

FIRENET: Fire Incident Risk Evaluation & Networked Early Tracking

Karl Jasper Parthier¹ Joris Parthier²

¹karl.jasper.parthier@kwr-hannover.eu

²joris.parthier@kwr-hannover.eu

Abstract

Early fire detection is critical to minimizing casualties, property loss, and environmental damage. Traditional camera-based systems in Germany provide continuous monitoring and faster response but are costly and rely on existing infrastructure, limiting feasibility in remote areas or forests with small wildfires (1 ha), therefore we present an economically viable approach that combines historical wildfire data with environmental and anthropogenic features using machine learning techniques such as Gradient Boosting to identify fire-prone areas. Within these areas, portable edge devices equipped with YOLOv8-based neural network models can be deployed to monitor smaller regions (200 m), leveraging natural topographical features such as hills or tall trees for their installation. In the event of a fire, the system can automatically transmit alerts to operators or fire authorities, reducing response times compared to traditional methods. Using the state of Thuringia, Germany, as a case study, we first demonstrate the feasibility of detecting historical wildfire patterns using Gradient Boosting. Separately, we developed and trained a YOLOv8 model on a custom dataset optimized for real-time wildfire detection on edge devices.

Keywords: Early fire detection, Gradient Boosting, YOLOv8, Edge devices, Wildfire monitoring

Contents

1	Introduction	2
2	Related Work	3
3	Wildfire Susceptibility Mapping	4
3.1	Data	4
3.1.1	Forest fire data	4
3.2	Predictor variables	5
3.2.1	Meteorology	5
3.2.2	Vegetation	5
3.2.3	Topography	6
3.2.4	Anthropogenic influences	6
3.3	Preprocessing	6
3.3.1	Spatial Stratified Random Sampling	7
3.4	Model Setup	8
3.4.1	Results	8
4	Real Time Wildfire Smoke Detection	9
4.1	Data	9
4.2	Data Preparation	9
4.3	Model Architecture	10
4.3.1	Backbone	10
4.3.2	Neck	10
4.3.3	Head	11
4.4	Loss	12
4.5	Evaluation Metrics	12
4.5.1	Recall	12

4.5.2	Precision	12
4.5.3	Intersection over Union (IoU)	12
4.5.4	Mean Average Precision (mAP)[25]	12
4.6	Experiments	13
4.7	Training	13
4.8	Experiments Results	13
5	Discussion	14
6	Conclusion	14
7	Acknowledgement	15

1 Introduction

Wildfires are a natural ecological process and have long played an essential role in many ecosystems [35]. In Germany, however, their occurrence poses an increasing challenge. In 2024, more than 563 wildfires were recorded, burning a total of over 334 ha. Eastern Germany is particularly affected, accounting for 377 fires and 307 ha of burned area. The state of Thuringia experienced 33 wildfires, which collectively burned 35 ha, making it the third most affected federal state, behind Saxony-Anhalt and Brandenburg [33]. Brandenburg is an exceptional case in Germany's wildfire landscape, recording both the highest number of fires (197 in 2024) and the largest burned area (223 ha) [33]. Its high susceptibility results from sandy-loamy soils, extensive pine monocultures, dry summers, and the close proximity of forests to human activity. Human activity remains the primary driver of wildfire ignition in Germany [23]. Although the frequency and magnitude of wildfires in Germany have increased over the last decade 1

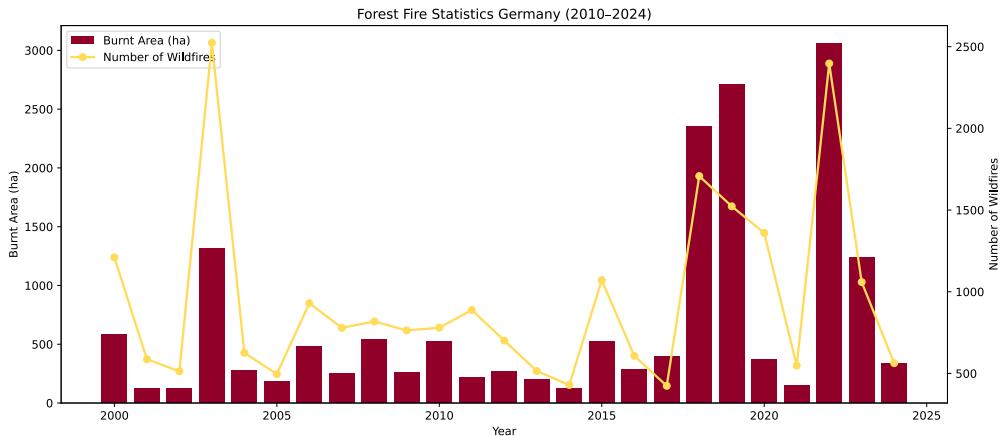


Figure 1: Forest fire statistics [33]

,the overall scale remains significantly lower than in high-risk regions such as California. While the average burned area per wildfire in Germany is around 0.59 ha, California reports values exceeding 130 ha per incident [7]. Due to the strongly right-skewed distribution in both regions, most fires are relatively small, but rare extreme events increase the average. This illustrates the need for systems specifically designed for Germany, as the wildfire landscape there differs significantly from other countries, such as the U.S., and even within Germany, wildfire susceptibility varies widely.

One system that has seen rapid adoption is camera-based wildfire detection, which, unlike traditional emergency reporting, is often hindered by inaccurate information about a fire's location, direction, or type (U. Cimolino, Chair of AK Waldbrand, personal conversation, 30.08.2025), and provides continuous, effective monitoring across vast areas. In Germany, one of the most widely adopted solutions is the Automatisiertes Waldbrand-Frühwarnsystem (AWFS), which uses optical sensors to monitor forests in real time. First deployed in Brandenburg in 2003, AWFS has since expanded to eight federal states: Brandenburg, Mecklenburg-Vorpommern, Saxony, Saxony-Anhalt, Lower Saxony, Berlin, North Rhine-Westphalia, and Hesse [26]. Other approaches, such as gas-sensor networks [44], have been piloted, while

satellite-based remote sensing has seen limited adoption due to insufficient spatial resolution for detecting the relatively small wildfires common in Germany (U. Cimolino, personal conversation, 30.08.2025). Despite these technological advances, early detection strategies remain highly heterogeneous across the country. Brandenburg, for instance, combines manned aerial surveillance with the automated camera-based AWFS system, whereas states like Rhineland-Palatinate rely solely on aerial surveillance, and others maintain no dedicated early detection capability at all. 2

