POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Anomaly Detection from Aerial Images for Search and Rescue Missions

Tesi di Laurea Magistrale in
Computer Science and Engineering

Author: **Luca Morandini**

Student ID: 963897
Advisor: Prof. Piero Fraternali
Co-advisors: Federico Milani
Academic Year: 2021-22

Ringraziamenti

Innanzitutto ringrazio il prof. Piero Fraternali per avermi dato l'opportunità di svolgere questa tesi e per essere sempre stato disponibile con i suoi preziosi consigli.

In particolare vorrei ringraziare anche Federico Milani per avermi guidato durante tutto il percorso con grande disponibilità *(e pazienza!)* e per avermi insegnato davvero tanto. Ringrazio anche Rocio Nahime Torres che ha contribuito allo sviluppo di questo progetto.

Infine, un enorme grazie ai miei genitori che mi hanno sempre sostenuto, anche nei momenti più difficili, e mi hanno dato la possibilità di seguire le mie passioni.

Abstract

The growing number of people carrying out activities in the mountains has led to an increasing number of accidents that occur in inaccessible places. When an injured, ill, or lost person requires the intervention of specialized rescue teams, the efficiency of operations is crucial. Search and Rescue missions are time-critical operations that often occur in complex and challenging environments. Drones equipped with optical or thermal cameras can accelerate rescue operations by quickly scanning wide areas of the ground, enabling fast localization of missing people. However, the large amount of acquired data requires a significant effort from the rescuers to manually inspect each image, searching for any trace of human presence. Computer Vision tasks, combined with Machine Learning techniques, can help rescuers by automatically detecting traces of human presence from the captured image, limiting the effort required for manual screening. This thesis explores Machine Learning and Deep Learning techniques to identify people from thermal images captured during Search and Rescue missions. The analyzed models approach the problem as an anomaly detection task where mostly background samples are available for model training. These approaches aim to identify anomalies, i.e. those images or areas different from the learned background features. Detailed analyses prove that Deep Learning architectures outperform classical Machine Learning techniques obtaining promising results on the available data set consisting of simulated missions in forest scenarios. The most promising Deep Learning method can obtain over 92.6% of F1-Score on the anomaly class and generate accurate anomaly heatmaps. The developed system can support rescuers during operations by highlighting the potential locations of missing people on thermal images captured during a Search and Rescue mission.

Keywords: anomaly detection, drones, search and rescue, aerial imaging, thermal imaging, computer vision, deep learning

Sommario

Il crescente numero di persone che svolgono attività in montagna ha portato ad un aumento degli incidenti che spesso si verificano in luoghi inaccessibili. Quando una persona ferita, malata o dispersa richiede l'intervento di squadre di soccorso specializzate, l'efficienza delle operazioni è fondamentale. Le missioni di ricerca e salvataggio sono operazioni critiche in termini di tempo che spesso si svolgono in ambienti complessi e difficili. I droni equipaggiati con telecamere ottiche o termiche possono accelerare le operazioni di soccorso scansionando rapidamente ampie aree del terreno, consentendo di localizzare persone disperse velocemente. Tuttavia, la grande quantità di dati acquisiti richiede un notevole sforzo da parte dei soccorritori per ispezionare manualmente ogni immagine ed individuare qualsiasi traccia di presenza umana. I metodi di visione computerizzata, combinati con tecniche di apprendimento automatico, possono aiutare i soccorritori rilevando automaticamente tracce di presenza umana sulle immagini catturate, limitando lo sforzo necessario per la scansione manuale. Questa tesi esplora le tecniche di apprendimento automatico e apprendimento profondo per identificare persone dalle immagini termiche catturate durante le missioni di ricerca e salvataggio. I modelli analizzati approcciano il problema utilizzando tecniche di rilevamento di anomalie in cui immagini di sfondo sono principalmente disponibili per l'addestramento dei modelli. Questi approcci mirano a identificare le anomalie, cioè quelle immagini o aree che differiscono dalle caratteristiche di sfondo apprese. Analisi dettagliate hanno dimostrato che le architetture di apprendimento profondo superano le classiche tecniche di apprendimento automatico, ottenendo risultati promettenti sulle immagini utilizzate per la valutazione dei modelli che rappresentano simulazioni di missioni in scenari forestali. Il metodo di apprendimento profondo più promettente è in grado di ottenere oltre il 92.6% di F1-Score sulla classe anomala ed è in grado di generare accurate mappe di calore per evidenziare le anomalie identificate. Il sistema sviluppato può supportare i soccorritori durante le operazioni di soccorso, evidenziando le potenziali posizioni delle persone disperse sulle immagini termiche catturate durante una missione di ricerca e salvataggio.

Parole chiave: rilevamento anomalie, droni, ricerca e salvataggio, immagini aeree, immagini termiche, visione computerizzata, apprendimento profondo

Contents

1 | Introduction

Search and Rescue (SAR) missions are time-critical operations [2] that can be very challenging and complex. These operations must be carried out as quickly as possible because time is an extremely important factor when searching for an injured person. As time passes, a missing person's survival probability decreases and the search area grows exponentially because a wider territory must be scanned to localize the lost person [71]. Moreover, the rescue team may be exposed to the same adverse conditions that compromised the victim such as rain, snow, or high winds that can pose high risks to rescuers.

In 2022, the Italian National Alpine and Speleological Rescue Corps (*CNSAS, Corpo Nazionale Soccorso Alpino e Speleologico*) reported 10,367 rescue missions for a total of 10,125 people rescued on the Italian territory [28]. From 2018, the annual rescue activity reports have registered a constant growth in the number of rescue operations which has increased by 6% in the last 5 years [27, 28]. The rising number of people who get involved in accidents while carrying out activities in the mountains, such as hiking, mountain biking, and skiing, requires the intervention of specialized rescue teams. According to the 2022 report published by CNSAS [28], the primary cause of accidents is slipping (45.9%), followed by inexperience in completing the performed activity (26.3%), illness (13.7%) and finally, adverse weather conditions.

Most of the rescue operations, specifically 80% of the missions performed by CNSAS in 2022 [28], take place in rough terrains and inaccessible areas such as mountains or forests. The organization of missions in these challenging areas is a complex task that requires scanning vast terrains, especially when the location of a missing person is imprecise or unknown.

Advances in Unmanned Aerial Vehicles (UAVs) [107] have enabled flying drones that can provide critical support to Search and Rescue operations and nowadays have become a standard in all rescue services globally [13, 49, 141]. In contrast to helicopters, drones are more flexible and cheaper in acquisition, maintenance, and operation.

Drones have the potential to shorten search times by quickly scanning a wider area, accelerating the localization of the victims [66]. UAVs equipped with optical or thermal cameras can survey the area of an accident and collect a massive amount of images to identify the location of a missing person [101]. However, the large quantity of captured images must be manually scanned to identify objects of interest, such as people, clothing, or technical equipment. When rescuers are under pressure and strained after hours of work, they can commit some errors that may potentially lead to missing some victims. Indeed, maintaining long-term concentration and attention, even for trained people, is challenging when searching for traces of people from a huge amount of images [126].

Computer Vision tasks, combined with Machine Learning techniques, can speed up this process by filtering frames and highlighting signals of human presence, reducing the burden of manually screening the entire captured video sequence [116]. Moreover, using an automated person detection system, only the areas with a higher likelihood of human presence are selected for manual inspection. Consequently, the human error rate can be limited because the time and energy required by rescuers to analyze images are significantly reduced.

The identification of people can be performed by leveraging supervised Computer Vision techniques such as image classification [41] or object detection [126, 129] that require an extensive annotated set of images to specify the location of a person. However, in SAR operations, images are recorded from a bird's eye view and this type of viewpoint is not contained in large available data sets [123] that are typically used to train detection models [126]. This implies that data collected for the development of detection models must be manually annotated. The huge amount of data required for training a supervised deep model should be composed of a balanced distribution of classes of objects that the model should be able to detect. This implies that in the use case of SAR missions, many examples of people lying on the ground must be collected to properly train a detection method. Unfortunately, most of the images captured during a typical SAR scenario depict background scenes and only a limited number of people images are available. Moreover, collecting a comprehensive set of images representative of the class of visible people is a complex task. Indeed, the visual appearance of a person viewed from a drone is highly variable due to the combination of articulated poses of an injured person, degree of tree occlusions, and drone altitude that generate a virtually infinite number of possibilities. Consequently, training a model to recognize every possible human appearance is an extremely complex task. Researchers tried to approach the localization of people from aerial thermal images using an object detection model, obtaining poor results and proving the difficulties of this use case [129].

In this thesis, people detection from drone imagery in SAR missions is approached as an anomaly detection task. This implies that the implemented models are trained to learn a comprehensive representation of background images and detect any anomaly that does not belong to the learned distribution. Consequently, when a lost person, a backpack, or any other human trace is captured from a drone image, the model should signal the presence of an entity that is not recognized as background. The works [7, 54, 116] exploit anomaly detection concepts to identify people in a SAR scenario from color images. However, rescuing lost or injured people often involves searching densely forested terrain, where sunlight is blocked by the trees and other vegetation that strongly occlude the ground. In these scenarios, thermal sensors are used to visualize the temperature difference between human bodies and the surrounding environment. For this reason, this thesis focuses on applying anomaly detection techniques on grayscale thermal images since they may be more relevant in wider mission scenarios, especially during night operations.

In the literature, few data sets are dedicated to the training of people detection models in SAR scenarios. The presence of occlusions, typical in wilderness Search and Rescue missions, is not recreated in popular aerial images data sets [87, 162] which therefore cannot be used for training models for this use case. The *Data: Search and Rescue with Airborne Optical Sectioning* [130] data set is composed of thermal and RGB images captured over forest scenarios associated with the annotations of people locations. This data set has been originally created to train an object detection model using the images derived from the combination of many frames captured during a flight. The images, which have different viewpoints of the ground, are integrated using the AOS (Airborne Optical Sectioning) algorithm proposed in the original work [129]. In this thesis, the single thermal frames are exploited for the people identification task without applying the post-processing combination. From the thermal images, a set of smaller tiles are extracted and each generated tile is then classified as background or anomaly based on whether it contains a person or not. After an extensive survey of the anomaly detection techniques from image data presented in the literature, several Machine Learning and Deep Learning models have been selected and implemented. All the fine-tuned models have been evaluated on the generated data set composed of tiles, obtaining 92.6% F1-score using the most promising model.

The objective of this thesis is not to try to replace the intervention of the rescuers but, on the contrary, to assist the responders during search missions. The developed system is designed to collaborate with the rescue teams by filtering most of the background images to: 1) reduce the burden of manually screening the entire captured videos and 2) highlight areas of the images with higher chances of containing an injured or lost person. However, it is the responsibility of the rescuers to analyze the results generated by the system and decide if a ground team operation should be commanded to verify if a person is actually present in the location indicated by the system.

The contributions of this thesis can be summarized as follows:

- The motivations supporting the choice of approaching the problem as an anomaly detection task are discussed, highlighting the challenges of the use case and the limits of other works proposed in the literature.
- A comprehensive review of the works proposed in the literature for the Search and Rescue scenario is presented.
- An extensive review of the state-of-the-art techniques for anomaly detection in images is conducted.
- A selection of the most appropriate methods for people identification in UAV imagery during a SAR mission is conducted.
- A data set for training and evaluating the implemented models is generated from the thermal images taken from a publicly available data set [130].
- Several Machine Learning and Deep Learning anomaly detection models are implemented and fine-tuned.
- A comparison of the results obtained by the selected models is provided, along with a detailed analysis of the strengths and weaknesses of each method.
- Anomaly heatmaps are generated on some example images to showcase the outputs generated by the proposed models during a realistic Search and Rescue mission.

The rest of the thesis is organized as follows:

- Chapter 2 introduces the concepts of anomaly detection and provides a detailed literature review of the proposed techniques for identifying anomalies in image data. Then, publicly available data sets and methods for Search and Rescue missions are presented and discussed.
- Chapter 3 presents the available data set, the preprocessing steps performed to obtain the training and evaluation sets, and the implemented methods with their corresponding hyperparameters.
- Chapter 4 describes the evaluation procedure adopted to identify the best configuration of parameters for each method and presents a quantitative and qualitative analysis of the results.

- Chapter 5 summarizes the work and discusses possible improvements and future directions.

2 | Related Work

In this chapter, an overview of the literature about anomaly detection and applications to assist Search and Rescue missions is presented. Section 2.1 introduces the main characteristics and challenges of anomaly detection with a focus on the Computer Vision and Artificial Intelligence methods available for detecting anomalies in visual data. Section 2.2 provides an overview of related works for supporting Search and Rescue missions and discusses the publicly available data sets that can be used for training models to perform anomaly detection on drone imagery.

2.1. Anomaly Detection

Anomaly detection is the set of techniques used to identify patterns in data that deviate from expected behavior [24]. Anomalies can provide critical information in a wide variety of application domains such as detecting credit card fraud [96, 115, 119], intrusion detection for cybersecurity [15, 25, 60], highlighting of malignant tumors in medical X-Ray images [3, 38], monitoring of machine failures [65, 140] or defect detection in industrial images [120, 149].

Some key assumptions used by anomaly detection techniques to discriminate between normal and anomaly samples are that normal instances are generated from a known distribution and can be clustered together in some feature space. Conversely, anomalies are rare, with a low likelihood of occurrence, and are substantially different from normal data.

An anomaly is defined as a pattern that does not conform to expected normal behavior. Therefore, a common approach is to model the representation of the normal data, i.e. a region in a feature space containing all the normal samples, and any observation that does not belong to the normal region is considered an anomaly. The main challenge of this approach is the definition of a normal region because often the boundary between normal and anomalous samples is nebulous due to noisy data that cannot be accurately classified as normal or anomaly. The definition of *anomaly* is relative to each application domain and can strongly depend on the use case. For example, a small variation of a monitored value in the medical domain might be an anomaly, while deviations in the stock market domain are the normality. Due to these challenges, anomaly detection problems cannot be solved with a general technique. Therefore, anomaly detection models should be flexible enough to leverage the specific characteristics of each application domain to learn a strong representation of *normality* [24].

An example scenario that motivates the essential difference of anomaly detection tasks with respect to traditional classification tasks is the use case of industrial machine monitoring where a system should detect any abnormal behavior which may indicate a possible failure. Measurements of normal operation are easy to obtain but samples of faults are difficult to collect, therefore only a few anomaly instances are available in the training data. In this case, it is not viable to wait until faults occur because they could damage the machines and put operators at risk. Moreover, due to the intrinsic nature of anomalies, it could be theoretically impossible to generate every possible anomalous instance because the underlying behavior is often dynamic in nature, thus new types of outliers for which there is no labeled training data available might arise [24]. For this reason, it is essential to develop a model capable of identifying any outlier only by leveraging the information collected during normal operation [68].

Based on the extent of labels availability, detection techniques can be classified into three categories: (1) *Fully Supervised Anomaly Detection*, (2) *Semi-Supervised Anomaly Detection*, (3) *Unsupervised Anomaly Detection.*

Fully Supervised Anomaly Detection

Models trained in a fully supervised manner assume that a comprehensive set of both normal and anomaly instances is available in the training set. Any new unseen data point is compared against the model to determine which class it belongs to. However, the scarcity of abnormal samples and their high intra-class variability are two major challenges that prevent obtaining an accurate representation of anomalous behavior.

Semi-Supervised Anomaly Detection

Techniques that approach the problem in a semi-supervised manner assume that the training data has labeled instances only for the normal class. This approach is more widely applicable than supervised techniques due to the high availability of normal data and the scarcity of anomaly instances. However, this class of models must be robust to a possible small fraction of outliers that could be wrongly introduced in the training set which are assumed to be normal.

In some cases, semi-supervised data sets may also include some samples of labeled anomaly data that could be used to accurately define the boundary that encloses the normal instances [122].

