 Results

The model correctly highlights:

* Areas with steep, high terrain
* South-facing slopes (more prone to dryness)
* These are known to be high fire-risk areas

The approach is explainable and easily adjustable by changing threshold values.

# 🚀 Potential Extensions

|  |  |
| --- | --- |
| Idea | Description |
| Add more WIFIRE images | To expand the dataset |
| Train ML classifiers | Logistic Regression, Random Forest (using terrain features) |
| Use deep learning (U-Net) | For full image segmentation |
| Include time-series data | To model fire spread patterns |
| Incorporate weather variables | Like temperature, wind, or humidity |

# 📚 Requirements

* Python 3.x
* numpy
* matplotlib
* Pillow (PIL)

Install via:

pip install numpy matplotlib pillow

# 💡 Conclusion

This project shows that even with **just NumPy and a single image**, it's possible to:

* Perform spatial terrain analysis
* Derive useful wildfire risk insights
* Generate training data for future models