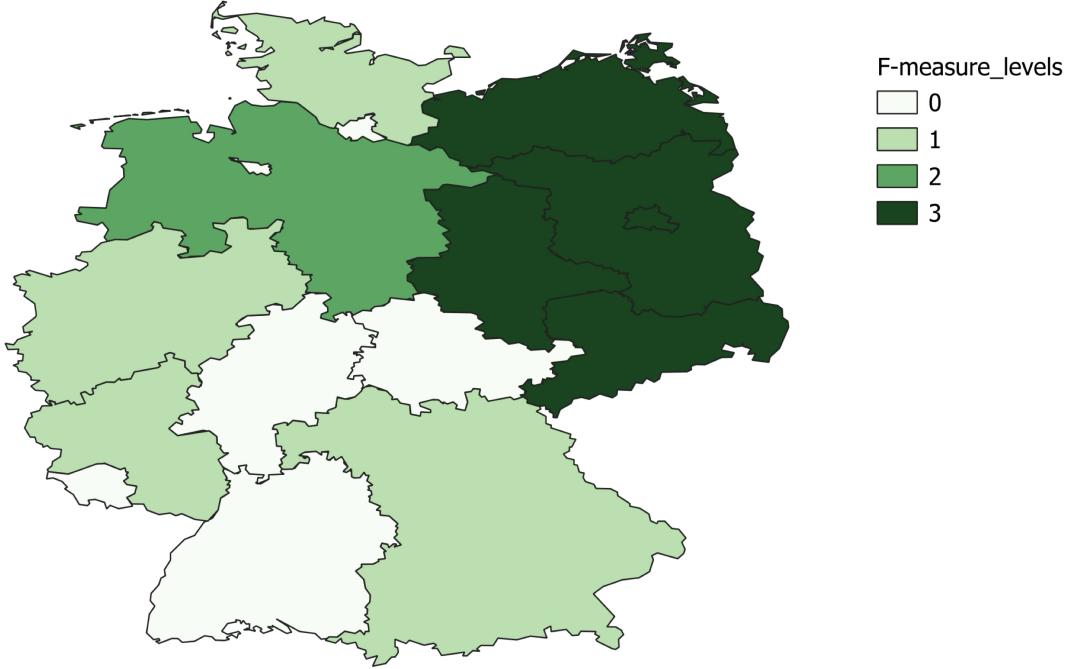


Figure 2: F_Measure Levels² across states. Source: <https://www.kiwuh.de/service/wissenswertes/wissenswertes/waldbrand-frueherkennung>

These differences in adaption reflect both varying wildfire risk levels and the limitations of camera-based systems, which are most effective in regions with large, contiguous forests but less efficient for small, scattered fires [32]. High costs and infrastructure requirements further constrain widespread adoption; for instance, deploying AWFS to cover 320.000 ha in Lower Saxony required an investment of €2.3 million [11].

Current research has largely focused on either wildfire susceptibility modeling or neural-network-based detection using smoke, but rarely combines these approaches or addresses practical challenges.

FIRENET (Fire Incident Risk Evaluation & Networked Early Tracking) addresses this gap by integrating spatial risk modeling with edge-based real-time wildfire detection, offering real-time inference, flexible installation on natural elevations, and the ability to be relocated as high-fire-risk zones shift, making it well-suited for detecting small wildfires in areas where traditional monitoring are impractical 3.

2 Related Work

Our work builds on recent efforts to model wildfire susceptibility in Germany, defined as the estimated likelihood of a forest fire, with particular emphasis on the national-scale study by [40] and the regional analysis of north-eastern Germany by [23]. While these studies demonstrate that machine learning techniques such as Random Forest can capture patterns in fire susceptibility, with reported model accuracies of 89% and 69–71%, respectively, they are both limited by their comparatively coarser spatial

²F_Measure categories (as of 2022): 0: No wildfire early detection. 1: Detection relies solely on manual methods (watchtowers, aerial surveillance). 2: Detection using AWFS. 3: Detection using a combination of AWFS and manual measures.

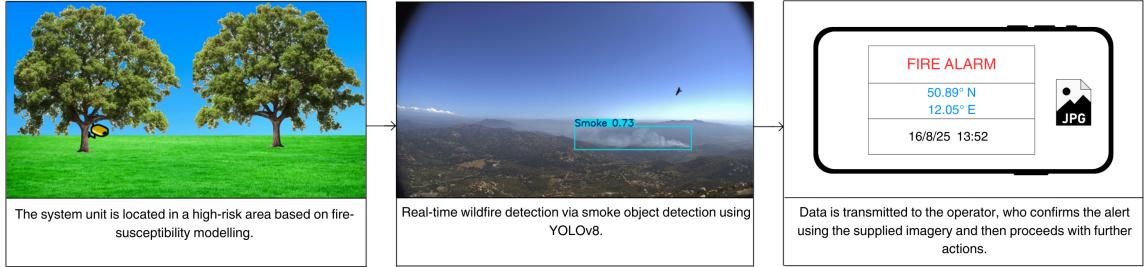


Figure 3: FIRENET’s approach (Images sourced from FIgLib and from Creative Commons, the only modifications to the images are cropping and resizing).

resolutions. For instance, the German study [40] used kilometer-level resolution, while the Brandenburg study [23] employed a 50 m grid. In contrast, we employ a substantially finer spatial resolution of 10 m. This choice is motivated by two factors. First, a fine resolution is necessary to match the limited sensing range of our edge-device deployment. Second, our modeling framework focuses specifically on potential ignition points, rather than all pixels affected by fire events. Because ignition is a prerequisite for subsequent fire spread, accurately identifying ignition-prone locations is both computationally efficient and directly aligned with early-warning objectives. In terms of predictors, our selection of features follows the general approach established in previous susceptibility studies [23, 40]. Based on findings from related susceptibility-mapping work by Akinci et al. [5], we employ Gradient Boosted Trees, specifically LightGBM (Light Gradient Boosting Machine) instead of Random Forest, as Gradient Boosted Trees outperformed Random Forest in their comparative evaluation. Furthermore, we apply K-means clustering to reduce sampling bias when selecting negative examples.

Concerning the algorithm used for smoke detection, research prior to the deep-learning era relied on hand-crafted features, often utilizing color cues and motion analysis in video sequences to identify potential smoke regions [8]. With the rise of deep learning, convolutional neural networks (CNNs) have achieved significant success in smoke detection. Hohberg [22] demonstrated how CNN-based methods can be applied to wildfire smoke detection. In addition, object-detection frameworks based on YOLO [37], such as YOLOv5s, have been successfully applied to smoke-detection tasks [47].

Other studies have explored more sophisticated network designs. For instance, SmokeyNet [12] integrates CNNs, LSTMs, and Vision Transformers (ViTs) to enable real-time detection of wildland fire smoke while also introducing the FIgLib dataset for model training and evaluation. This paper motivated our use of HPWREN Fire Ignition images (FIgLib) as input for our neural-network models. Our choice of the YOLOv8 architecture was driven by its recent success in object detection, widespread adoption, ease of implementation, and the demonstrated performance of earlier YOLO versions on similar smoke-detection tasks [47].

3 Wildfire Susceptibility Mapping

This section addresses the following research questions:

1. Which environmental and anthropogenic variables most strongly influence forest-fire ignition within the federal state of Thuringia?
2. Which specific areas exhibit the highest susceptibility to forest fires?

3.1 Data

3.1.1 Forest fire data

To model wildfire susceptibility (WFS) in Thuringia, both forest-fire records and relevant predictor datasets were collected and pre-processed. Statistical and geospatial information on forest fires were obtained from the [42] and the [41]. Two datasets were provided, covering the periods 2015–2023 and 2024–2025, respectively. The dataset for 2015–2023 included the following attributes: date, Thuringian Forest Office, district, compartment, forest site, and burned area (ha). The dataset for 2024–2025 additionally provided date and time information, as well as x - and y -coordinates of wildfire occurrences.

Since only the second dataset included x and y coordinates, while the first dataset provided only the forest site as the location, a dataset containing the geospatial coordinates of the forest sites was requested from the forestry office [42]. The x and y coordinates of the fire occurrences were then assigned to the centroid of the corresponding forest sites.

3.2 Predictor variables

To model WFS, a total of 27 predictor variables were analyzed. The predictor categories were largely adopted from [16], [40], and [23], and encompass meteorological, vegetation-related, topographic, and anthropogenic parameters. In the following sections, the predictor variables are presented in greater detail.