Unsupervised Anomaly Detection

In an unsupervised learning setting no labeled data is available, therefore the outliers are detected based on the intrinsic properties of the data instances. The training set is assumed to be composed only of normal samples, however, some anomaly samples could be wrongly introduced in the data set. For this reason, models must be robust to the presence of a small subset of outliers. This category of models assumes that in the test set, normal instances are far more frequent than anomalies.

Image Anomaly Detection

Anomaly detection on image data is an active field of research and the high level of interest in this subject can be demonstrated by the large number of models presented in the literature [37, 63, 100, 144, 155, 156]. The problem of detecting outliers from visual data has been approached in many different ways by computing statistical image characteristics or exploiting advanced Machine Learning techniques. In recent years, the evolution of complex Deep Learning models enabled more accurate detection and localization even of small irregularities.

When the target of anomaly detection is the image data, it is referred to as *visual anomaly detection* or *image anomaly detection*.

Applications of visual anomaly detection are widespread in various sectors such as monitoring of road traffic [128], detection of irregular tissues in medical images [3] and surveillance tasks [51]. Another important field of application for anomaly detection techniques is the industrial inspection of manufacturing lines [149]. In this scenario, automatic systems assure product quality by continuously monitoring production processes and signaling any detected defect.

Thanks to the advancements in UAVs technology [117, 137], the use of drones is rapidly growing across a wide range of applications such as traffic monitoring [67], post-disaster investigations [13, 41, 70], agricultural operations [33, 69], and crowd surveillance [102, 145]. Nowadays, techniques of anomaly detection from UAV imagery can be applied to a wide selection of practical use cases. These approaches often have applications in the surveillance of critical infrastructures [17], crowd monitoring scenarios [6], or abnormal events identification [26]. These systems constantly monitor a predefined area and signal any unexpected behavior detected in the scene to the security personnel who could intervene to mitigate possibly dangerous situations.

Anomaly detection on images is implemented with a different approach with respect to other types of data, such as time series or text. In fact, the high dimensionality of image data and the locality property of pixels information requires extracting some useful features that could be later processed by classical anomaly detection methods. Thus, an anomaly detection pipeline can be logically subdivided into two steps: the **feature extraction** part that generates a compact representation of the input image in the form of a *feature vector* and the **anomaly detection** step that identifies abnormalities from the extracted information. The anomaly detection techniques proposed in the literature can be broadly categorized into three classes [156]: (1) *Density Estimation*, (2) *One-Class Classification*, (3) *Image Reconstruction*.

Density Estimation

Density Estimation methods estimate the probability distribution of normal images and identify whether a newly observed sample is anomalous by computing the distance from the normal distribution. The intrinsic assumption is that the features extracted from anomaly images do not belong to the representation derived from normal images. The main limitation of this approach is the need of having a large number of training samples for estimating a reasonable probability density. Moreover, when dealing with images, this problem becomes more challenging due to the high dimensionality of visual data.

One-Class Classification

One-Class Classification models aim to build classification models when the anomalous class is either absent, poorly sampled, or not well defined [68]. This situation imposes training models exploiting only one class, by attempting to define a decision boundary in the feature space to enclose normal images. These methods require a reduced number of training samples because they do not need to estimate a probability distribution of the normal data. However, they still suffer from the problems of scalability and high dimensionality when dealing with feature vectors extracted from images that may have a large number of components.

Image Reconstruction

Image Reconstruction techniques map the image to a low-dimensional vector representation and then try to reconstruct the original image by finding the inverse mapping from the latent representation to the image space. The concept behind this approach is the compression of redundant information in normal data. Since the models are trained only with normal samples, the learned compact representation should not be able to properly model anomalous features. As a consequence, anomalous areas of the image should not be accurately reconstructed and thus the reconstruction error should be larger than in normal images. Based on the value of the reconstruction error, which is an indication of the difference between the original image and the reconstructed one, these models are able to detect anomalous images by thresholding the resulting scores.

Initial anomaly detection methods were based on modeling the background of an image and detecting anomalous pixels by computing statistical properties. Later, thanks to the advances in Artificial Intelligence, more advanced and complex approaches based on Machine Learning and Deep Learning techniques were proposed. These models build compact representations of normal images and identify the abnormal samples as outliers of the learned distribution. The approaches of image anomaly detection models can be categorized in: 1) **Spectral Techniques** that are based on the probability distribution of pixel intensities, 2) **Machine Learning Models** that use variants of traditional Machine Learning methods to learn a boundary only from normal samples [68] and 3) **Deep Learning Models** that learn a complex representation of the normality [100].

2.1.1. Spectral Techniques

One of the most simple yet popular anomaly detectors is the Reed-Xiaoli (RX) algorithm [118]. This method models the background of an image as a multidimensional Gaussian distribution of pixel intensities and computes for each pixel an *anomaly score* that is proportional to the distance from the background distribution. Assuming that anomalies and background pixels have different intensity distributions, abnormal pixels have a higher distance from the normal distribution. Therefore, anomalous pixels can be detected by thresholding the anomaly scores. The original variant of the RX detector (Global RX) assumes the image background to be homogeneous, but this hypothesis may not always be adequate. Therefore many variants of the base algorithm such as Local RX and Kernel RX [81] were developed to address this problem. The baseline version (Global RX) computes the background color distribution over the entire image, thus the computation is influenced by the presence of anomalous areas. Local RX, differently from Global RX, computes the local background distribution of a smaller area by using a sliding window that is moved over the image. This window contains another inner window which is used to exclude the possibly anomalous pixels at the center from influencing the computed background model. Therefore, the background distribution in an area of the image is estimated only by considering the pixels between the outer and the inner window. Kernel RX [80] is another variant that extends the original Global RX by introducing a non-linear kernel function that maps the original image space into a high-dimensional feature space. This approach enables the estimation of a complex non-linear distribution that better models the background pixels improving the detection of outliers. The RX detector and its variants are often used to identify anomalies in hyperspectral or multispectral images with a high number of channels. In these images, each pixel is characterized by a large amount of information that is used to build a representative high-dimensional distribution of the background where anomalous pixels can be easily identified as outliers of the background model. Some applications on RGB images were presented in the literature [7, 54, 116, 136], despite color images being composed only of three channels which provide less information. However, color anomaly detection techniques were successfully implemented to identify pixels with a significant spectral difference from the neighbors distribution [7]. The background representation computed by spectral models is very simple since it is based only on the intensity value of each individual pixel without considering the relations with the neighboring pixels. For this reason, this technique applied on grayscale images is not effective because a single channel does not provide enough information to correctly discriminate an anomalous pixel from the approximate distribution of pixel intensities.

2.1.2. Machine Learning Techniques

In a conventional multi-class classification problem, the learning of the optimal decision function is supported by the availability of samples from all the classes involved. However, anomaly detection tasks are characterized by the scarce availability of anomalous data, therefore only normal instances can be exploited to define the classification margin. As a consequence, the boundary should be defined to enclose the majority of the training data but should be tightened around the normal samples to minimize the likelihood of including anomaly samples.

Machine Learning methods are implemented in a wide range of application scenarios and many models have been introduced in recent years [161]. However, many algorithms are designed to be trained with a supervised data set that is representative of all the classes. In an anomaly detection setting, since abnormal samples are rare, there is no supervision information. Moreover, anomaly samples have high variability in shape, color and size that would prevent supervised learning techniques to capture salient features of the anomalous patterns [156]. The choice of the Machine Learning model strongly depends on the specific application and characteristics of the data, therefore there is no one-size-fits-all solution. The literature proposed many hybrid models that combine multiple Machine Learning methods to address specific challenges of anomaly detection such as the scarcity of abnormal samples [103].

Features Extractors

Extracting relevant features to represent the intrinsic content of images is a challenging problem in Computer Vision [58]. In pattern recognition and image processing, the feature extraction step is a special form of dimensionality reduction [74] that builds a compact representation in the form of a *feature vector*. Accurately selecting the appropriate feature extraction method is essential because it has a relevant impact on the performance of the Computer Vision task [74]. In the case of anomaly detection tasks, a good feature extraction technique should be able to extract similar feature vectors for patterns within the normal data but it should effectively discriminate them from the representations of anomaly samples.

Color is one of the most important features of images [112] and a number of color spaces have been studied in the literature, such as RGB, LUV, and HSV [54]. In the literature, many important color features have been proposed, including color histogram [62], color moments (CM) [39], color coherence vector (CCV) [110], and color correlogram [57]. Texture is a very useful characterization for a wide range of images and is a key element of human visual perception for recognition and interpretation [58, 112]. Usually color is a pixel property while textures are associated with a group of pixels. Texture analysis is used in a very broad range of applications such as the biomedical field, industrial automation, document analysis, remote sensing, and face recognition. A large number of texture feature extraction methods are proposed in the literature [44, 58] but selecting the correct method for each particular application is not an easy choice because there are many factors that must be taken into account. For instance, approaches that are based on the structure may be appropriate only for regular texture. Moreover, if the application requires invariance to some image transformation (e.g. rotation, translation, scaling), the set of possible techniques is reduced.

Textural feature extraction methods can be broadly classified in: 1) *statistical* methods, 2) *structural* methods, 3) *transform-based* methods, 4) *model-based* methods, and 5) *graph-based* methods [58].

Statistical methods compute statistics-based metrics on the spatial distribution of gray-level values in the neighborhood of a pixel. GLCM considers the relationship between two neighbor pixels and counts the occurrences of gray-level combinations in a predefined set of directions and distances. The information on texture captured by the GLCM is used to compute a set of statistical metrics such as the Haralick features [52]. A similar approach is the GLRLM [42] method which evaluates sets of consecutive pixels with the same gray-level value in a given direction. For each direction, a run-length histogram is computed and then a set of features that characterize the image texture can be derived. Basic Image Features (BIF) [30] is another statistical method that extracts features by applying simple filters on the image and then represents texture as histograms over a visual vocabulary of features constructed by clustering. Local Energy Pattern [160] is another statistical histogram-based representation. Local feature vectors describe local-oriented energies that are then used to build the frequency histogram of the global texture representation. Local Binary Pattern (LBP) [104] focuses on the patterns of intensity variations within a subregion of interest by combining the analysis of local structures with the analysis of occurrences. The main drawback of the original variant is the sensitivity to rotations that have been solved in some evolutions such as Dominant Rotated LBP (DRLBP) [99] that normalizes each local descriptor with respect to a reference in the local neighborhood. Histogram of Oriented Gradients [31], mainly used for object recognition, is based on the idea that object shapes can be characterized by distributions of intensity gradients and builds a local descriptor that is a histogram of the occurrences of gradient orientations. Similarly, Histogram of Gradient Magnitudes (HGM) [138] computes a histogram of the local gradient magnitudes but can obtain rotation invariance by ignoring the gradient directions. Another statistical approach is Deterministic Walk (DW) [10] which uses a "tourist" to explore the texture as a walk following a deterministic rule. From the study of the trajectories distribution, a histogram-based signature is computed to discriminate the image texture.

Structural approaches decompose the image into primitives and use their spatial arrangements to characterize textures. An example of this technique is Scale-Invariant Feature Transform (SIFT) [94] which builds a descriptor of gradient location and orientation of interesting points that are detected with the Difference of Gaussians computed from the original image. SIFT descriptors are often used as local texture descriptors for vocabulary-learned methods [43].

Transform methods represent an image in a space that better describes the characteristics of a texture. In the Fourier transform-based approaches [95], an image is decomposed into its frequency components and represented as a weighted combination of vertical and horizontal sinusoids of various frequencies. The features are robust to rotation and translation but do not accurately describe local variations of textures. Gabor Decomposition [97] filters an image with a bank of Gabor filters at different scales and orientations allowing the computation of frequency and orientation information at multiple resolutions. This method is robust against illumination changes or noise. Additionally, some evolutions are also invariant to rotations. Wavelet Transform [86] performs the texture analysis in both the spatial and frequency domains by approximating the image texture using transformations of a basis function called the mother wavelet. The derived wavelet coefficients are used as texture representation. This method is not robust to changes in texture direction but some evolutions have rotation-invariant property [61]. LETRIST [143] extracts features that are robust to rotation, illumination, and scale. The image is convolved with directional Gaussian derivative filters at multiple scales. On the extremum responses of the computed image derivatives, a set of linear and non-linear transformations are applied to derive a set of features. The derived discriminant features are combined in a histogram that is used as the image descriptor.

Model-based methods represent texture using mathematical models. The Complex Network [23] method represents the pixels as nodes of a graph and then maps similarities between pixels as links between the network nodes. The weights of the links are computed from the gray-level values. This approach presents a higher success rate of classification when compared to traditional texture analysis methods but it is sensitive to noise. Wold Decomposition [89] models texture by decomposing it into three components that measure periodicity, randomness, and direction of the texture. The estimated parameters are used as the texture descriptor.

Graph-based approaches represent the image pixels as a graph of points on which texture features are extracted. Local Graph Structures [1] method computes features in a local neighborhood of the computed graph by measuring the distribution of local micropatterns. Graph of Tourist Walk [11] generates graphs from the trajectories produced by the tourist walks of the Deterministic Walk method. The texture descriptor is built from the statistical position and dispersion calculated from the graphs. This method is robust in the recognition of microtextures.

In Table 2.1, some of the most commonly used textural feature extractors used in Computer Vision tasks along with their invariances are summarized.

Features Classifiers

The main challenges of using Machine Learning algorithms for anomaly detection tasks are the need for large amounts of labeled data and the risk of overfitting the prevalent class of normal samples. Some of the traditional Machine Learning models have been adapted to learn from normal data in an unsupervised or semi-supervised setting that better fits the requirements of anomaly detection tasks [103]. Machine Learning classifiers for anomaly detection tasks can be categorized in: 1) *One-Class Classification* methods and 2) *Density Estimation* methods.

One-Class Classification techniques define a boundary to enclose areas in the fea-

Name	Year	Type	Invariances
Grey Level Co-occurrence Matrix (GLCM)/Haralick [52]	1973	statistical	rotation
Grey Level Run-Length Matrix (GLRLM) [42]	1975	statistical	rotation
Local Binary Pattern (LBP) [104]	1996	statistical	illumination
Gabor Decomposition [97]	1996	transform	noise, rotation, illumination
Wold Decomposition [89]	1996	model	rotation, scaling
Scale-Invariant Feature Transform (SIFT) [94]	2004	structural	rotation, scale, affine transformations
Histogram of Oriented Gradients (HOG) [31]	2005	statistical	geometric, photometric transformations
Complex Network [23]	2008	model	rotation
Wavelet Transform [86]	2010	transform	scale, illumination
Basic Image Features (BIF) [30]	2010	statistical	rotation
Deterministic Walk (DW) [10]	2010	statistical	scale
Graph of Tourist Walk [11]	2011	graph	rotation, noise
Local Graph Structures (LGS) [1]	2011	graph	illumination, scale, shift
Fourier Transform [95]	2013	transform	rotations, translation, noise
Local Energy Pattern (LEP) [160]	2013	statistical	rotation, brightness, contrast
Dominant Rotated LBP (DRLBP) [99]	2015	statistical	illumination, rotation
Histogram of Gradient Magnitudes (HGM) [138]	2015	statistical	rotation
LETRIST [143]	2018	transform	rotation, noise, illumination, scale

Table 2.1: List of most commonly used textural feature extractors used in Computer Vision tasks along with their invariances.

ture space that contain the normality class and classify as an anomaly any point that is mapped outside the learned boundary. One-Class SVM (OC-SVM) [134] learns a hyperplane in the feature space that separates the normal class from the origin point which is selected to represent anomaly samples. This model is very sensitive to outliers in the training set and cannot perform well with high-dimensional features. The evolution of this method is Support Vector Data Description (SVDD)[150] which adopts a hypersphere to enclose normal data in the feature space while excluding anomalies. SVDD has the main advantage of being able to leverage the knowledge of a few abnormality examples available during training to improve the definition of the separating boundary. Isolation

Forest [90] builds a set of binary trees during training that tries to separate each sample from the rest of the normal data in fewer possible steps. Points that can be easily isolated are considered anomalies because they are distant from the distribution of normal data. In a similar approach, k-Nearest Neighbors [29] can be used to identify as an anomaly any observation isolated from the normal data. An instance is considered isolated if the nearest neighbor has a distance above a predefined threshold.

Density Estimation methods approximate the distribution of normal class data and detect anomalies as samples that do not belong to the normal distribution. Kernel Density Estimation [109] leverages the estimate of the probability density function of normal data points. Normal instances have high local densities, therefore data points that are located in sparse regions with low local densities are detected as anomalies. Similarly, Local Outlier Factor [18] compares the estimated local density of a given instance with the local densities of its k nearest neighbors. A point that has a significantly lower density with respect to the mean density of the neighbors is classified as an anomaly, indicating that it is isolated from other samples.

In Table 2.2, some of the most popular Machine Learning classifiers for anomaly detection tasks are summarized.

Name	**Year**	**Type**
Kernel Density Estimation (KDE) [109]	1962	density estimation
K Nearest Neighbors (KNN) [29]	1967	one-class classification
Local Outlier Factor (LOF) [18]	2000	density estimation
One-Class SVM (OC-SVM) [134]	2001	one-class classification
Support Vector Data Description (SVDD) [150]	2004	one-class classification
Isolation Forest (IFOR) [90]	2008	one-class classification

Table 2.2: List of most popular Machine Learning classifiers for anomaly detection tasks.

2.1.3. Deep Learning Techniques

The development of Deep Learning [84] and the increasing interest in these powerful techniques pushed the researchers to adopt deep convolutional models in many anomaly detection scenarios [20]. Deep convolutional networks are implemented in a wide range of applications such as image classification, object detection, face recognition, and image segmentation. The capability of extracting compact representations from high-dimensional data, such as images or videos, is the key aspect that allows deep models to outperform traditional techniques in challenging tasks such as visual anomaly detection [100]. Deep Learning techniques provide more effective anomaly detection compared to traditional methods because they perform end-to-end optimization of the entire detection pipeline. This enables the fine-tuning of the learned representation specifically to tailor the anomaly detection task [108].

Some application fields in which Deep Learning techniques are commonly used are medical anomaly detection [38] and anomaly localization in industrial images [149] where models have proved to be very reliable and robust to realistic situations.

Deep Anomaly Detection Models

Deep anomaly detection techniques can be categorized in three conceptual paradigms [108] that are: (1) *Deep Learning for Feature Extraction*, (2) *Learning Feature Representations of Normality*, (3) *End-to-End Anomaly Score Learning*.

Deep Learning for Feature Extraction

These methods aim at leveraging Deep Learning methods to extract compact representations from high-dimensional data that are exploited to perform anomaly detection tasks. With this approach, the feature extraction and anomaly scoring steps are fully disjointed and independent from each other, thus the Deep Learning methods are purely employed as dimensionality reduction. The main advantage of using Deep Learning models instead of classical dimensionality reduction methods, such as Principal Component Analysis (PCA) [19, 133], is the better capability in extracting semantic-rich features. In fact, these models are able to extract more powerful representations with respect to computing statistical properties of the image [14]. On the opposite side, the fully disjointed nature of feature extraction and abnormality detection steps leads to suboptimal anomaly scores because the extracted features are not optimized for the detection task. Therefore, these techniques may not preserve relevant information to discriminate normal data from anomaly instances.

Learning Feature Representations of Normality

These techniques learn the representations of data instances by optimizing a generic objective function that is not primarily designed for anomaly detection. However, the learned representation can still empower the anomaly detection task because the models are forced to capture the most relevant normal data regularities. These methods learn a representation of normal data and then any divergence of a new observation indicates the occurrence of an anomaly. The methods that exploit this approach are data reconstruction, generative modeling, and One-Class Classification.

Autoencoders (AE) [12] aim to learn a low-dimensional feature representation space from which the data instances can be accurately reconstructed. In anomaly detection tasks, these networks are used for learning a compact representation of normal data that minimizes the reconstruction error. To learn such representation, the model has to retain only the most relevant information of the normal instances. As a result, the anomaly instances should be poorly reconstructed because they are characterized by features different from the normal data. Therefore, the reconstruction error can be directly used as an anomaly score. However, this approach is not optimized for detecting irregularities because the reconstruction objective function is designed for dimensionality reduction rather than anomaly detection. Some implementations that leverage an Autoencoder for the classification of anomalous images are DCAE [98] and RCAE [21].

Generative Adversarial Networks (GANs) [48] is another popular technique for visual anomaly detection. This network is composed of a generator network that outputs synthetic samples similar to the instances in the training set, and a discriminator network that has the objective of identifying generated samples from the real data. These models aim at learning a latent feature space that precisely captures the normality underlying the normal data. The basic assumption is that the generator network should learn to accurately generate normal data instances but poorly perform at recreating anomalies samples. Some implementations of this technique are AnoGAN [132], EBGAN [159], GANomaly [4] and OCGAN [111]. The main advantage of GANs is that they demonstrated superior capability in generating realistic instances with respect to Autoencoders. The downside is that these networks are more difficult to train. Furthermore, similarly to reconstruction methods, the anomaly detection performance is suboptimal because the objective function is optimized for data synthesis rather than anomaly detection.

One-Class Classification methods, such as OC-CNN [106] and OC-NN [22], learn a description of the normal data in the training set that is used to detect whether new instances conform to the learned distribution. Deep SVDD [121] learns a boundary in the latent representation space that encloses the majority of the normal training samples. During inference, the observations mapped outside the boundary are classified as anomalies. In this case, the advantage of deep models is the capability of learning an optimized mapping that separates normal and anomaly samples in the feature space. Deep SAD [122] extends this model by leveraging a small number of anomaly data during training to better refine the boundary that encloses the normal data by excluding the available anomaly instances. This model has been later extended with a Fully Convolutional (FC) architecture that, using a similar loss, exploits segmentation techniques to accurately localize anomalous areas [92]. Other models, trained in a one-class learning setting, are capable of accurately localizing abnormal areas of the image by using semantic segmentation techniques. PaDiM [32] is an implementation that extracts features with a Convolutional

Neural Network (CNN) from patches of the input image and builds an anomaly map to localize defects. Other more advanced models, such as CFlow [50] or FastFlow [157], use a normalizing flows architecture to accurately localize areas of the images that contain anomalies. Finally, methods such as MemAE [46], TrustMAE [148], and CAVGA [154] exploit attention-based memory modules to focus the decoder network on specific regions of the image to better localize anomalous areas.

End-to-end Anomaly Score Learning

In this approach, the anomaly scoring is not dependent on existing anomaly measures (e.g. reconstruction error) but a neural network directly learns the anomaly score in an end-to-end fashion. Different from simply learning the representation of normality, these methods simultaneously learn the feature representations and anomaly scores. Thanks to the unified objective, the training is greatly optimized because the implemented loss functions directly aim at learning the anomaly score in an end-to-end manner. End-to-end One-Class Classification models aim at optimizing a discriminative model to separate normal instances from adversarially generated pseudo anomalies. The one-class model is built on the discriminator network of a GAN model that is directly used to classify anomalies. An implementation of an adversarially learned One-Class Classification network is ALOCC [124] that optimizes two networks through the GANs approach with one network trained as the one-class model, and the other trained to enhance the learning of normal instances by generating distorted outliers. The advantage of this method is that anomaly classification models are adversarially optimized in an end-to-end fashion. However, the instability of GANs training may generate instances with different quality, obtaining unstable anomaly classification performance.

In Table 2.3, some of the most common Deep Learning anomaly detection models are presented.

Transfer Learning

Training a Convolutional Neural Network from scratch is a challenging task that requires a large volume of data for the network to learn useful data representations. However, the lack of available anomalous instances in anomaly detection scenarios makes the problem more difficult. For this reason, Transfer Learning techniques [163] can overcome this issue by improving the learning of a *target task* by exploiting the knowledge gained while approaching a different *source task*. With transfer learning, the knowledge learned to solve the *source task* is used as a baseline for the training of the model that tackles the *target task*. In this case, the training of the *target task* aims at improving the accuracy by fine-tuning the weights of the pre-trained network, which are already able to extract useful information. Therefore, a smaller data set is required for the fine-tuning of the

Name	Year	Architecture	Task
DCAE [98]	2011	autoencoder	classification
RCAE [21]	2017	autoencoder	classification
AnoGAN [132]	2017	GAN	segmentation
DeepSVDD [121]	2018	CNN	classification
GANomaly [4]	2018	GAN	classification
EBGAN [159]	2018	GAN	classification
ALOCC [124]	2018	GAN	classification
OC-CNN [106]	2019	CNN	classification
OC-NN [22]	2019	CNN	classification
OCGAN [111]	2019	GAN	classification
MemAE [46]	2019	attention based	segmentation
DeepSAD [122]	2020	CNN	classification
FCDD [92]	2020	fully convolutional	segmentation
PaDiM [32]	2021	CNN	segmentation
CFlow [50]	2021	normalizing flows	segmentation
TrustMAE [148]	2021	attention based	segmentation
CAVGA [154]	2021	attention based	segmentation
FastFlow [157]	2021	normalizing flows	segmentation

Table 2.3: List of most common Deep Learning models used for anomaly detection tasks.

parameters because the initialized weights should converge faster to the optimal solution.

The first step of transfer learning is the selection of a proper pre-trained model architecture that can obtain good results on the *target task*. Then, the parameters of the target network are initialized with the values of the source model and, before starting the training process, the weights of the first layers are frozen. The layer freezing is performed to prevent the gradient descent optimization from updating the low-level features learned by the original network. This prevents losing useful information that would negate the advantages gained with the transfer learning technique. Only the top layers, which represent high-level features, are updated since they model task-specific data representations. Therefore, it is a good practice to fine-tune only the features learned by the top layers to optimize the results of the *target task*. On the other hand, lower-level features learned in the first layers of the network are useful for a much wider variety of tasks since they extract basic image features that are used to build higher-level abstract representations. These initial network layers contain the useful knowledge gained on the *source task* that can be transferred to the *target task*. Usually, the most common and well-known CNN architectures used for transfer learning are DenseNet [56], AlexNet [73], Inception [147], VGG [142] and ResNet [53]. These models were originally trained to perform image classification on the ImageNet [123] data set that is composed of more than 14 million images of objects classified in

20,000 classes. These models, despite obtaining very high performance meaning that they were able to learn discriminating features, are trained only on natural images (e.g. cars, airplanes, dogs). Therefore, when these pre-trained networks are used as a source model for transfer learning, they inevitably have learned features that are characterized by an intrinsic bias [63] towards these natural images. Therefore, when shifting the domain problem, such as in an anomaly detection task on a different category of data (e.g. medical images, industrial defect detection), the encoders of these networks may not be capable of extracting relevant features that are useful to solve the *target task*.

Self-Supervised Learning

Self-Supervised Learning (SSL) techniques gained popularity in recent years [5] since they allow to pre-train a deep model without the need for any type of supervision information. The task used for pre-training is called *pretext task* while the task that leverages the extracted knowledge and fine-tunes the weights to tackle the real problem is called *target task*.

The unlabeled data in the training set are exploited to solve the *pretext task* and the learned data representation is transferred to the model that will solve the *target task*. Self-Supervised Learning approaches are similar to Transfer Learning techniques since they both transfer knowledge by initializing the network that solves the *target task* with the weights of the model trained to solve the *source task*. The main difference between the two approaches is that with Transfer Learning the source model is trained on a large labeled data set, whereas Self-Supervised Learning leverages an unlabeled data set to learn data representations. In fact, the motivation to apply Transfer Learning is to avoid training from scratch a model that is impracticable when the available labeled data is limited. Conversely with Self-Supervised Learning, by training a model with unlabeled samples which are similar to the data used during the *target task* training, the *pretext task* is capable of extracting features that are useful also to solve the *target task*.

Self-Supervised Learning methods can be broadly categorized as: 1) techniques that learn to reconstruct an input image and 2) methods that solve a supervised classification task by exploiting automatically generated pseudo-labels. Generative approaches use Autoencoders based on an encoder-decoder architecture [12] to reconstruct a given input image. The encoder generates a compact representation of the input in the latent space while the decoder aims at reconstructing the original input from the latent representation. The encoder is forced to learn characterizing features of the input data to construct a relevant representation that is leveraged by the decoder to accurately reconstruct the input. The features learned by the encoder, which are representative of the input data, can be used as a baseline for transferring knowledge to the *target task* [125]. Other approaches are based on a Generative Adversarial Network (GAN) architecture [48] that is composed of a generator network that produces realistic images starting from input noise and a discriminator that identifies generated images from real samples. To produce realistic images, the generator is forced to learn an accurate feature representation of the input noise. However, the network should learn the representation of the input data, not the distribution of random noise. Consequently, Bidirectional GAN (BiGAN) networks [36] introduce an encoder to map an input image to the feature representation (noise space) allowing the network to invert the generation process and obtain a feature representation of the input samples [91]. A possible *pretext task* for generative approaches consists in removing a region of the input image and training a network to generate the missing pixels [158]. To be effective in the reconstruction, the encoder needs to understand the image content to allow the model to generate a reasonable prediction of the missing region.

Predictive approaches are based on automatically generated pseudo-label that are derived from known transformations applied to the input image. The idea is that a model should learn a useful representation of the data while learning to predict which transformation has been applied to the input. In order to correctly predict the generated pseudo-label, the network has to understand the context of the image. Therefore it is able to extract relevant features that can be used by the *target task* network. An example of a *pretext task* is to rotate the input image and train a model to predict the orientation applied to the input [45]. To correctly predict the applied rotation, the model has to extract discriminating features that characterize the semantic content in the image.

The main challenge of Self-Supervised Learning is that *pretext tasks* must be carefully selected and the domain of the downstream task must be taken into consideration to learn useful information. For example in the case of aerial imagery, the rotation is not a representative feature of the samples because a drone could capture images from any orientation. Therefore, a pre-train task that randomly rotates an image and tries to predict the transformation might not be suitable for this domain because it does not allow the network to learn useful features.

Self-Supervised Learning for Anomaly Detection

Anomaly detection models in an unsupervised setting need to familiarize themselves by exploiting only the available normal data to distinguish anomaly samples in the learned features space. Self-Supervised Learning approaches allow a model to learn both low-level features (e.g. color, texture) and high-level features (e.g. shape, position, direction) of the normal samples, thus improving the sensitivity to anomalies. After training, a model should distinguish anomaly instances that do not have the characteristics that have been learned only from normal samples.

In recent years, many anomaly detection models have been presented to improve performance by leveraging Self-Supervised Learning techniques [55, 135]. Due to the unsupervised setting of anomaly detection tasks, some *pretext tasks* specifically designed for these scenarios have been introduced. RotateNet [45] trains a model to predict the geometric transformations such as rotations, translations, and horizontal flipping that are applied to the input image [45]. Another method randomly removes partial regions of an image and learns to reconstruct the original sample by performing in-paintings [158]. By learning to reconstruct only normal samples, the network implicitly learns the distribution of normal data. Therefore, it should be unable to accurately reconstruct anomalous regions that would be identified due to a higher reconstruction error. Another *pretext task* specifically designed for increasing the performance of local defect detection models is CutPaste [85] which simulates an anomaly sample by clipping a patch of a normal image and pasting it back at a random location. With this approach based only on normal data, the network is capable of detecting unknown anomalous patterns of an image without the need for anomaly samples.

2.2. UAV Search and Rescue

The goal of a Search and Rescue mission is to search for injured or lost people in the shortest amount of time when their locations are imprecise or unknown. The adoption of drones helps in quickly scanning wide areas of the ground without requiring any expensive machines such as helicopters or airplanes. The large number of images captured during a flight must be accurately analyzed by rescuers to identify traces of people or technical equipment. However, when rescuers are under pressure after hours of work, they might miss some potential victims. The combination of powerful deep models that can run on constrained devices and the improvements in UAV technology that enabled longer flights opened the possibility of applying Deep Learning techniques to aerial imagery for a wide variety of remote sensing scenarios [105]. Especially in the use case of Search and Rescue missions, many dedicated Deep Learning models have been developed to automate the detection of potential targets [13, 126, 129]. In this section, a review of the publicly available data sets for anomaly detection tasks from drone imagery is presented. Then, data sets related to Search and Rescue operations are discussed and finally a brief overview of the related work for detecting people from drone imagery is introduced.