Category	Feature Name	Data Source
Meteorology	temperature_2m_mean(max/min)_90	[21] via [3]
Meteorology	shortwave_radiation_sum_90	[21] via [3]
Meteorology	precipitation_sum_90	[21] via [3]
Meteorology	dew_point_2m_mean_90	[21] via [3]
Meteorology	relative_humidity_2m_mean_90	[21] via [3]
Meteorology	surface_pressure_mean_90	[21] via [3]
Meteorology	soil_moisture_0_to_7cm_mean_90	[21] via [3]
Meteorology	vapour_pressure_deficit_max_90	[21] via [3]
Meteorology	soil_temperature_0_to_7cm_mean_90	[21] via [3]
Meteorology	wind_gusts_10m_mean_90	[21] via [3]
Meteorology	wind_speed_10m_mean_90	[21] via [3]
Vegetation	tree_cover_density	[10]
Vegetation	forestedge	[10]
Vegetation	canopy_height	[29]
Vegetation	forest_class_broadleaved_forest	[10]
Vegetation	forest_class_coniferous_forest	[10]
Topographic	slope	[20]
Topographic	aspect	[20]
Topographic	dem	[20]
Topographic	TWI	Derived from DEM
Topographic	TRI	Derived from DEM
Anthropogenic	road_dist (m)	[34] via [17]
Anthropogenic	camp_site_dist (m)	[34] via [17]
Anthropogenic	dist_to_urban (m)	[34] via [17]
Anthropogenic	dist_to_railways (m)	[34] via [17]
Anthropogenic	dist_to_waterbodies (m)	[34] via [17]
Anthropogenic	dist_to_waterways (m)	[34] via [17]

Table 1: Predictor variables considered in the study.

3.2.1 Meteorology

Meteorological data were retrieved from the Historical Weather API by Open-Meteo and aggregated over three-month periods, following [19] to incorporate meteorological conditions preceding forest fires. The API is based on ERA5-Land hourly data from 1950 to the present, provided by the Copernicus Climate Data Store. Additional climatic variables, such as wind speed and solar radiation, can also influence fire occurrence [4]. Lightning strikes were excluded in this study.

3.2.2 Vegetation

Vegetation data layers, including tree cover density and forest type, were obtained from the WEFEO data platform. Based on the tree cover density layer, the distance to the forest edge was calculated using the GDAL “proximity” tool, with areas exhibiting a tree cover density of 50% or less classified as non-forested. Canopy height data were retrieved from the Global Canopy Height 2020, which utilizes a

probabilistic deep learning model that integrates GEDI LiDAR observations with Sentinel-2 imagery to provide global estimates of canopy height from satellite data [29].

3.2.3 Topography

The Digital Elevation Model (DEM) for Germany, obtained from the Continental Europe Digital Terrain Model via OpenTopography, was utilized in conjunction with GDAL software to derive additional terrain attributes, including slope, aspect, and the Terrain Ruggedness Index (TRI). [9] conducted a comprehensive analysis, demonstrating that slope, elevation, aspect, and the Topographic Wetness Index (TWI), and by extension the terrain ruggedness index, are among the most frequently employed topographic parameters in environmental and hydrological studies.

For the calculation of TWI, the following formula was adopted [6]:

$$\text{TWI} = \ln \left(\frac{A_{\text{upslope}} \cdot (\text{pixel size})^2}{\tan(\theta_{\text{radians}})} \right) \quad (1)$$

where A_{upslope} represents the upslope contributing area per pixel, pixel size is the linear dimension of a DEM pixel, and θ_{radians} is the local slope expressed in radians.

3.2.4 Anthropogenic influences

Finally, anthropogenic factors have been shown to influence the occurrence of historical forest fires in Germany (Gnilke and Sanders, 2021). To quantify the spatial relationship between these factors and fire events, the GDAL “Distance to Nearest Hub” tool was employed to calculate distances to roads, railways, settlements, camp sites and waterbodies. Shapefiles for roads and railways were obtained from [34] via [17]).

3.3 Preprocessing

The open-source software QGIS 3.40 Bratislava was employed for the processing, analysis, and visualization of geospatial data. Python 3.14, together with the packages Pandas, Numpy, and Matplotlib, was used to manage and analyze tabular data. Machine learning workflows, including data splitting and model evaluation, were conducted using Scikit-learn functions such as `train_test_split`, `StratifiedKFold`, and `RandomizedSearchCV`, while gradient boosting was performed with LightGBM. An overview of the main data processing steps is provided in the following figure. First, the tabular meteorological data were acquired from the Open-Meteo API and appended to the ignition points. The ignition points were then converted to vector data, and the distances to anthropogenic features, also represented as vector data, were calculated and added to each ignition point. The raster layers were downloaded and preprocessed. Preprocessing included projecting all data to a common coordinate reference system (EPSG: 25832) and resampling to a spatial resolution of 10m, using bilinear interpolation for numeric variables and nearest-neighbor interpolation for categorical variables. At this stage, the ignition points were positioned on top of all relevant raster layers, whose values for that specific point could then be extracted using QGIS’s Sample Raster Values tool.

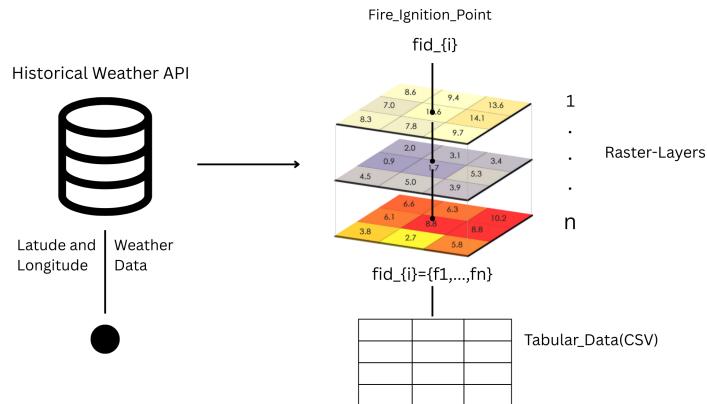


Figure 4: Own illustration showing the data gathering process

3.3.1 Spatial Stratified Random Sampling

To establish meaningful relationships between predictor variables and the occurrence of fire, it is necessary to obtain information not only about pixels that experienced fire but also about those that did not. Therefore, we firstly investigate the spatial distribution of wildfire ignition points using the Average Nearest Neighbor Index (ANNI). Hereby, we ensured that negative samples are drawn from within fire-prone areas rather than far away from them (which random sampling would do), so that the model learns the actual relationship between features and the label, and not just cluster distance. The Average Nearest Neighbor Ratio is defined as:

$$\text{ANN} = \frac{D_o}{D_e}, \quad (2)$$

where D_o denotes the observed mean distance between each feature and its nearest neighbor, and D_e represents the expected mean distance under a random spatial distribution. The statistical significance of the observed spatial pattern was assessed using the corresponding z -score:

$$z = \frac{D_o - D_e}{SE}, \quad (3)$$

where SE is the standard error. For the Thuringia wildfire dataset, the analysis yielded a z -score of -15.36 and a nearest neighbor index of 0.519 , indicating a statistically significant clustered spatial pattern. For further methodological details, refer to [13] and [39].