2.2.1. Data Sets

The quality of the available data set used for training the model is of the utmost importance for the development of an accurate anomaly detection method. Many works evaluate the performance of novel models on data sets derived from popular classification data sets such as MNIST [82] or CIFAR [72] where one class acts as normal data and a subset of the remaining classes simulates the anomaly instances [4, 21, 98, 121, 122]. However, this approach does not accurately simulate the characteristics of anomalies in a real scenario. Therefore, results obtained during the model evaluation phase may not reflect the performance in a real-world application. The unavailability of a standard data set for assessment is a relevant limitation when developing novel anomaly detection models because available data sets are very far from realistic situations [100]. Thus, to comprehensively validate a novel anomaly detection method, having access to a more realistic and representative data set is crucial.

In recent years, the evolutions of UAVs enabled the application of drones in a wide range of complex scenarios [137] where the bird-eye view perspective can be exploited to better localize the objects of interest in each application. Consequently, it is necessary to improve the semantic understanding of visual data collected from aerial perspectives that are not represented in other popular visual data sets. Since anomaly detection from drone imagery has gained interest and proved to have potential in many application scenarios, some dedicated data sets specifically oriented to support these use cases have been published in the literature. Mini-Drone Video Dataset (MDVD) [16] is used for surveillance tasks and is composed of 38 videos of normal situations (e.g. parked vehicles or people walking), and suspicious behaviors (e.g. fighting scenes or people stealing cars). For the development of anomaly detection or change detection methods, the UAV Mosaicking and Change Detection Dataset (UMCD) [9] provides low-altitude drone videos acquired at a distance from the ground between 6 and 15 meters. This data set contains 50 aerial videos collected in different environments such as urban, dirt, and countryside at different times of the day. The main limitation of these standardized visual anomaly detection data sets is that they usually differ from reality since they are captured under controlled conditions [100]. For example, in UCSD Ped1 and Ped2 data sets [87] that depict video surveillance scenarios, the considered anomalies are very simple (e.g. golf cart traveling on the sidewalk) and the real-world anomalous events are not properly reflected. Consequently, the performance obtained during operation in a real application could be worse than the results assessed during model validation. Despite these data sets being composed of images with varying characteristics specific to each use case, they all depict scenes in an urban environment where anomalies are constituted by anomalous behaviors of people (e.g. crowd monitoring) or objects placed in unusual areas of the image (e.g. change detection, aerial surveillance). In the use case Search and Rescue scenario in a mountain environment, the forest foliage often occludes the ground and as a consequence, human bodies could be partially or totally hidden by the trees inducing the targets to have highly variable shapes. As a consequence, the common anomaly detection data sets may not be useful due to the different nature of the images.

A possible data set for training models to detect people in a Search and Rescue mission is the SARD Database [127]. This data set depicts casualties captured from drone images in Search and Rescue scenes where actors simulate exhausted and injured persons. The 1,981 images contain 6,532 annotated people that were captured during the daylight from different camera angles and altitudes (between 5 and 50 meters). However, this data set has some aspects that are strongly different from the typical frames captured in forest scenarios because the images mainly depict more generic terrains (e.g. roads, high and low grass, and rocks) while only a limited subset of images captures forest environments.

The only available data set captured over forest scenarios is *Data: Search and Rescue with Airborne Optical Sectioning* [130] that, in the original work, is used to train an object detection network for identifying lost people during a Search and Rescue mission in forest scenes. The data set is composed of 12 flights captured over forests of different types (broadleaf, conifer, mixed) and 6 flights in open field performed at an altitude of 30-35m. For each flight, both thermal and RGB frames are provided, aligned to cover the same view of the ground. The ground truth is composed of 9,684 annotated bounding boxes that enclose the entire body of all the people placed on the ground. This data set was later expanded with a similar data set *Data: Autonomous Drones for Search and Rescue in Forests [78]* that is composed of 17 shorter flights that captured similar images over other forests with visual appearances similar to samples of the original data set.

Challenging Aspects

The application scenario is very complex because the anomaly detection models should be able to identify most of the possible targets while keeping the rate of false alarms at a minimum. However, some challenging characteristics strictly related to the use case of detecting people in a dense forest must be taken into consideration. Two of the main challenges which are intrinsic to any wilderness Search and Rescue data set are: (1) *Trees Occlusions*, (2) *High Variability Of Pixel Intensities.*

Trees Occlusions

Rescue missions in forest scenarios are characterized by the presence of tree occlusions that induce a disturbance on the ground visibility which cannot be neglected [76]. Depending on the forest type, which determines the average structure of trees and thus the visibility of the ground, 20% to 50% of targets can be totally hidden when viewed from an individual viewpoint (single thermal image). However, this does not imply that up to half of the targets can be missed during a rescue mission, because a person which is completely hidden in a frame can be visible from a consecutive frame. The reason is that the visibility of a person lying on the ground can significantly change when the point of view slightly changes because the randomness of the tree irregularities allows the targets to be partially visible. Since most of the people's heat traces are partially occluded by trees, the appearances of lost people from the aerial images are highly variable. Therefore, the captured targets can have very different shapes that are influenced by the posture of the person lying on the ground and the structure of the branches that disrupt the ground visibility. Figure 2.1 shows some examples of targets that have very different visual appearances when viewed from a drone thermal camera. Due to the high variability between samples, training an object detection model to localize shapes of human bodies from raw thermal images has been demonstrated to obtain very weak performance [129].

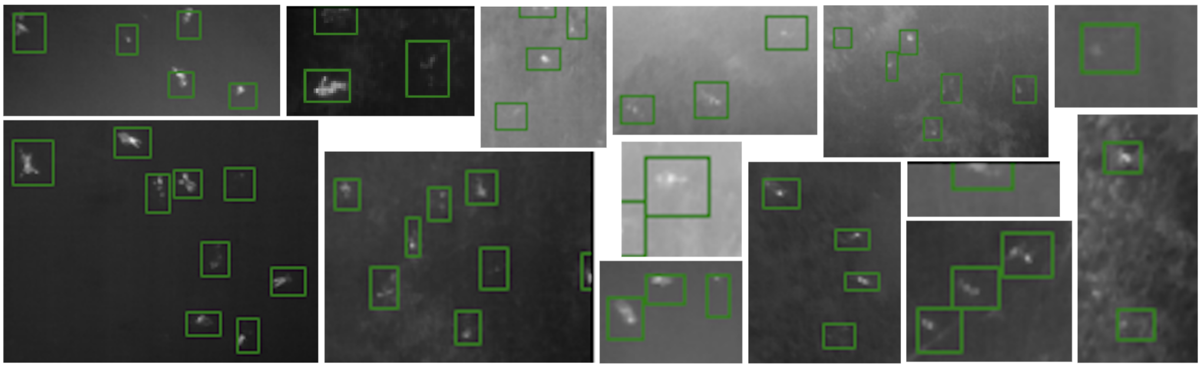
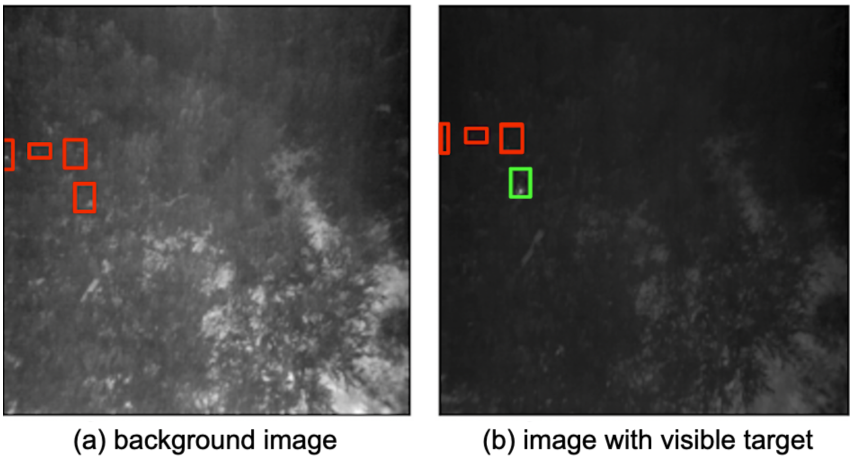


Figure 2.1: Examples of targets captured from aerial images that are characterized by very different visual appearances caused by tree occlusions.

High Variability Of Pixel Intensities Most of the commercially available thermal cameras are equipped with a *non-radiometric* sensor which measures only the relative differences between temperatures in a scene, without providing any information about the absolute values. Then, all the measures are normalized between the maximum and minimum temperature to obtain an intensity value in the range [0,255] for each pixel of the resulting image. The consequence of the normalization is that the same absolute temperature may be associated with different pixel intensities in various images. This is caused by the fact that the normalization is dependent on the context of each captured scene. For this reason, consecutive frames in a flight may have very different intensity distributions because the temperature range is influenced by the presence or absence of a visible target. For example, if no human heat trace is captured by the thermal camera (Figure 2.2.a), the background scene has a smaller range of temperatures and, after the normalization, all the image pixels take values in the entire intensity range [0,255]. Ground areas that are slightly warmer than the rest of the background are associated with brighter pixel intensities. On the other hand, if a person with a heat trace significantly warmer than the background is visible in the scene (Figure 2.2.b), the temperature range is wider. Thus, most of the background areas that have colder temperatures than the human body are associated with darker pixel intensities. As a consequence, the frames in the data set have a highly variable distribution of pixel intensities, implying that even images of the same flight can be visually very different, as shown in Figure 2.2.


(a) background image

(b) image with visible target

Figure 2.2: Examples of two consecutive frames captured during a flight. In frame (a) no target is visible and the background temperatures are represented by the full range of pixel intensities. In frame (b) the visible person has warmer temperatures associated with brighter intensities whereas the colder background areas are represented by darker pixels. Red boxes are empty annotations in the ground truth while green boxes indicate the presence of a visible person.

2.2.2. Artificial Intelligence-based Applications

In the literature, many applications for detecting people in Search and Rescue missions have been proposed that exploit RGB cameras [126] or thermal sensors [129] mounted on a drone. In particular scenarios, such as in maritime rescue operations, multi-spectral cameras are adopted [41]. The most popular Computer Vision tasks implemented to localize people are *object detection* [41, 126, 129] and *anomaly detection* [7, 54, 116]. The object detection approach learns human body features and exploits that knowledge to recognize similar characteristics in new images. Anomaly detection techniques, as already discussed in Section 2.1, model the distribution of background samples and identify people as the outliers of the learned background representation.

UAVs have been used for many years in catastrophic scenarios such as earthquakes and tsunamis [93] and, at the same time, detection techniques have improved to deal with complex situations such as the presence of occlusions or articulated poses of injured people. Before the adoption of powerful neural network models, the effort was focused on implementing traditional Computer Vision techniques to identify people from drone imagery. For instance, in [35] people detection in urban locations is performed by identifying candidate person silhouettes on thermal images characterized by temperatures in the range of human body heat. The extracted regions are then projected on the color images where an object detector uses Haar-like features to classify potential targets. This method was developed to operate in post-disaster scenarios with good visibility of the ground, thus it is not robust to occlusions that could partially hide bodies. The work presented in [8] trains an ensemble of models capable of detecting the full-body or upper-body of a person to increase robustness to partial occlusions. To improve the detection capability, the images are augmented with information on the height and pitch angle of the drone to better estimate the scale that a person should have in the frame. However, this model is trained to localize people only in an indoor scenario under controlled light conditions. Therefore, performance could be worse in different situations, such as outdoor Search and Rescue missions.

Other techniques rely on the spectral information of the image pixels to identify any color anomaly which, in a rescue scenario, may indicate the presence of an injured person (e.g. clothes with different colors from the background). Spectral anomaly detection methods from drone images have been successfully implemented in works such as [54, 116] where Reed-Xiaoli algorithm [118] models the background color distribution and identifies any unusually-colored object in the image by generating an anomaly map that is shown to responders to indicate potential locations of injured people. Other methods based on colors exploit custom algorithms to identify pixels with a higher contrast with respect to the background [146] and filter the identified regions based on a minimum area to improve robustness to image noise. These methods are well-suited for wilderness Search and Rescue imagery because objects of interest (e.g. clothes, backpacks, technical equipment) do not have a specific pattern but are often characterized by colors significantly different from the background. Moreover, since only the color spectrum is analyzed, these methods are less influenced by occlusions or variations in lighting conditions. On the other side, the main downside of these approaches is that they detect any possible object not belonging to the background (e.g. cars, bicycles, roofs) that could generate a high number of False Positives.

More recently, Convolutional Neural Network models have been trained to detect people in Search and Rescue missions from color images captured in daylight conditions. For instance, [126] trained an object detection model to identify injured people in non-urban areas (e.g. roads, high and low grass, rocks) and introduced a methodology to simulate different weather conditions to increase the robustness of the method in a wider range of scenarios. Another object detection network presented in [13] aims at detecting people involved in incidents caused by avalanches in a winter environment. In the case of wilderness Search and Rescue scenarios, where vegetation strongly occludes the ground, reducing the visibility of people, a promising technique has been presented in [75, 77, 79, 129] that introduces a preprocessing step named *Airborne Optical Sectioning* (AOS). AOS combines thermal images captured from different perspectives aiming at reducing the influence of occlusions and emphasizing the targets placed on the ground. This technique starts by aligning images captured from different poses, then averages the pixels of different images that refer to the same point of the ground. Since each image has a different perspective of the occlusions, their influence on the ground visibility changes. Therefore, the combination of many frames significantly reduces the impact of the occluding vegetation, improving the visibility of the ground. However, to align the images, the accurate pose estimation of the thermal frames is based on a Computer Vision technique that computes the relative pose of RGB images captured by a dedicated camera installed in parallel to the thermal camera. The main limitation of this pose estimation technique is the computationally intensive task to align the RGB frames. Moreover, this technique cannot be applied in low-light conditions preventing the application of this detection model in night operations. Further research presented in a later work demonstrated that similar performance can be obtained by aligning the images using the imprecise GPS poses measured with an onboard sensor mounted on the UAV [131]. The people identification task is performed with an object detection network trained on the thermal frames captured during a flight combined with the AOS preprocessing. An alternative implementation exploits the AOS preprocessing to combine the RGB images captured during a flight and identifies injured people as anomalies in the pixel spectral information using the RX algorithm [7].

Use-Case Limits

Despite the active research for assisting responders during rescue operations, there are still many limits in the presented work that could hinder the performance that can be obtained when applying the proposed techniques to realistic Search and Rescue scenarios.

Specifically, three main problems can be identified which are not completely addressed by methods available in the literature: (1) *Sensor Type*, (2) *Detection Method*, (3) *Data Set Peculiarities*.

Sensor Type

Most of the applications detect people on images captured with RGB cameras and address the problem only during good daylight conditions. Methods proposed in these works do not provide any support for night missions where RGB images contain little or no information and thus cannot be used for detecting people. However, night operations are not rare and therefore a detection system capable of supporting rescuers in a wide range of conditions would be more useful.

Detection Method

Many works train object detection models to specifically identify features associated with people samples. Therefore, other signals of human presence (e.g. backpacks or clothes) are not taken into consideration by these systems even if they could provide relevant information to indicate the presence of an injured or lost person. Moreover, a person partially occluded by trees, or with an articulated and unnatural pose, could potentially be missed because these models could not recognize a human body if it has shapes that are significantly different from the examples seen during training.

Data Set Peculiarities

Data sets used for training often do not reflect the intrinsic characteristics of a real rescue mission scenario. Some works exploit a set of images characterized by the absence of occlusions where people are completely visible from the drone camera [146]. In other data sets, there is a relatively high number of people placed near each other (because a large number of samples is required to train supervised models such as image classifiers, and object detectors) and, in these cases, the samples used for training can be very different from the images captured during a real mission [131]. In both cases, a model that obtains excellent results during the performance assessment phase could significantly underperform during a more complex realistic scenario. This would make the implemented system unusable for a search operation where time is critical and any malfunction of the detection model can negatively impact the mission effectiveness.