To ensure that only relevant non-fire pixels are generated, we implemented a spatially stratified random sampling approach. First, we applied K-Means to the coordinates of the wildfire ignition points, calculating distances with the Euclidean metric (which is appropriate given EPSG:25832). To determine the appropriate number of clusters (k) for our data, we applied the elbow method. First, a set of candidate k values was defined. Then, the within-cluster sum of squares (WCSS), formally defined as

$$\text{WCSS} = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2, \quad (4)$$

where

$$\begin{aligned} k &= \text{number of clusters,} \\ C_i &= \text{points in cluster } i, \\ x &= \text{a point in } C_i, \\ \mu_i &= \text{cluster centroid,} \\ \|\cdot\| &= \text{Euclidean distance,} \end{aligned}$$

was calculated.

WCSS, which is the sum of the squared distances from each data point to its cluster centroid, was then plotted against the number of clusters k on the x-axis. The point on the plot where the line bends sharply resembles an elbow, indicating where adding more clusters does not significantly reduce the WCSS. Following this methodology, we found $k = 4$. Drawing on the clustering results, the distinctive groups were polygonized using Geos' convex hulls (Figure 5).

Based on these convex hulls, random sampling was performed. Before sampling, the different strata (polygons) were filtered to include only areas with tree density greater than 50%. To ensure that no fires were accidentally classified as non-fires, we assumed a circular fire spread and calculated the corresponding radius, applying a safety margin by doubling it. Additionally, all entries with null values, where site information was available, were excluded. Since it was not possible to obtain information about areas managed by the Bundesforst, all locations owned by them were also filtered out using the Waldflaeche dataset via the ThüringenViewer Dataplatform [1]. To perform random sampling on the fully filtered polygons, we employed a program that, for each point in a filtered group, would select another random point from the same group with the same date. The resulting non-fire points were then combined with the fire points to complete the set of training points. For this purpose, the training points were assigned to the respective classes of "fire" and "non-fire". This data frame formed the basis for training Gradient Boosted models to predict WFS. By applying the sampling method described above, we provided the model with a balanced dataset. In total, the model was trained on 477 points, comprising 224 positive and 223 negative examples

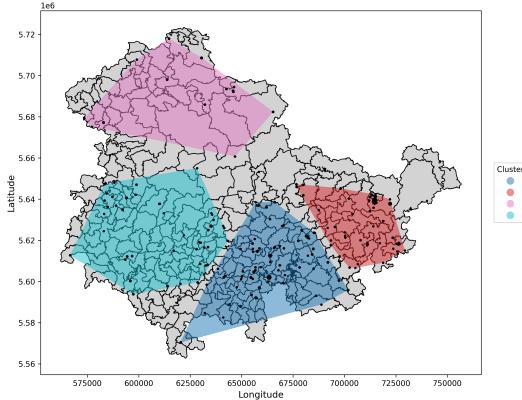


Figure 5: Strata based on KMeans

3.4 Model Setup

To assess WFS in Thuringia, a gradient-boosted classifier was used. The model employed binary classes (fire vs. non-fire) to predict current and future WFS. Gradient boosting is a widely used machine-learning algorithm in forestry and remote sensing applications [38] and [28]. The gradient-boosted algorithm is based on the boosting approach developed by [14] and later generalized by [15]. It constructs an ensemble of decision trees sequentially, with each new tree trained on the residual errors (negative gradients) of the previous ensemble. The probability score of a pixel being predicted as a fire pixel represents its susceptibility to a forest fire. The gradient-boosted classifier used is LightGBM, introduced by Microsoft [27]. Alternatively, XGBoost is often employed in research and is extremely similar.

3.4.1 Results

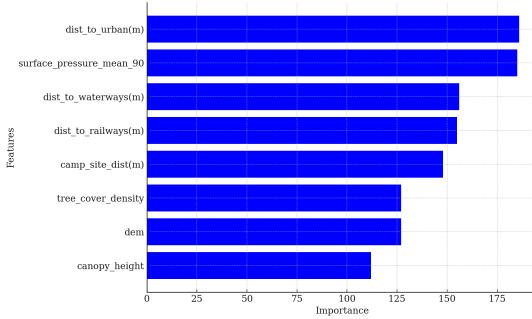


Figure 6: Feature importance

Based on these findings a final gradient bossting model was built only includng all relvent features. To optimize model performance, a GridSearchCV with 4-fold cross-validation on the training set was conducted. The final model reached 0.674 accuracy on the training set and 0.66 accuracy on the test set.

Parameter	Best value
learning_rate	0.05
max_depth	2
min_child_samples	3
n_estimators	20

Table 2: Hyperparameter settings. For a detailed explanation of the parameters, see <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>

The now fully optimized model was used to predict the wildfire susceptibility of the forest site Ernsee in Thuringia. The results are shown below.

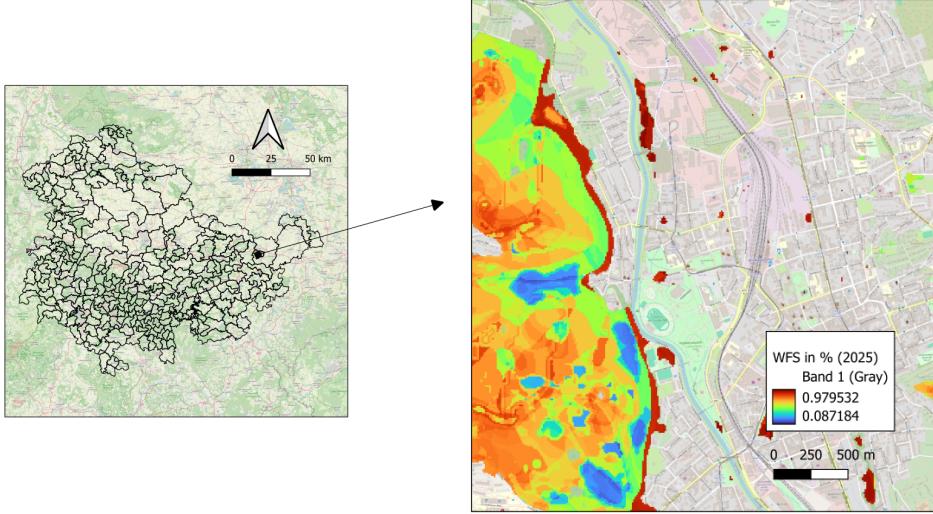


Figure 7: Detailed maps of WFS in the forest site of Ernsee (Weida) in the federal state of Thuringia

4 Real Time Wildfire Smoke Detection

4.1 Data

The data used in this study is the Fire Ignition Image Library (FIgLib), provided by the High Performance Wireless Research and Education Network (HPWREN) and first introduced by Dewangan’s 2022 paper [12]. FIlG is a publicly available benchmark dataset for wildfire smoke detection, based on imagery from fixed-view cameras and, more recently, PTZ public-safety cameras located on remote mountain tops in Southern California. As of the end of 2024, the network included over 475 cameras [24] and currently contains 494 unique image sequences (3072×2048 pixels), typically spanning 20 seconds and consisting of 81 images, covering the period from 40 minutes before fire ignition to 40 minutes after, with each image representing a 60-second interval <https://cdn.hpwren.ucsd.edu/HPWREN-FIgLib-Data/index.html>, (accessed 27 November 2025). Our dataset builds upon FIgLib by selecting representative images from each fire event, with those captured approximately 24 minutes after ignition used as positive examples and those captured about 9 minutes before ignition used as negative examples.

4.2 Data Preparation

The number of fires and images used in the training, validation, and test splits for our dataset (using an 80/10/10 split) are shown in Table 3. We performed a fire-based split, selecting the negative and positive examples for each fire event into the same set as this approach prevents data leakage between splits and ensures that test-set performance is not overstated, in contrast to an image-based split. Some fires in FIgLib were deemed inappropriate because they were not suited to our setup. These included fires recorded at night, sequences captured in black and white, and cases with questionable smoke presence. The exact list of fires omitted and included in the different splits can be accessed at this link: <https://github.com/KarlJasper95/FIRENET/tree/main/Fire%20Preparation>, (accessed 27 November 2025).

Table 3: Train/Val/Test Splits of the FIgLib dataset

Split	# Fires	# Images
Train	316	632
Validation	40	80
Test	40	80
Omitted	98	196
Total	494	988

Further noteworthy is that the dataset includes 409 positive examples and 383 negative examples, this stems from the fact that some negative examples from a given fire overlap with other fires, or because a fire sequence begins only after ignition. In such cases, the affected examples were treated as positives, as we did

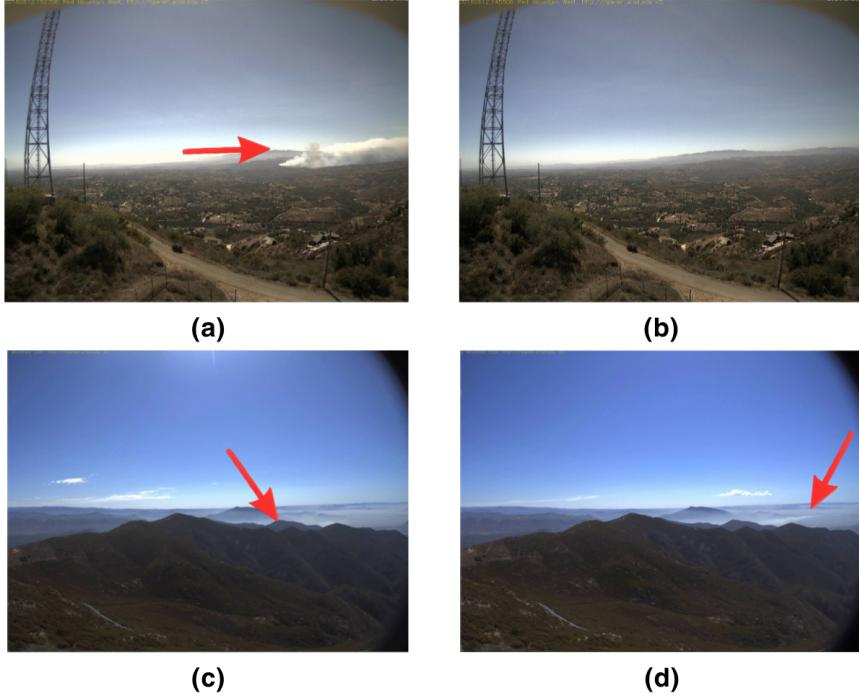


Figure 8: Example images from the FIgLib dataset[12]: (a) clear example of wildfire smoke, (b) easily identifiable non-smoke example, (c) haze without wildfire smoke, (d) confusing fire smoke and haze

not want to omit these fire events entirely. In addition, to the labels FIRESMOKE /NONEFIRESMOKE for each image, the smoke for 409 fires has been manually annotated with bounding boxes. The whole dataset containing training, validation and test folder with the corresponding images, labels and bounding boxes can be accessed via this link <https://www.kaggle.com/datasets/karljasper/firenet-data>, (accessed 27 November 2025).

4.3 Model Architecture

YOLOv8 is an iteration of models belonging to the You Only Look Once (YOLO) [37] family of object detection models, first developed and released by the Ultralytics team in January 2023. YOLO performs object detection by predicting bounding box coordinates and their corresponding class labels in a single forward pass of the network. The model architecture is flexible, allowing a diverse range of tasks, such as object detection, object tracking, and instance segmentation. YOLO is available in different size categories, ranging from nano (n) with 3.2 million parameters to extra-large (x) with 68.2 million parameters [43]. A parameter refers to the learnable weights and biases that the model must optimize. Each version follows the same underlying structure and can be broken down into three parts: the backbone, neck and head (see Figure 9).

4.3.1 Backbone

The backbone is responsible for extracting relevant features from the input image. Structurally, the backbone used in YOLOv8 is similar to the Darknet-53 convolutional neural network first introduced in YOLOv3 [36]. Different is the use of Cross-Stage Partial (CSP) connections, which divide the feature map, the set of features detected by the convolutional kernels, into two branches: one branch is passed through a series of bottleneck layers, while the other is forwarded directly. The two branches are then concatenated, allowing the network to leverage both deep semantic features and low-level spatial information.

4.3.2 Neck

The Neck connects the features extracted by the Backbone with the Head. Its main function is multi-scale feature fusion, which involves combining features from different layers of the Backbone so that the

network can accurately detect objects of all sizes. It achieves this by building an enhanced feature pyramid, merging high-resolution spatial details from earlier layers with the deeper, more abstract semantic features from later layers. Important modules here are the SPPF (Spatial Pyramid Pooling Fast), which is used to create a multi-scale representation and a large receptive field on the deepest features, and the C2f module, which serves as the efficient building block for transforming and combining the semantic and spatial features during the feature pyramid construction.

4.3.3 Head

The Head generates the final predictions, which include the bounding box coordinates, object confidence scores (indicating whether an object exists at that location), and class labels. In total, the Head contains three detection heads that are responsible for generating these final predictions for small, medium, and large objects respectively [46].