3 | Data Set and Methods

In this chapter, the publicly available data set used for evaluating the anomaly detection techniques is presented along with the preprocessing steps applied to clean the ground truth and generate a proper training/validation/test split of images. Then, a brief overview of the data set main issues is discussed. Finally, the anomaly detection methods and techniques studied in this thesis are introduced.

3.1. Data Set

UAVs are widely used in many application domains and much effort is dedicated to improving Computer Vision tasks from drone imagery. Despite the growing interest in the use of drones, few works in the literature are focused on rescue missions. Consequently, there is a lack of available data sets oriented at Search and Rescue missions in forest scenarios, a task that presents particular issues to be addressed (Section 2.2.1). For example, data sets such as those available for crowd surveillance [87] cannot be applied because they contain people from different perspectives, e.g., standing or walking in urban environments. Other data sets for generic SAR missions [126] mostly contain totally visible people without any occlusion thus representing simpler scenarios in comparison to the task of finding lost people in dense forests.

The only available data set dedicated to rescue missions in forest scenarios is *Data: Search and Rescue with Airborne Optical Sectioning* [130]. As shown in Table 3.1, the data set comprises 12 flights performed at an altitude of 30-35m over different types of forest (broadleaf, conifer, mixed) and 6 flights in open field. Figure 3.1 shows some images of the various forest types.

Two pieces of information are provided for each flight: thermal and RGB frames, aligned so that they cover the same ground view, and the associated drone pose at the capturing instant. Among all flights, there are 6,095 RGB images with a size of 6,000x4,000 px and a Ground Sample Distance (GSD) of 0.23-0.27 cm/px. Each RGB image is coupled with a smaller thermal image with a size of 640x512 px and a GSD of 5.6-6.6 cm/px. In

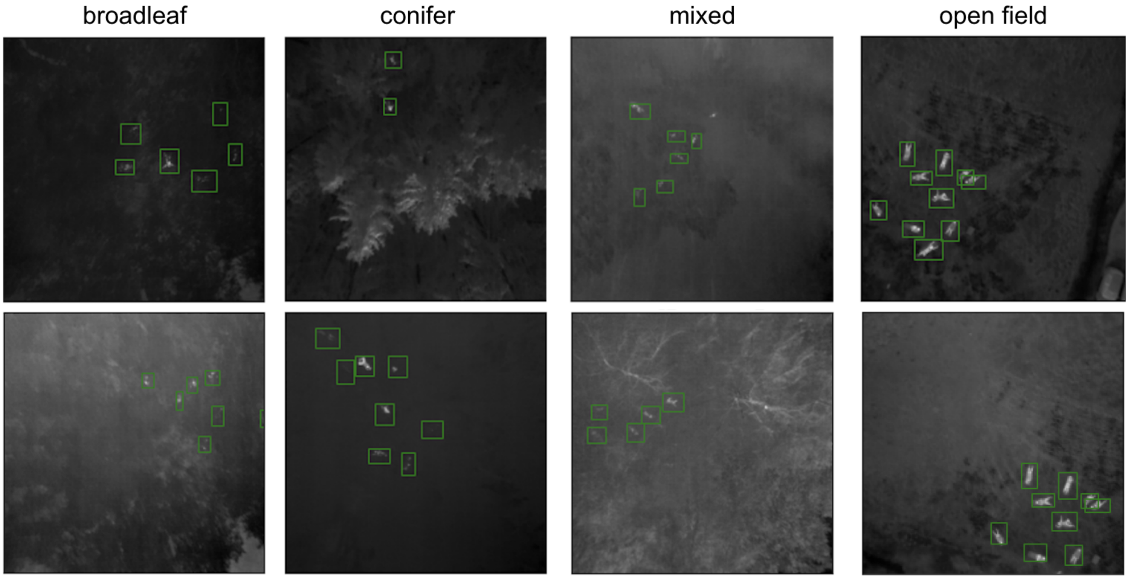


Figure 3.1: Frames from flights over different forest types (broadleaf, conifer, mixed) and in open field. Images taken from the available data set [130]. Green boxes enclose the people annotated in the ground truth.

the original work, high-resolution RGB images are used for precise pose estimation, while thermal images, cropped to 512x512 px, are used for the people detection task. Each flight covers an area of the ground of 30x30 meters and simulates a mission scenario where a variable number of subjects (between 2 and 10) are placed on the ground with articulated poses to imitate ill or injured people. Each person is annotated in the ground truth with a bounding box that encloses the entire body shape. In the original work [129] the full set of thermal images captured during a flight are combined with the *Airborne Optical Sectioning (AOS)* algorithm to highlight the view of the ground by reducing the effect of the occlusions. This technique consists of averaging the pixel intensities of many frames that have different perspectives of the same point on the ground to make the targets more visible in the resulting image. Examples of application of this algorithm are shown in Figure 3.2.

The *AOS-integrated* frames were used by the authors to manually annotate the ground truth bounding boxes of the targets placed on the ground. Since the transformation from each thermal image pose and the resulting *AOS-integrated* image is known, the inverse transformation was used to automatically annotate all the thermal images starting from the manually annotated ground truth.

The original data set has been designed to perform fully supervised object detection but, in this thesis, it will be exploited for the anomaly detection task which requires less

Flight	Forest	People	Images	Annotations	Date
F0	Conifer	3	402	1,040	4 Oct 19
F1	Broadleaf	10	153	1,107	24 Oct 19
F2	Broadleaf	10	260	2,203	24 Oct 19
F3	Mixed	6	380	1,682	25 Oct 19
F4	Mixed	6	366	960	25 Oct 19
F5	Conifer	10	327	1,895	8 Nov 19
F6	Conifer	10	321	1,816	8 Nov 19
F7	Broadleaf	2	31	62	20 Nov 19
F8	Broadleaf	0	358	0	17 Jan 20
F9	Broadleaf	0	358	0	17 Jan 20
F10	Mixed	0	399	0	10 Apr 20
F11	Conifer	0	418	0	10 Apr 20
O1	Open field	10	370	2,880	8 Jan 20
O2	Open field	10	364	2,610	8 Jan 20
O3	Open field	6	414	1,465	22 Jan 20
O4	Open field	6	382	1,599	22 Jan 20
O5	Open field	5	406	1,119	7 Feb 20
O6	Open field	5	386	1,043	7 Feb 20

Table 3.1: Flights of the original data set [130]. For each flight, information about the forest type, people lying on the ground, number of images, and total number of annotations are reported. Many annotations may refer to the same person because they can be seen in multiple frames.

supervision. Firstly, RGB images are discarded because this work focuses only on the people detection problem from thermal images that are less influenced by the presence of tree occlusions. Moreover, some images are not relevant to this work (e.g. open field), thus only a subset of flights is kept for training anomaly detection models. Finally, the automatic annotation of thermal images, starting from the *AOS-integrated* frames, introduced noise in the ground truth because the annotated people may be completely hidden in some thermal images. The empty annotations should be removed from the ground truth. Finally, the remaining images are split into training/validation/test sets to define a proper data set for training and evaluation of the analyzed anomaly detection models.

3.1.1. Data Set Preprocessing

A series of steps have been applied to the original data set to: 1) remove unused images, 2) clean the noise in the annotations and 3) organize the various flights to train and validate the implemented anomaly detection models.

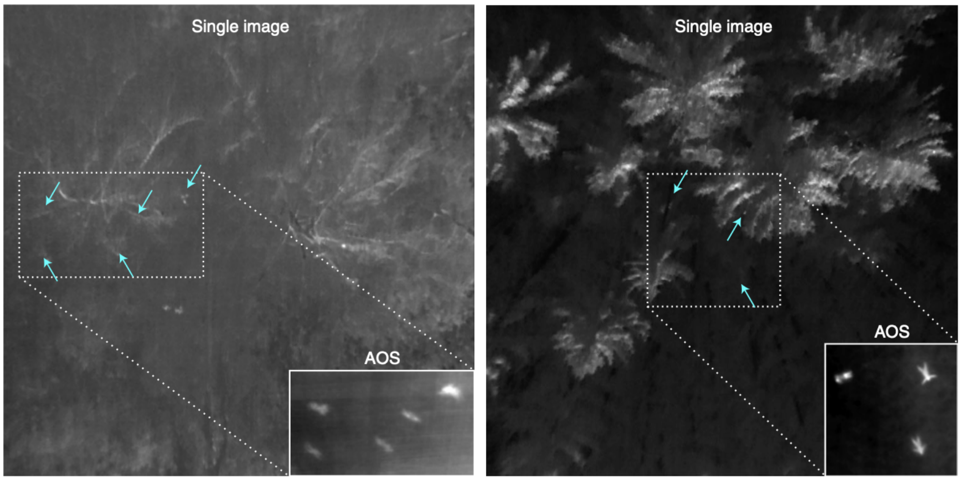


Figure 3.2: Examples of thermal frames captured over mixed and conifer forests. The arrows indicate partial heat signals of occluded people. The insets show AOS results obtained by the combination of multiple thermal frames. Images taken from [129].

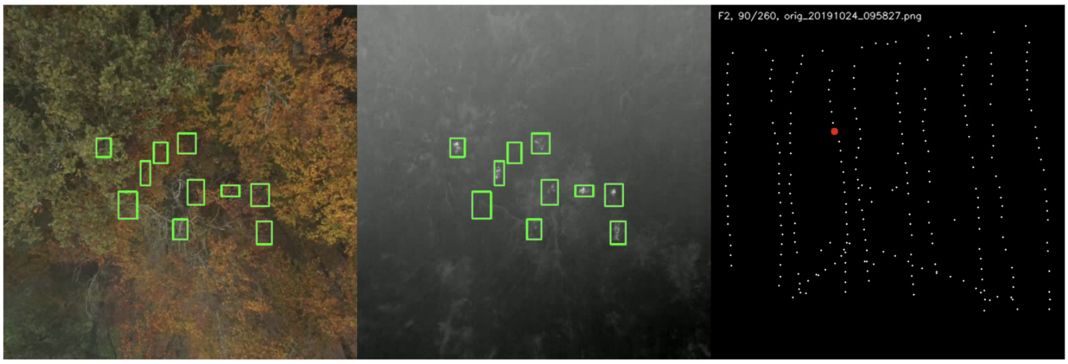
Sensor selection

Forest occlusions have a more substantial impact on the RGB images, causing the majority of the targets to be entirely hidden. Instead, in thermal frames, targets are usually at least partially visible because some heat traces are detected through the foliage. For this reason, RGB frames were discarded and the presented work is oriented to applying anomaly detection models only on thermal frames which are less influenced by the occlusions. Moreover, infrared images can be helpful also during night missions when RGB frames would provide no relevant information to the rescuers because they would be totally dark. Figure 3.3 shows a pair of RGB and thermal images captured over a broadleaf forest. From the RGB image (Figure 3.3.a) no target is visible, whereas on the thermal frame (Figure 3.3.b) many people are partially visible.

Ground truth cleaning

Due to the automatic annotation process of thermal images, there is no certainty that the targets are effectively visible. The main issue of this approach is that in many raw thermal frames, some people are completely occluded by the trees and as a consequence, some bounding boxes do not contain any visible target. This can lead to apparently worse results due to an increase in False Negatives since the models cannot detect some hidden targets that are still annotated in the ground truth.

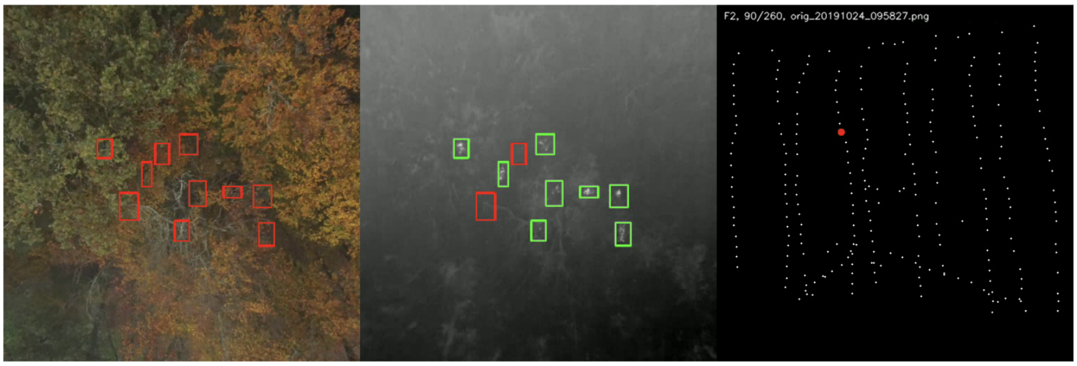
To remove such noise from the ground truth, every bounding box has been manually annotated to determine if it effectively contains at least a partially visible target or if the trees completely occlude the ground. This cleaning step was performed using the



(a) RGB frame (b) Thermal frame (c) Drone pose

Figure 3.3: Pair of RGB frame (a) and thermal frame (b) extracted from a flight over broadleaf forest. Green boxes enclose the people annotated in the ground truth. No target is visible from the RGB image whereas, with the thermal image, most of the people can be detected. The red point on the right (c) indicates the current pose of the drone in the flight trajectory.

AnnotatorLocalization tool provided by the *ODIN* framework [152]. This framework allows defining ad-hoc meta-annotations for the ground truth data. Each bounding box has been annotated with a *visible* property that indicates whether the contained target is at least partially visible or not. Figure 3.4 shows an example of bounding boxes containing visible and occluded targets. After the manual annotation, all the empty bounding boxes (target completely hidden) were removed from the ground truth and only the visible targets were used for the evaluation of the tested models.



(a) RGB frame (b) Thermal frame (c) Drone pose

Figure 3.4: Pair of RGB frame (a) and thermal frame (b) extracted from a flight over broadleaf forest. Green boxes indicate visible targets while red boxes are empty annotations that have been removed from the ground truth. The red point on the right (c) indicates the current pose of the drone in the flight trajectory.

Flights selection

The original data set contains various flight scenarios: different forests, open-field flights, and flights with no targets. Flights performed in open field (*O1-O6* in Table 3.1) are not representative of a typical Search and Rescue scenario because the targets are always visible due to the absence of occlusions. For this reason, these flights were discarded and not used for training and evaluation of the detection models. Moreover, in the original data set, there are four flights (*F8, F9, F10, F11* in Table 3.1) that do not contain any targets. These flights were also discarded to avoid having too many empty frames in the data set used for the experiments.

3.1.2. Anomaly Detection Data Set

After the flights selection step, the original 6,095 thermal images were reduced to 2,240 frames and, with the manual cleaning of non-visible targets, more than 30% of ground truth annotations of the selected flights have been removed. The remaining images captured over the various forest types are visually different and have distinctive characteristics such as the shape of the trees and the degree of occlusion of the ground.

Data collection flights cover a small area of the ground. Therefore, all the frames were captured very close to each other resulting in many similar images due to the large overlap between consecutive frames. For this reason, all the frames from a flight were assigned to the same set to avoid having similar images in different sets and prevent data leakage. Since an anomaly detection model should work on most Search and Rescue scenarios, the training data must comprise frames from all the available forest types to avoid overfitting a specific scenario. Defining a balanced separation of flights to build a standard data set with train and test splits is not straightforward because the distribution of images and targets by forest type is not balanced, as shown in Figure 3.5. Conifer samples are prevalent both in image samples and in the number of targets because two of the three flights on conifer forests have many images. On the other hand, flights on mixed forests are shorter with fewer frames and have a small number of targets because the visibility through this type of forest is limited. Moreover, despite broadleaf forest flights having fewer frames, they have a higher number of visible targets because the trees are less dense and induce a lower degree of ground occlusion.

Thermal frames of the selected flights have been grouped in three subsets to obtain the *training*, *validation*, and *test* sets. The main objectives of the split were to obtain a balanced proportion of bounding boxes (visible targets) in each set and to distribute forest types on different sets with the constraint of having at least one flight for each

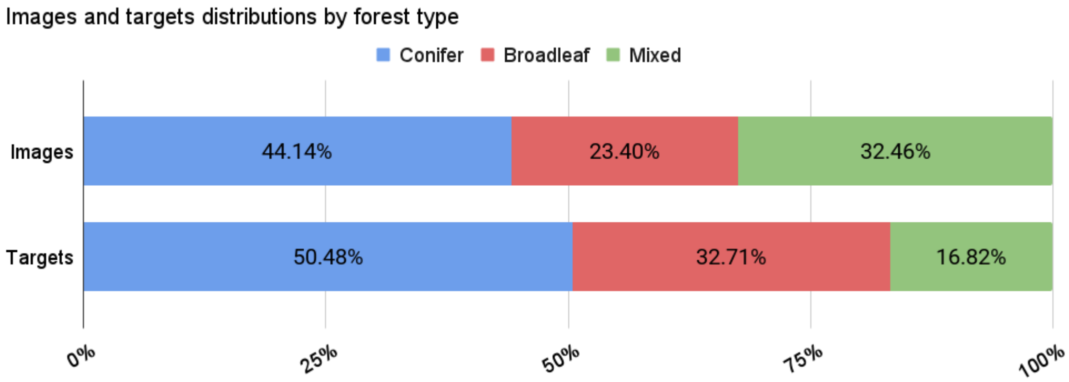


Figure 3.5: Images and targets distributions by forest type in the original data set [130].

forest type on the training set. At the same time also the number of images in each set should be balanced to avoid having many annotations on a reduced number of frames. The split was performed to distribute 70% of the annotations in the *training set* with images from broadleaf/conifer/mixed forests, 10% of bounding boxes from a mixed forest flight in the *validation set*, and 20% of the annotations in the *test set* captured from flights over broadleaf and conifer forests. Table 3.2 presents the final split.

Set	Flights	Forest types	Images	Targets
Training	F2, F4, F5, F6, F7	broadleaf, conifer, mixed	1,305	4,972 *(70%)*
Validation	F3	mixed	380	791 *(11%)*
Test	F0, F1	broadleaf, conifer	555	1,349 *(19%)*

Table 3.2: Data set used for the training and evaluation of anomaly detection techniques.

3.1.3. Data Set Issues

The resulting data set is a fair enough representation of realistic search missions. Still, it presents characteristics that can be very different from a real case scenario and can introduce a strong bias in the learning process, thus preventing the trained models from properly operating during a real mission. It should be noted that the original problem is very complex and difficult to approach due to some intrinsic aspects that cannot be removed, such as the presence of a high degree of occlusions caused by the tree foliage, the high flight altitudes that must be used to avoid any obstacle (e.g. high trees or power lines) that make the targets appear very small in the captured frames, and the need to support nighttime operations.

Before presenting the implemented anomaly detection methods, a list of the most important issues of the data set and the main deviations from a real mission are described.

Flights performed in daytime

All the images selected from the original data set were captured during the daytime between October and November. During this period, solar radiation is still sufficiently intense to heat the ground inducing warmer temperatures on tree crowns and on the exposed branches due to the reflection of sunlight [129]. The effect is that some upper forest foliage and branches exposed to direct sunlight can have heat footprints similar to the heat traces emitted by people. This can induce confusion in the model, which can struggle to discriminate a possible lost person from a warmer branch because the pixel intensities are similar. In some cases, the shape of a partially exposed branch can resemble the outlines of a human body.

Limited number of possible mission scenarios

The available images were captured only in the daytime during a limited period of the year with similar light and weather conditions and over a restricted selection of terrain types. Only three different types of forest (broadleaf, conifer, mixed) are shown in the frames, therefore other typical SAR scenarios such as rocky terrains or open field mountains are not available for testing the anomaly detection models. Moreover, more challenging weather conditions such as fog, rain, or snow are not represented in this data set or any other data set in the literature. Finally, realistic samples of post-disaster scenarios such as heartquakes, avalanches, or landslides are not publicly accessible and thus, models can not be assessed in these critical missions that have strongly different features with respect to the case of searching for an injured person in a forest.

Placement of people on the ground

The data set authors performed drone flights to obtain many targets for training an object detection model. For this reason, many people (up to 10) were placed in a constrained area of the ground to obtain many samples. Although this scenario allows obtaining many diversified examples, it does not represent a realistic scene because usually only a few people can be seen inside an image during a search mission. Therefore the presence of many people introduces a bias in the data set that is a significant deviation from real data that can be observed during a search operation. This is because, with many targets located nearby, the chances of detecting at least one person are much higher and thus the performance of a model might degrade during a real use case. The performance assessment should consider this aspect to try to estimate the behavior of a model on a different set of images. Moreover, only a small percentage of background images do not contain any visible target, conversely to a realistic scenario where the majority of the flight would be over an empty forest and only a limited number of images would contain some traces of human presence.

Flight altitude

The data collection flights were performed 30m to 35m above the ground to keep a safety margin from the maximal tree height. However, this altitude is still relatively low because in some cases it is necessary to fly at a more elevated distance from the ground to avoid other obstacles present in the scene (e,g. electrical power lines). In this case, the targets would appear much smaller and more difficult to detect. The drone altitude may generally change during operation for other reasons, such as corrections due to high winds. This implies that a tolerance on the mean target size should be considered when developing an anomaly detection model.

Frames similarity

Images from a drone flight are captured near to each other and the predefined flight path covers only a small area on the ground (approximately 30x30m). This induces a high percentage of overlap, and similarity, in consecutive frames. The consequence is that there is some redundancy in the frames of the data set that can introduce a bias in the model evaluation step. If a model is confused by a particular shape or texture of an area of the ground, the confusion could be repeated for all the frames that have a similar view of the ground, degrading the performance.

Ground truth noise

The visual appearance of tree occlusions has high intra-class variability and very often background areas can be very similar to a target which may confuse the anomaly detection model. Additionally, some bounding boxes may appear bigger than the body silhouette because some parts of a person could not be visible or the entire body could be hidden by the occlusions.

3.2. Target Detection

Anomaly detection on drone imagery has some challenging aspects which need to be considered when developing a model. First of all, robustness to image rotations is essential because the orientation of a person in the scene is not fixed. Then, a method should have some degree of robustness to scale variations because the flight altitude could vary between different flights or missions (trees may be higher or the presence of obstacles such as power lines may impose a higher flight altitude).

3.2.1. Anomaly Detection Pipeline

This work aims to develop an anomaly detection system that can be deployed to assist rescuers during real missions to quickly locate potential targets in a video stream, reducing the burden of manually analyzing a large amount of captured images. Consequently, accurate localization at pixel-level precision is not essential. A coarse indication to highlight potential areas of the images where a target might be located is sufficient to guide the attention of the responders. For this reason, the anomaly detection problem was approached as a classification task of tiles extracted from the images captured during a mission flight. Each generated tile is classified as *background* or *anomaly* depending on whether or not a target is detected. The predictions of tiles extracted from an image can then be combined to generate an *anomaly heatmap* that allows the localization of targets by highlighting detected anomalies in the original image, helping rescuers focus their attention on areas that have a higher chance of containing an injured person.

To test various combinations of feature extractors and classification models, the pipeline shown in Figure 3.6 has been implemented to generate tiles from the images and extract features that are classified by an anomaly detection model. A final evaluation step assesses the model performance.

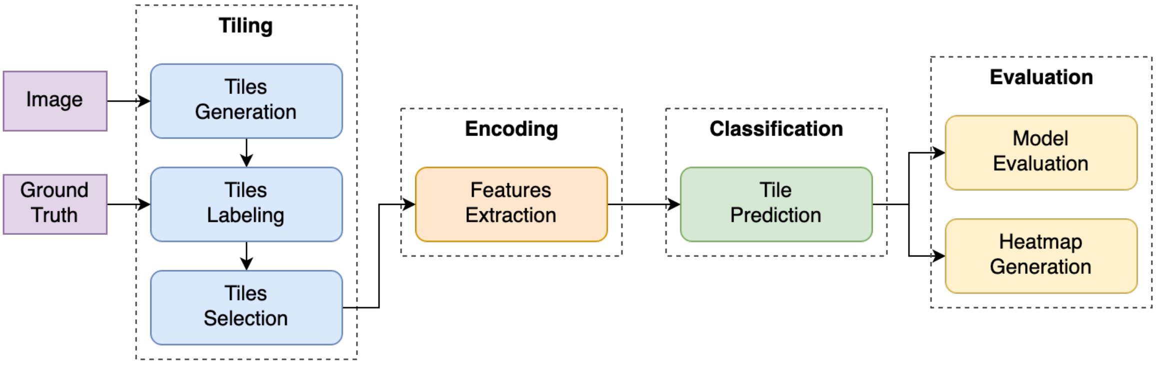


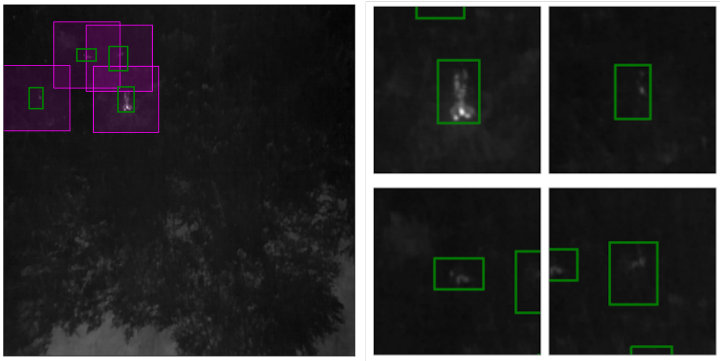
Figure 3.6: General pipeline for training and evaluating anomaly detection models.

Tiling

From each image in the data set, a set of square tiles is extracted and for each tile a label (*background*, *anomaly*) is assigned based on the presence or absence of targets. The tile dimension defines the amount of context that is used to extract textural features. Since the descriptive capability of the features can be strongly influenced by the amount of information given to the feature extractor, the tile size is a parameter that has been explored during the experimental phase.

Since the anomaly detection task is approached as a classification problem, it is essential to generate a precise set of annotated images that can be used to train the classification models. The only ground truth available in the anomaly detection data set (discussed in Section 3.1) is the set of bounding boxes enclosing the visible targets on the ground. To reduce the confusion in the tiles data set and prevent the generation of outliers due to incorrect annotation of some uncertain tiles, two distinct strategies have been adopted to generate *background* and *anomaly* tiles.

To extract *anomaly* samples, one tile of the required size is centered on each bounding box defined in the ground truth. If a target is too close to the image border, the tile is shifted to the minimum amount of space required to keep the entire tile area inside the image boundaries. In general, an *anomaly* tile could contain more than one target if there is a group of nearby people in an image. The influence of the number of targets in a tile has been studied in Chapter 4. This generation method allows obtaining the same number of tiles independently from the selected tile size, enabling a reasonable performance comparison for different tile sizes. Figure 3.7.a shows an example of anomaly tiles on the original image while Figure 3.7.b shows the extracted anomaly tiles.


(a) original image

(b) extracted anomaly tiles

Figure 3.7: Example of anomaly tiles on the original image (a) and the set of extracted anomaly tiles centered on each target in the image (b).

On the other side, *background* tiles are extracted using a sliding window approach with a stride that is a percentage (25%, 50%, 75%) of the tile size. Overlapping tiles prevent the loss of information that could happen when using a static grid and offer more data to train anomaly detection models. All the tiles that intersect any bounding box in the ground truth, even if partially, are discarded to avoid having outliers (i.e. tiles with a visible target) in the *background* class of the training set.

These generation strategies reduce any possible noise and ensure that anomaly tiles effectively contain at least one visible target, while normal tiles are empty and depict only background areas.

The major issue of this approach for generating *background* tiles is that the number of extracted tiles from an image can vary greatly based on the selected tile size, e.g. 9 tiles are generated with a size of 240 pixels and a stride of 50% whereas 81 tiles are extracted with a size of 96 pixels and the same stride. With smaller sizes the method generates many more *background* tiles so the ratio with *anomaly* tiles changes considerably (same number of *anomaly* tiles is generated for all the sizes). Therefore, the results of different experiments would be incomparable and with a largely imbalanced data set the classification model could underperform. The solution adopted to solve this problem and generate the same number of *background* tiles for every selected size is to define a maximum tile size and apply the sliding window with that size. Each extracted tile is cropped at the center to the required size. Figure 3.8 shows an example of tiles extracted from an image. Figure 3.8.a is a representation of the sliding window used to generate the candidate *background* tiles, while Figure 3.8.b highlights the final cropped tiles generated from the image.

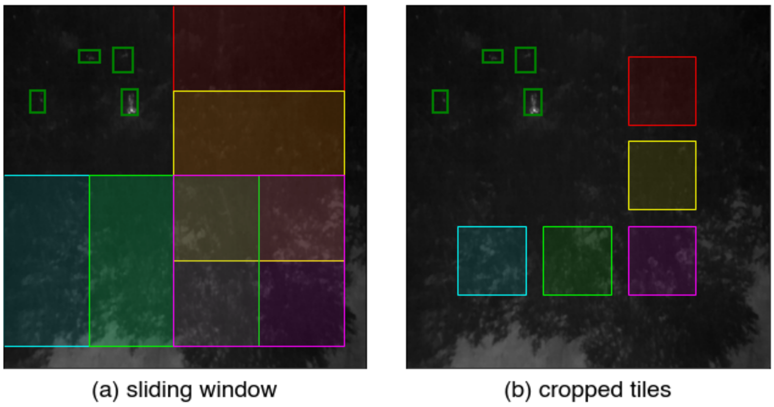

(a) sliding window (b) cropped tiles

Figure 3.8: Candidate normal tiles (a) generated using a sliding window of 240x240 px. Actual normal tiles extracted (b) with a size of 96x96 px.

This way, all the models are trained and tested on the same number of samples. The only difference between various experimental configurations is the amount of context given to the feature extractor. Figure 3.9 shows the different contexts around each target due to the variation of tile size. It is possible to notice that a tile size of 96x96 pixels (corresponding to 6x6 meters) covers mostly one target while 240x240 pixels (14x14 meters) depicts a very large area with multiple targets.

Traditional feature extraction techniques require additional steps to post-process and

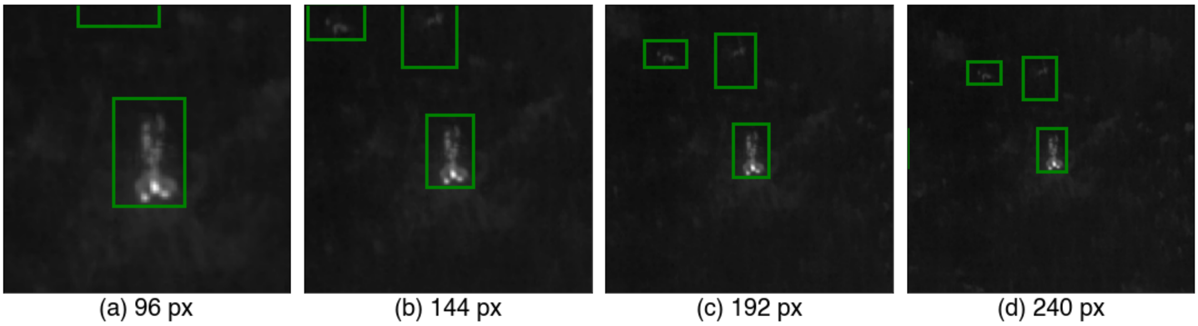


Figure 3.9: Variation of context around a target depending on the selected tile size.

select the extracted features before feeding them to the Machine Learning classifier. Figure 3.10 shows in detail the image encoding phase, composed of multiple procedures to extract features from an input image, normalize their values range, and select the most distinctive features fed to the downstream classifier.