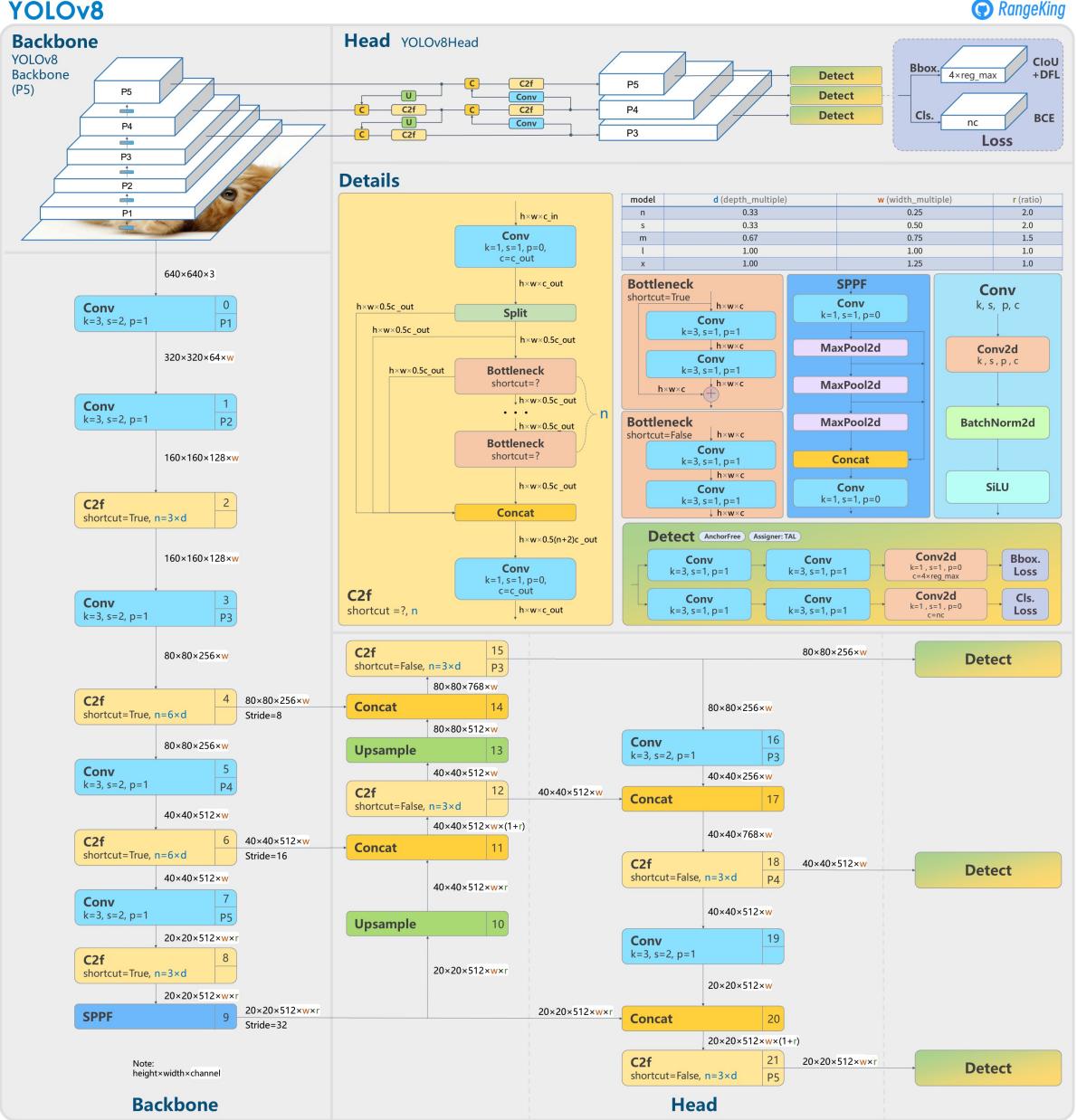


Figure 9: Overview of YOLOv8 architecture by GitHub user Range King.

4.4 Loss

YOLOv8 uses an advanced loss function for the detection task. During training, the gradients of this loss are used to iteratively update the model's weights and biases. Since the model predicts both bounding boxes and class probabilities, the loss function must account for both aspects. The total loss is defined as:

$$L_{\text{total}} = L_{\text{box}} + L_{\text{cls}} + L_{\text{dfl}} \quad (5)$$

where:

- L_{box} : Refers to the loss used for the bounding box coordinates and, among other factors, incorporates IoU (Intersection over Union) loss [48].
- L_{cls} : Refers to the loss associated with the class prediction, based on binary cross-entropy [18].
- L_{dfl} : This additional term helps the model generate more precise bounding box edges by predicting them as discrete probability distributions instead of single values [30].

4.5 Evaluation Metrics

Term	Description
TP	Correctly detected fire/smoke instance
TN	Correctly identified non-smoke example
FP	Predicted a non-smoke example as smoke
FN	Predicted a fire/smoke image as non-fire/smoke

Table 4: Explanation of terms used in model evaluation.

4.5.1 Recall

Recall measures the proportion of correctly detected fire/smoke instances out of all actual fire/smoke instances. A high recall means the model detects most of the objects, resulting in fewer false negatives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4.5.2 Precision

Precision measures the proportion of correct detections out of all detections made. High precision indicates fewer false positives and higher quality predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

4.5.3 Intersection over Union (IoU)

IoU quantifies the degree of overlap between the predicted bounding box and the actual bounding box. It ranges from 0 to 1.

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

4.5.4 Mean Average Precision (mAP)[25]

The mAP (mean Average Precision) metric is based on the Precision–Recall (PR) curve. The Average Precision (AP) for a single class is defined as the area under the PR curve, which quantifies how well the model detects objects and correctly classifies them across different confidence thresholds:

$$AP_c = \int_0^1 p_c(r) dr$$

where $p_c(r)$ denotes the precision as a function of recall r . In practice, this integral is approximated using discrete recall thresholds.

The mean Average Precision (mAP) is computed by averaging the AP values across all classes:

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^C AP_c$$

mAP is typically calculated at different IoU thresholds. mAP@0.5 considers only detections with an IoU of 0.5 as true positives, while a more rigorous metric, mAP@50:95, evaluates IoU thresholds from 0.5 to 0.95 with a step size of 0.05.

4.6 Experiments

We experimented with different types of transfer learning, as our dataset was too small to train a model from scratch. For transfer learning, we used pretrained weights from the YOLO models by Ultralytics, which were trained on the COCO dataset, a widely used benchmark for object detection [31]. Transfer learning is based on the idea that the information learned by the same network on a different task can be transferred to the detection task at hand. In general, there are two common approaches: The first is weight initialization, where the pretrained weights and biases are used as a starting point instead of random values. The second is layer freezing, in which certain layers of the network are kept frozen, meaning their weights are not updated during training. This approach leverages the idea that the network has already captured useful feature representations, allowing the model to focus on learning how to map these features to the new task. Since we plan to run our models on edge devices, only the nano and small versions of YOLOv8 (YOLOv8n and YOLOv8s) were used. We experimented with different degrees of freezing, ranging from no_fr (no freezing) to fr_10 (10 layers frozen) or fr_20 (20 layers frozen).

4.7 Training

Models were trained for 50 epochs with a patience of 10 (training was stopped if the validation loss did not improve for 10 consecutive epochs). A batch size of 8 was used, and images were resized to 640×640 (img_size = 640). The AdamW optimizer was employed, and data augmentation was applied using horizontal flipping and subtle color perturbations and other [2]. Model training was performed on a rented high-performance computing server utilizing an NVIDIA A100 GPU. For better understanding of the training process, Figures 10, 11, and 12 show comparisons between each YOLO variant under box loss, classification loss, and mAP@50.

4.8 Experiments Results

Table 5 reports Precision, Recall, mAP@50, and mAP@50:95 for different transfer learning strategies and YOLOv8 model sizes. Overall, the results indicate that models trained without freezing layers achieve the highest performance. The nano model without freezing (Yolov8n_no_fr) achieved a Precision of 0.732 and mAP@50 of 0.638, while the small model without freezing (Yolov8s_no_fr) reached the highest Precision of 0.801 and a comparable mAP@50 of 0.636. Freezing layers generally reduced both precision and recall, with deeper freezing (20 layers) leading to the lowest performance across all metrics. It is also worth noting the difference between the metrics mAP@50 and mAP@50:95, which suggests that the model needs improvement in generating accurate bounding boxes. This is further supported by the box loss and the mAP@50 during training, which steadily decline/increase and fail to exhibit the characteristic of a rapid rise/rapid fall followed by stabilization typically seen in well-performing models. A video of the model’s performance on selected test examples can be viewed at this link: <https://youtu.be/CL2PnPmowlM>, (accessed 30 November 2025).