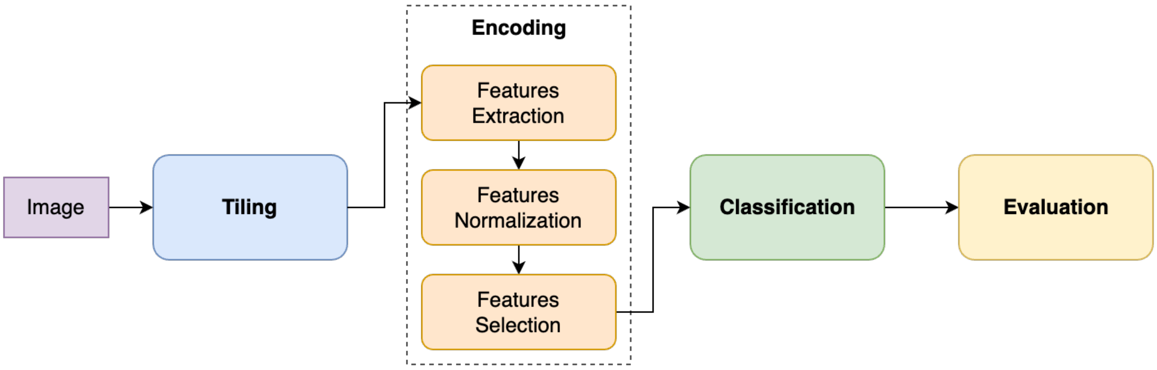


Figure 3.10: Pipeline for Machine Learning models with a multi-step encoding phase.

Feature Normalization

Most feature extraction methods generate features with a wide range of values and different magnitudes of scale. If a feature has a range of values that is orders of magnitude larger than others, it might dominate the objective function of the classifier, making it unable to learn correctly from other features. Therefore, in these cases, it is necessary to apply some normalization on the extracted features to prevent the classifier from relying only on the most dominant features. All extracted features should have the same order of magnitude to contribute equally to learning the decision boundary of the classifier. For this reason, a *min-max normalization* is applied channel-wise to project all the features in the range of values [0,1].

Typically, in Machine Learning tasks, features are normalized over the entire set that should represent the distribution of the data and, thus, the values of the features. In the data set under analysis, though, due to the high variability of the pixel intensities caused by the thermal camera, feature value ranges can vary greatly from one image to another. The solution to this problem is to perform feature normalization on the features extracted from the tiles of each image. This approach can be seen as a batch normalization where each batch comprises tiles from the same image. This method can reduce the influence of the high variability of the images in the data set while allowing to obtain normalized features that have the same range of values for all the channels and thus contribute equally to the learning process of the classifier.

Feature Selection

Most Machine Learning anomaly detection models (e.g., OC-SVM or Isolation Forest) could underperform with very high-dimensional data due to poor computational scalability and curse of dimensionality [22, 121, 122]. Moreover, training simple Machine Learning classifiers with too many features can introduce noise that might induce a model to overfit even with optimal parameter selection [151]. For this reason, it is good practice to reduce dimensionality by selecting only the most important features before training a classification model.

The strategy adopted in this work is to maintain the best discriminative features based on univariate statistical tests. The features extracted from the training samples are evaluated with the *ANOVA* (Analysis of Variance) statistical method to check the similarity with each other and assign them a score. Then, the k highest scoring features are kept, i.e., the set of features that have high variances and are not similar to each other.

3.3. Machine Learning Techniques

Machine Learning methods are composed mainly of two phases: feature extraction and feature classification. During the feature extraction phase, a general method extracts a compact representation of the image, which can be used to perform many Computer Vision tasks such as pattern recognition or image classification. Instead, in this work, the feature classifier is an anomaly detection algorithm that classifies features as normal or anomalous. Section 3.3.1 presents the implemented feature extractors, while Section 3.3.2 discusses the anomaly detection algorithms exploited in this work.

3.3.1. Feature Extraction

Computer Vision researchers have proposed many feature extraction techniques [58] but from all the methods presented in Section 2.1.2, only a subset has been implemented and analyzed in this work. After researching the state-of-the-art feature extraction techniques, some methods have been initially discarded because they do not have the invariance properties required for anomaly detection from drone imagery. As an example, HOG [31] was specifically designed for object recognition tasks where targets have features characterized by specific orientations. Consequently, in a rescue operation scenario, occlusions or people with articulated poses could prevent the detector from identifying a person [8]. In their original variants, other methods, such as LBP [104] or Wavelet Transform [86], are inappropriate since they are not rotation-invariant, which is important in drone-captured images.

The feature extraction techniques that have been implemented and tested are: Haralick features [52], SIFT [94], HGM [138] and LETRIST [143]. Moreover, a method based on the Histogram of Pixel Intensities has been implemented as a baseline.

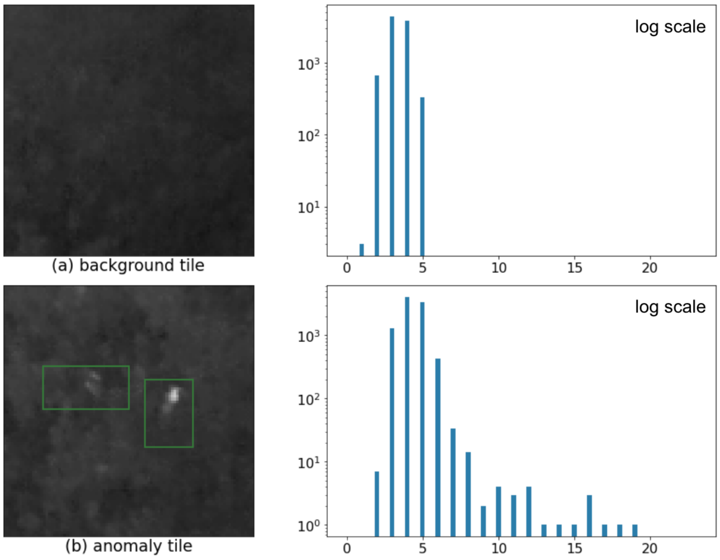
Baseline: Histogram of Pixel Intensities

The baseline for anomaly detection is a statistical method based on analyzing the distribution of gray-level values. Given an input image, the intensities of all the pixels are quantized into a small number of ranges, and the frequencies of the occurrences are counted using a histogram. An example is shown in Figure 3.11. This simple method considers only the values of individual pixels and not their relations to neighboring pixels; therefore, the histogram-based descriptor can only represent global information while no spatial or textural information is captured. The generated image descriptor is invariant to translation and rotation but is affected by noise and illumination changes. The histogram is normalized using the *min-max* normalization before being fed to the classifier.

Haralick

Haralick features [52] is a global texture descriptor that is composed of a set of several statistics computed from a Gray Level Co-occurrence Matrix (GLCM).

The GLCM is a square matrix with a size corresponding to the number of gray levels values in the image and, as the name suggests, it counts the co-occurrences of gray levels in the image. It is computed by considering the relation of each pixel with a neighbor pixel defined by a displacement vector. For each pixel in the image, the gray-level is compared



(a) background tile

(b) anomaly tile

Figure 3.11: Examples of background tile (a) and anomaly tile (b) with their respective histograms of pixel intensities *(values are in log scale)*.

with the value of the pixel at a certain parametric distance d on a specific orientation θ. The co-occurrence is computed in 4 directions $\theta = \{0°, 45°, 90°, 135°\}$, thus one GLCM is generated for each orientation (horizontal, right diagonal, vertical, left diagonal axis). Using a displacement vector $\delta = (d_x, d_y)$ with module d and orientation θ, a GLCM G is formally computed with Equation 3.1 where $I(x, y)$ indicates the pixel value of image I in position (x, y) and i, j are the gray levels that occur as neighbors. An example of GLCM construction is shown in Figure 3.12.

$$G(i,j) = \sum_{x=1}^{w}\sum_{y=1}^{h} \begin{cases} 1, & \text{if } I(x,y) = i \text{ and } I(x+d_x, y+d_y) = j \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

From the 4 generated matrices, a set of statistics are computed to characterize the distribution of gray-level co-occurrence in each of the four directions. To obtain rotation-invariance, the derived features are averaged over all four directions. The extracted Haralick features correspond to 14 statistics that are computed from the GLCM: Angular Second Moment, Contrast, Correlation, Sum of Squares (Variance), Inverse Difference Moment, Sum Average, Sum Variance, Sum Entropy, Entropy, Difference Variance, Difference Entropy, Information Measure of Correlation 1, Information Measure of Correlation 2, Maximal Correlation Coefficient. The complete mathematical formulations for all the features are fully described in the original paper [52]. Usually, in most implementations, the last feature (Maximal Correlation Coefficient) is discarded because it is considered

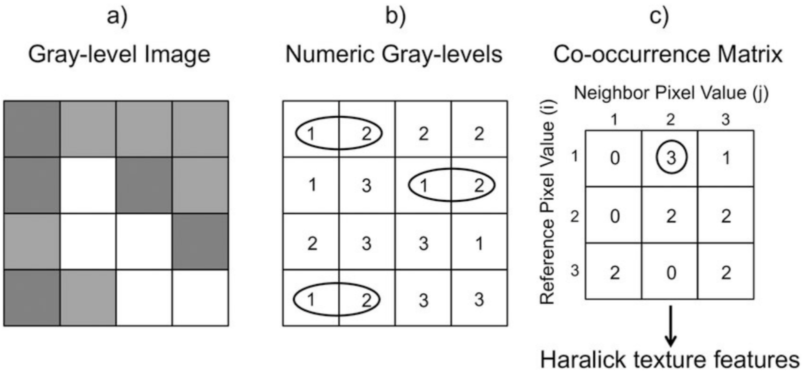


Figure 3.12: Example of GLCM calculation. Grayscale image (a) with the corresponding gray-level pixel intensities (b). GLCM (c) is computed by counting the number of occurrences of a pixel with value i having a neighbor with value j. Figure taken from [34].

unstable, thus obtaining a feature vector of 13 dimensions.

SIFT

Scale-Invariant Feature Transform (SIFT) [94] is an algorithm to detect and describe local features in images. The features are invariant to image scale and rotation. Moreover, they are robust to affine distortion, viewpoint change, addition of noise, and illumination change.

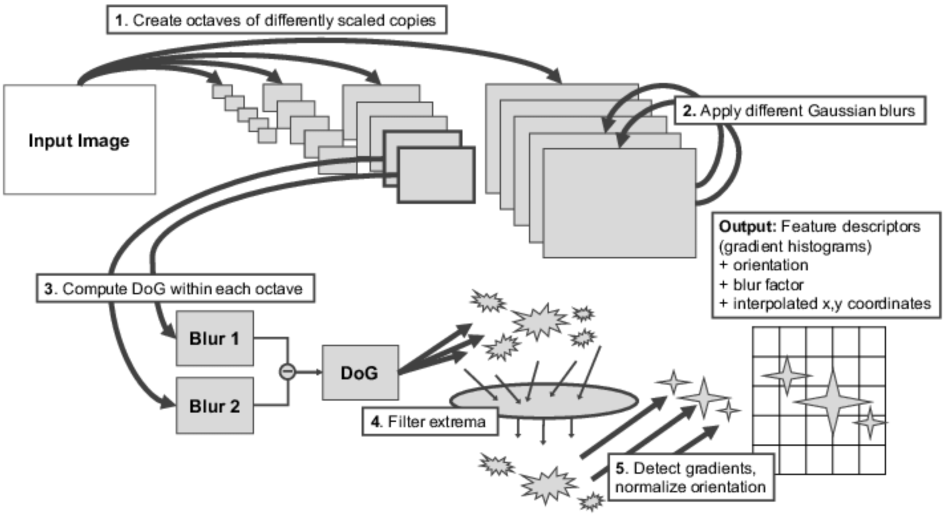

1. Create octaves of differently scaled copies
Input Image
2. Apply different Gaussian blurs
Output: Feature descriptors (gradient histograms)
+ orientation
+ blur factor
+ interpolated x,y coordinates
3. Compute DoG within each octave
Blur 1
Blur 2
DoG
4. Filter extrema
5. Detect gradients, normalize orientation

Figure 3.13: Steps of the SIFT algorithm. Figure taken from [114].

The SIFT algorithm can detect keypoints at multiple scales by using scale-space filtering that is produced from the convolution of Gaussian kernels at various scales σ. The steps of the algorithm, shown in Figure 3.13, are the following:

1. The scale-space is divided into octaves and the image size is halved in each octave;
2. Within each octave, the images are blurred by the convolution with Gaussian kernels with increasing σ scales;
3. Then, the Difference of Gaussians (DoG) is generated by computing the difference of Gaussian blurring of an image with two different scales;
4. The candidate keypoints are identified as the local extrema of the DoG. The identified keypoints are then filtered based on some conditions and only the most relevant ones are kept;
5. From the image region around each keypoint, the gradient magnitude and direction are computed and an orientation histogram, weighted by the gradient magnitude, is created. To achieve invariance to image rotation, the highest peak in the histogram, along with any peak above 80% of its value, is considered to calculate the orientation. This approach creates keypoints with the same location and scale but with different directions to increase the robustness to rotations.

For each keypoint with a known location, scale, and orientation, a descriptor of the local image region is computed. The 128-dimensional descriptor is composed of gradient histograms computed from the sub-blocks of the local region around the keypoint.

The information represented by a single descriptor is not enough to discriminate a keypoint associated with a background area from another located on a target. For this reason, the implemented method uses the keypoints extracted from an image to build a descriptor that is then classified to detect anomaly samples. The training process consists of extracting SIFT keypoints from all the normal images in the training set (that contain background only) and the keypoints that represent the same visual feature are grouped together with a clustering algorithm to identify the most relevant normal keypoint descriptors that characterize the class of normal images. The vocabulary is learned by using the *K-means* clustering algorithm which determines the centroids for all the identified clusters which are used as visual words of the dictionary. During inference, SIFT keypoints are detected from each image but, since in some cases a large number of descriptors can be generated, they are randomly downsampled to a maximum number. The subsampling is required to avoid generating a histogram of word counts with unusual distribution when many keypoints are detected. The *Visual Bag-of-Words* (BoW) is generated by associating each keypoint to the nearest visual word in the vocabulary and building a histogram by counting the occurrences of each word. The resulting histogram is used as a feature vector of the image and has a dimension equal to the number of visual words. Examples of tiles with the associated Bag of Words descriptors are shown in Figure 3.14.

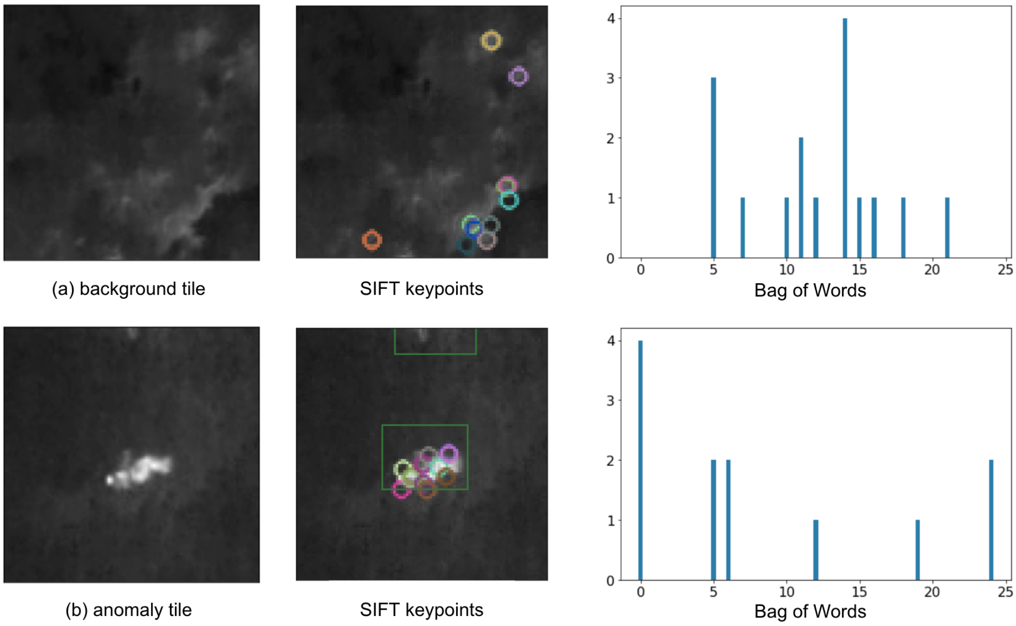


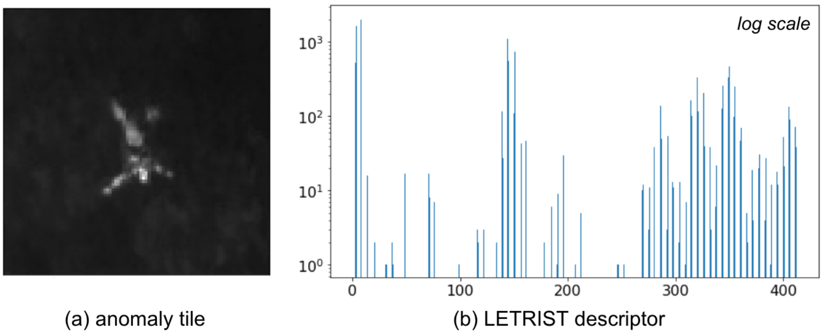
Figure 3.14: Examples of background tile (a) and anomaly tile (b) along with the detected SIFT keypoints and the respective Bag of Words descriptors.