Table 5: Comparison YOLOv8 Variants on test set

Model	Variant	Precision	Recall	mAP50	mAP50:95
YOLOv8n	no_fr	0.732	0.595	0.638	0.336
	fr_10	0.599	0.497	0.515	0.290
	fr_20	0.470	0.190	0.180	0.0877
YOLOv8s	no_fr	0.801	0.573	0.636	0.317
	fr_10	0.611	0.500	0.479	0.226
	fr_20	0.568	0.286	0.297	0.158

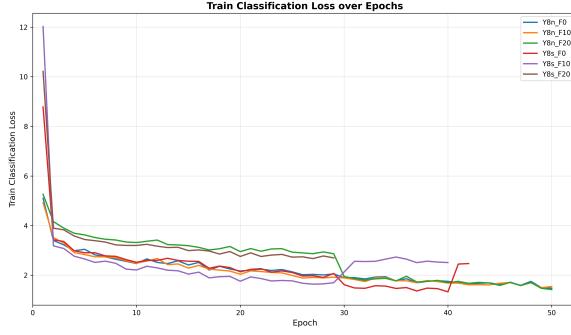


Figure 10: Train Classification Loss - YOLO variants

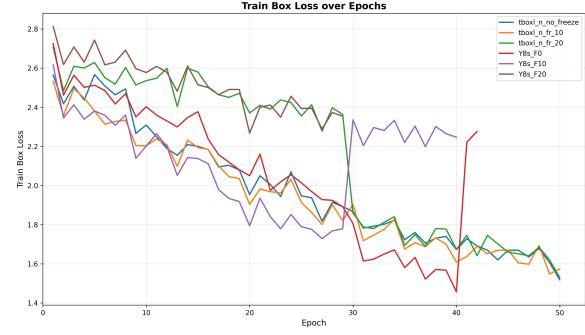


Figure 11: Train Box Loss - YOLO variants

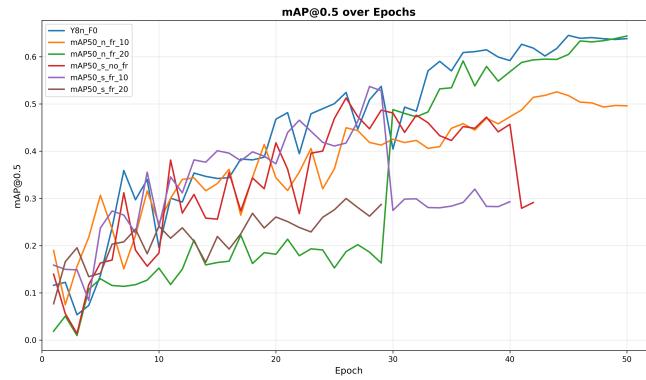


Figure 12: mAP@50 - YOLO variants

5 Discussion

Our work on the WFS maps of Thuringia shows that this mapping is effective even at lower resolution, confirms that human activity and surface air pressure are major factors in susceptibility. In future work, we will implement viewshed analysis to identify positions from which high-susceptibility zones are visible, guiding the placement of edge devices. We also plan to train the model with data from Saxony and potentially Rhineland-Palatinate to increase robustness. Additionally, combining susceptibility mapping with fire spread analysis could quantify how ignition at one location may cause more damage than at another, thus improving the modeling.

The results of the wildfire smoke detection are very promising (see <https://youtu.be/CL2PnGmowlM>). A real smoke trial in the Harz region is currently being planned to evaluate how well the trained model transfers to German conditions, in cooperation with local authorities. Regarding model improvement, using alternative pretrained weights instead of COCO may be beneficial. Better-suited datasets such as FASDD (Flame And Smoke Detection) [45], could be considered. For fine-tuning, multiple images from a single fire sequence in FIgLib could be used. Since no openly available wildfire detection datasets like FIgLib exist for Germany, images collected from webcams in Germany can be augmented, and Blender will be used to generate artificial positive examples specifically for these webcam image sequences.

6 Conclusion

FIRENET addresses the need for a Germany-specific early detection system that is practical for monitoring small fire events (one hectare or less). In terms of practical implementation, we demonstrated how wildfire susceptibility can be mapped to identify fire-prone areas in Thuringia, achieving an accuracy of 0.66. Additionally, we showed that a YOLO-based neural network could be effective 5, at least in Southern California and identified the most suitable YOLOv8 variant for our specific requirements.

7 Acknowledgement

We thank Dr. Lutz Wiehlmann for his guidance and supervision, and Dr. Monika Sester for the opportunity to complete our internship at the Institut für Kartographie und Geoinformatik, Leibniz Universität Hannover. Special thanks to Jens Golze and Frank Thiemann for their support with geodata interpretation, and to Jörg Thiel and Heiko Hirschfeld for providing essential forestry data. Last but not least, we are grateful to Ulrich Cimolino for sharing his invaluable insights on wildfire management.

References

- [1] Thüringen viewer – geodateninfrastruktur thüringen (gdi-th). <https://thueringenviewer.thueringen.de/thviewer3/index.html#LayerInfoDataDownload>. Accessed: 2025-11-30.
- [2] Yolo data augmentation. <https://docs.ultralytics.com/guides/yolo-data-augmentation>. Accessed: 2025-11-30.
- [3] Open-meteo historical weather api. <https://openmeteo.com/en/docs/historical-weather-api>, 2025. Accessed: 30 Nov 2025.
- [4] A. Abdollahi and B. Pradhan. Explainable artificial intelligence (xai) for interpreting the contributing factors feed into the wildfire susceptibility prediction model. *Science of The Total Environment*, 879:163004, 2023. doi:10.1016/j.scitotenv.2023.163004.
- [5] Z.A. Akinci, H. Akinci, and M. Zeybek. Comparison of diverse machine learning algorithms for forest fire susceptibility mapping in antalya, türkiye. *Advances in Space Research*, 74(2):647–667, 2024. doi:10.1016/j.asr.2024.04.018. URL <https://doi.org/10.1016/j.asr.2024.04.018>.
- [6] K. J. Beven and M. J. Kirkby. A physically based, variable contributing area model of basin hydrology / un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant. *Hydrological Sciences Bulletin*, 24(1):43–69, 1979. doi:10.1080/02626667909491834.
- [7] California Department of Forestry and Fire Protection. Cal fire — incidents 2024. <https://www.fire.ca.gov/incidents/2024>, 2024. Accessed: 2025-11-30.
- [8] Junzhou Chen and Yong You. Early fire detection using hep and space-time analysis, 2013. URL <https://arxiv.org/abs/1310.1855>. arXiv preprint arXiv:1310.1855.
- [9] S. D. Chicas and J. Østergaard Nielsen. Who are the actors and what are the factors that are used in models to map forest fire susceptibility? a systematic review. *Natural Hazards*, 114:2417–2434, 2022. doi:10.1007/s11069-022-05495-5.
- [10] Copernicus Land Monitoring Service (CLMS). High resolution layer – tree cover and forests (hrl tree cover forests). European Environment Agency / Copernicus, 2025. URL <https://land.copernicus.eu/en/products/high-resolution-layer-forests-and-tree-cover>. Latest version accessed: 30 Nov 2025.
- [11] CZ. Kameras spüren waldbrände auf, 2025. URL <https://www.cz.de/lokales/celle-lk/kameras-spueren-waldbraende-auf-64AEE4643275DBF276E37DAAAA.html>.
- [12] Anshuman Dewangan, Yash Pande, Hans-Werner Braun, Frank Vernon, Ismael Perez, Ilkay Altintas, Garrison W. Cottrell, and Mai H. Nguyen. Figlib & smokeynet: Dataset and deep learning model for real-time wildland fire smoke detection. *Remote Sensing*, 14(4):1007, 2022. doi:10.3390/rs14041007. URL <https://doi.org/10.3390/rs14041007>.
- [13] Esri ArcGIS Pro. How average nearest neighbor (distance) spatial statistic works, 2025. <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-statistics/h-how-average-nearest-neighbor-distance-spatial-st.htm>.
- [14] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi:10.1006/jcss.1997.1504. URL <https://doi.org/10.1006/jcss.1997.1504>.

- [15] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001. doi:10.1214/aos/1013203451.
- [16] A. Gantéaume, A. Camia, M. Jappiot, C. Hernando, M. Guijarro, and C. Lampin. A review of the main driving factors of forest fire ignition over europe. *Environmental Management*, 51(3):651–662, 2013. doi:10.1007/s00267-012-9961-z.
- [17] Geofabrik GmbH. Geofabrik openstreetmap data downloads. <https://www.geofabrik.de/>, 2023. Last accessed: 14 March 2024.
- [18] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009. Chapter 4, p. 120, equation (4.20).
- [19] W. He, S. Shirowzhan, and C. J. Pettit. Gis and machine learning for analysing influencing factors of bushfires using 40-year spatio-temporal bushfire data. *ISPRS International Journal of Geo-Information*, 11(6):336, 2022. doi:10.3390/ijgi11060336.
- [20] Tomislav Hengl, Leandro Leal Parente, Josip Krizan, and Carmelo Bonannella. Continental europe digital terrain model at 30 m resolution based on gedi, icesat-2, aw3d, glo-30, euDEM, merit dem and background layers, 2020. URL <https://doi.org/10.5281/zenodo.4724549>. European Digital Terrain Models (EU DTM).
- [21] H. Hersbach, B. Bell, P. Berrisford, G. Biavati, A. Horányi, J. Muñoz Sabater, J. Nicolas, C. Peubey, R. Radu, I. Rozum, D. Schepers, A. Simmons, C. Soci, D. Dee, and J.-N. Thépaut. Era5 monthly averaged data on single levels from 1940 to present. Copernicus Climate Data Store (CDS), 2019. Accessed: 30 Nov 2025.
- [22] Simon Philipp Hohberg. Wildfire smoke detection using convolutional neural networks, September 2015. URL https://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/Betreute_Arbeiten/Master-Hohberg.pdf. Accessed: 30 November 2025.
- [23] Katharina H. Horn, Stenka Vulova, Hanyu Li, and Birgit Kleinschmit. Modelling current and future forest fire susceptibility in north-eastern germany. *Natural Hazards and Earth System Sciences*, 25:383–401, 2025. doi:10.5194/nhess-25-383-2025. URL <https://nhess.copernicus.org/articles/25/383/2025/>.
- [24] HPWREN. Hpren 2024 year in review, 2025. URL <https://www.hpren.ucsd.edu/news/20250101/>.
- [25] Jonathan Hui. map (mean average precision) for object detection. Medium blog post, 2018. URL <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. Accessed: 2025-12-01.
- [26] IQ-Firewatch. Automatisiertes waldbrand-frühwarnsystem (awfs). <https://www.iq-firewatch.com/de/>, 2025. Accessed: 30 November 2025.
- [27] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc.. isbn:9781510860964.
- [28] Jonathan Koh. Gradient boosting with extreme-value theory for wildfire prediction. *Extremes*, 26(2):273–299, 2023. doi:10.1007/s10687-022-00454-6.
- [29] N. Lang, W. Jetz, K. Schindler, and J. D. Wegner. A high-resolution canopy height model of the earth. *Nature Ecology & Evolution*, 7:1778–1789, 2023. doi:10.1038/s41559-023-02206-6. URL <https://doi.org/10.1038/s41559-023-02206-6>.
- [30] X. Li, C. Lv, W. Wang, G. Li, L. Yang, and J. Yang. Generalized focal loss: Towards efficient representation learning for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3139–3153, March 2023. doi:10.1109/TPAMI.2022.3180392.

- [31] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*, 2014. URL <https://doi.org/10.48550/arXiv.1405.0312>.
- [32] MDR. Frühwarnsysteme für waldbrände in thüringen laut landesforstanstalt unnötig, 2025. URL <https://www.mdr.de/nachrichten/thueringen/waldbraende-warnsystem-automatisch-gefaehr-klima-100.html>.
- [33] Federal Ministry of Food and Agriculture (BMEL). Waldbrandstatistik 2024. Dataset (XLSX) available on the BMEL statistics website, 2024. URL <https://www.bmel-statistik.de/forst-holz/waldbrandstatistik>. Accessed on 30 November 2025.
- [34] OpenStreetMap Contributors. Openstreetmap. <https://www.openstreetmap.org>, 2023. Last accessed: 14 March 2024.
- [35] J.G. Pausas and J.E. Keeley. Wildfires as an ecosystem service. *Frontiers in Ecology and the Environment*, 17:289–295, 2019.
- [36] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. doi:10.48550/arXiv.1804.02767. URL <https://arxiv.org/abs/1804.02767>.
- [37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015. URL <https://arxiv.org/abs/1506.02640>. arXiv preprint arXiv:1506.02640.
- [38] M. Seddouki, M. Benayad, Z. Aamir, M. Tahiri, M. Maanan, and H. Rhinane. Using machine learning coupled with remote sensing for forest fire susceptibility mapping: Case study tetouan province, northern morocco. In *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS-Archives)*, Volume XLVIII-4/W6-2022, pages 333–342, 2023. doi:10.5194/isprs-archives-XLVIII-4-W6-2022-333-2023.
- [39] SEER Lab, University of Florida. Average nearest neighbor analysis, 2025. <https://youtu.be/h0uHbT4cQqo?si=-JSIv1wHNOGvZwyT>.
- [40] B. Thies. Machine learning wildfire susceptibility mapping for germany. *Natural Hazards*, 121:12517–12530, 2025. doi:10.1007/s11069-025-07292-2. URL <https://doi.org/10.1007/s11069-025-07292-2>.
- [41] Thuringia Geoportal. Forest data portal, 2025. <https://geoportal.thueringen.de/themen/forst>.
- [42] Thuringian Forestry Office. Forest fire data and statistics, 2025. <https://www.thueringenforst.de/startseite>.
- [43] Ultralytics, 2025. URL <https://docs.ultralytics.com/de/models/yolov8>. Accessed: 2025-11-30.
- [44] Volksstimme. Pilotprojekt zur waldbrandfrüherkennung gestartet. <https://www.volksstimme.de/panorama/pilotprojekt-zur-waldbrandfruherkennung-gestartet-3560075>, 2024. Accessed: 30 November 2025.
- [45] M. Wang, L. Jiang, P. Yue, D. Yu, and T. Tuo. Fasdd: An open-access 100,000-level flame and smoke detection dataset for deep learning in fire detection. *Earth Syst. Sci. Data Discuss. [preprint]*, 2023. doi:10.5194/essd-2023-73. URL <https://doi.org/10.5194/essd-2023-73>. Received: 24 Feb 2023.
- [46] Muhammad Yaseen. What is yolov8: An in-depth exploration of the internal features of the next-generation object detector, 2024. URL <https://doi.org/10.48550/arXiv.2408.15857>. arXiv:2408.15857, submitted 28 Aug 2024.
- [47] H. Yin, M. Chen, W. Fan, Y. Jin, S. G. Hassan, and S. Liu. Efficient smoke detection based on yolo v5s. *Mathematics*, 10(19):3493, 2022. doi:10.3390/math10193493. URL <https://doi.org/10.3390/math10193493>.

- [48] D. Zhou et al. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, pages 85–94, Quebec City, QC, Canada, 2019. doi:10.1109/3DV.2019.00019.