Usually, this approach is implemented in supervised classification tasks where keypoints from all the classes are available during training and, therefore, the clustering step can build clusters that are representative of all the classes. In the case of anomaly detection, only normal keypoints are available, thus the expectation is that anomaly images should have different histogram distributions with respect to the histograms generated from normal images.

LETRIST

LETRIST (Locally Encoded TRansform feature hISTogram) [143] is an image descriptor for texture classification which characterizes local texture structures and their correlation. The extracted features are robust to rotation, illumination, scale, viewpoint changes, and also to Gaussian noise. LETRIST is a histogram representation that explicitly encodes the joint information within an image across feature and scale spaces. Figure 3.15 shows an example of an anomaly tile with the associated LETRIST descriptor.

Before extracting features, LETRIST applies standard normalization to the input image to have zero mean and standard deviation. This normalization is useful to remove global affine illumination changes.



(a) anomaly tile

(b) LETRIST descriptor

Figure 3.15: Example of anomaly tile (a) and associated LETRIST descriptor (b), *(values are in log scale)*.

The image is convolved with a set of first and second directional Gaussian derivative filters to compute the extremum (maximum and minimum) responses in scale space. The extremum filtering step is introduced to capture the information contained in the first-order and second-order differential structures and to extract local features that are rotation-invariant. According to the theory of steerable filters [40], any orientation of the first or second derivative of a Gaussian can be synthesized by taking a linear combination of several basis filters. The five basis filters used to define the image derivatives are G_x, G_{xx} which are respectively the scale-normalized first and second derivatives of G along the x-axis and similarly for G_y, G_{xy}, G_{yy}.

Given an image I, the responses of the first and second Gaussian derivative filters at orientation θ are: $I_1^\theta = G_1^\theta * I$ and $I_2^\theta = G_2^\theta * I$. Then, the extremum response of the first derivative $I_{1,max}^\theta$ is the maximum value of I_1^θ over all θ and similarly, the extremum responses of the second derivative $I_{2,max}^\theta$, $I_{2,min}^\theta$ are the maximum and minimum values of I_2^θ over all θ. To capture multi-scale feature properties, the extremum responses are computed at different scales which in the original formulation are $\sigma = \{1, 2, 4\}$.

On the obtained extremum responses, a set of linear and non-linear operators are applied to construct a set of transform features:

- gradient magnitude g (3.2): maximum response of the first directional Gaussian derivative filter;
- extrema difference d (3.3): the difference between maximum and minimum responses of the second directional Gaussian derivative filter;
- shape index s (3.4): provides a quantitative measure of local second-order curvature (e.g. caps, ridges, saddles, ruts, and cups);
- mixed extrema ratio r (3.5): captures the correlation of the first-order and second-

order differential structures where c is a scale factor to adjust the ratio.

$$g = I^{\theta}_{1,max} \tag{3.2}$$

$$d = I^{\theta}_{2,max} - I^{\theta}_{2,min} \tag{3.3}$$

$$s = \frac{1}{2} - \frac{1}{\pi} \arctan\left(\frac{I^{\theta}_{2,max} + I^{\theta}_{2,min}}{I^{\theta}_{2,max} - I^{\theta}_{2,min}}\right) \tag{3.4}$$

$$r = \frac{2}{\pi} \arctan\left(c \cdot \frac{d}{g}\right) = \frac{2}{\pi} \arctan\left(c \cdot \frac{I^{\theta}_{2,max} - I^{\theta}_{2,min}}{I^{\theta}_{1,max}}\right) \tag{3.5}$$

The constructed transform features, denoted as $F = \{g, d, s, r\}$ are rotationally invariant because are derived from extremum responses generated by the convolution with steerable filters that are invariant to rotation. The computed transform features are quantized into discrete texture codes via scalar quantization. Transform features $\{g, d\}$ with non-negative values, are quantized with a binary threshold based on the mean value of the respective transform feature map which is robust to image rotation. Transform features $\{s, r\}$ with values in the range $[0, 1]$, are quantized using a uniform quantizer with multi-level thresholding.

Finally, the texture codes are jointly encoded across scales to build multiple histograms which are concatenated to form the image feature representation. Transform features $\{g, d, s\}$ are jointly encoded across two adjacent scales (e.g. $(\sigma_1, \sigma_2), (\sigma_2, \sigma_3)$) while the transform feature $\{r\}$ is encoded across all the scales. The final feature representation is a 413-dimensional image descriptor.

HGM

Histogram of Gradient Magnitudes (HGM) [138] is a low-dimensional, rotation-invariant local texture descriptor. The feature descriptor is based on the gradient magnitude of pixel intensities that give the amount of the difference between pixels in the neighborhood, indicating the edge strength.

The gradient of an image measures how the image content is changing and provides two pieces of information: the magnitude of the gradient measuring how quickly the image changes and the gradient orientation indicating the direction in which the image is changing most rapidly. HGM is rotation-invariant because it computes the histogram using only the gradient magnitudes of pixels, ignoring the gradient orientations.

Firstly, a Gaussian blur of kernel size 3 is applied to an image for noise removal and then the image is converted to grayscale. The derivatives of the image I are calculated in x and y directions by convolution with Sobel operators as shown in Equations 3.6.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \tag{3.6}$$

The gradient magnitude is computed from the derivatives with Equation 3.7.

$$G_{mag} = \sqrt{G_x^2 + G_y^2} \tag{3.7}$$

The last step is to calculate the normalized histogram over 16 bins of all pixels magnitude values.

The image descriptor is a 16-dimensional histogram of gradient magnitudes. Since the computed histogram is already normalized with values in a small range, the normalization of extracted features is not required.

Figure 3.16 shows two examples of tiles with a visualization of their gradient magnitudes. Figure 3.16.a shows a background tile that has high variability of the gradient magnitudes but is restricted in a limited range (the upper bound is 77.8). Therefore the gradients are distributed on many bins of the associated HGM descriptor. Figure 3.16.b shows an anomaly tile with a weakly visible target. Despite the target being almost completely hidden, the magnitudes on its borders are much higher with respect to the gradients of background areas. The magnitude range is wider (the upper bound is 135.8) and therefore most of the gradients computed on background areas are concentrated in the lower bins of the associated histogram.

3.3.2. Feature Classification

Two popular anomaly detection classifiers have been analyzed: OC-SVM [134] and Isolation Forest [90].

One-Class SVM (OC-SVM)

OC-SVM [134] is an effective classifier often used for anomaly detection tasks where rare occurrences of outliers need to be identified. OC-SVM adapts the well-known supervised classification Support Vector Machine (SVM) model [113] for the unsupervised anomaly detection task. The SVM method aims to find a hyperplane that maximizes the margin

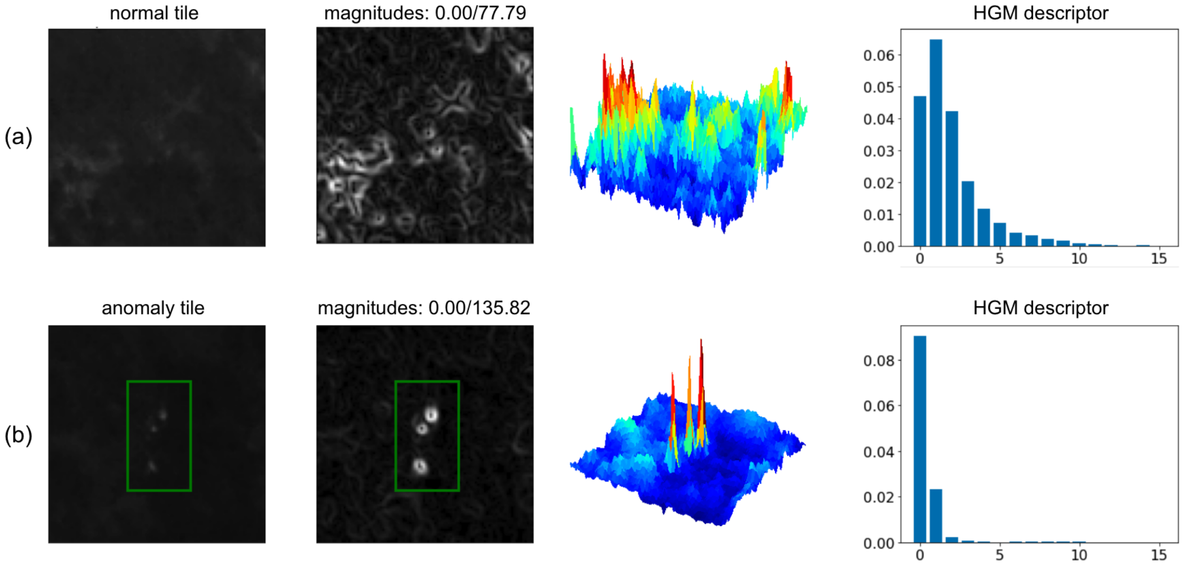


Figure 3.16: Examples of normal tile (a) and anomaly tile (b) along with their respective plots of the gradient magnitudes and the associated HGM descriptors.

separation between different classes. However, in one-class problems, the information regarding the anomaly class instances is not available. To deal with this issue, OC-SVM separates inliers (normal instances) from the outliers by finding a hyperplane that maximizes the boundary from the origin, i.e. all the observations with low similarity with respect to the training data, as shown in Figure 3.17. Similarly to SVM, OC-SVM can generate a non-linear boundary by using specific kernels: *Radial Basis Function (RBF)*, *Polynomial*, or *Sigmoid*. The hyperplane in the kernel space induces a non-linear surface in the feature space that allows the model to learn complex non-linear normal data distributions.

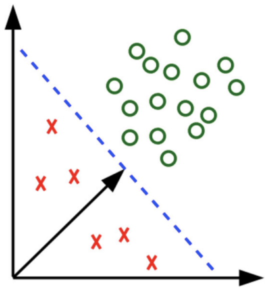


Figure 3.17: OC-SVM defines a hyperplane to separate the anomalies (red crosses) from the normal distribution (green circles). Figure taken from [106].

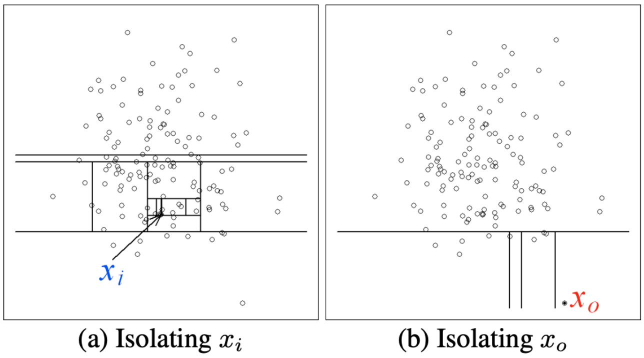
Another important aspect of OC-SVM is the ν hyper-parameter with values inside the interval $\nu \in (0, 1]$ which defines an upper bound on the fraction of training samples that are allowed to be wrongly classified (considered as outliers by the decision boundary) and a lower bound for the number of samples that are support vectors. This is known as the ν-property that allows incorporating a prior belief into the model about the fraction of outliers in the training set. Higher values allow for a greater number of wrongly classified training samples, reducing the risk of overfitting while providing better generalization. This parameter is a regularization term that can be used to fine-tune the trade-off between overfitting and generalization.

After training the OC-SVM on normal data, the model can be used to classify new unseen instances and detect anomalies through a decision function that outputs a discrete value: $+1$ if the input is a normal data point, -1 for an anomaly sample. A possible limitation of the OC-SVM that must be addressed during training is that, even with optimal parameter selection, it can be sensitive to overfitting in the presence of noise [139]. Moreover, when the training data is noisy and contains many outliers near or in the normal class, the OC-SVM will estimate a large boundary that encloses areas of the feature space where the normal class has low density, resulting in many False Negatives.

Isolation Forest (IFOR)

Isolation Forest [90] is a popular unsupervised Machine Learning algorithm for detecting anomalies within a data set. It is a model-based unsupervised outlier detection method that isolates observations by randomly selecting a feature and then splitting the data between the maximum and minimum values of the selected feature. The process is repeated until all possible splits have been made or a limit on the number of splits is reached. Recursive partitioning can be represented by a tree structure and an ensemble of similar decision trees (on different subsets of features) is used to isolate anomalies from the rest of the data, as shown in Figure 3.18.

The path length from the root node to the final node reached by a sample is equivalent to the number of splittings required to isolate it from the rest of the data. Assuming that anomalies have attribute values that differ from normal data, random partitioning produces noticeably shorter paths for anomalies that are more likely to be separated in early partitioning. Hence, when a forest of random trees collectively produces shorter path lengths for particular samples, they are highly likely to be anomalies. The average path length over a forest of random trees is a measure of normality and can be used to separate normal and anomaly samples. Isolation Forest is not based on distance or density measures to detect anomalies, thus eliminating the major computational cost of distance calculation in all distance-based and density-based methods. Moreover, this technique



(a) Isolating x_i (b) Isolating x_o

Figure 3.18: Examples of Isolation Forest splittings. Isolating a normal point x_i from the normality distribution requires many splittings (a). On the other hand, isolating an anomaly point x_o can be performed with a few splits (b). Figure taken from [90].

can handle high-dimensional problems containing many irrelevant attributes.

3.4. Deep Learning Techniques

Many Deep Learning anomaly detection architectures have been proposed in the literature [100]. Some networks can obtain state-of-the-art results on challenging tasks but are characterized by complex architectures, thus becoming computationally expensive.

Thanks to the recent advance of anomaly detection in Computer Vision applications, many methods have been introduced, such as GANomaly [4], AnoGAN [132] and ALOCC [124]. Although these methods are specifically designed for anomaly detection, they have some major limitations in the anomalies that they can identify. For example, the ALOCC and GANomaly methods assume a substantial difference exists between the concept of normal data and anomaly samples. This assumption is reasonable for anomaly detection in natural images. However, when searching for a target from UAV imagery, the difference between background images and frames containing targets is minor because only a small area of the image is influenced by the presence of a target [3].

Moreover, the application scenario requires performing anomaly detection as fast as possible on a large number of images to quickly identify victims. Therefore, developing a lightweight model with few parameters is preferable to ensure real-time performance. Consequently, an architecture based on the original **LeNet-5** network [83] has been implemented.

3.4.1. Architecture

Deep Learning models trained for binary classification tasks output a score in the range [0,1] that indicates the probability that the input belongs to one of the two classes. In an anomaly detection setting, approached with a classification task, a network predicts an anomaly score that measures the degree of abnormality of the input and indicates the likelihood of being an anomaly. The computed anomaly score is unbounded, therefore this value must be thresholded to determine the class of the input tile, as shown in Figure 3.19. The definition of the anomaly threshold for the tile classification step may change for the various models. Some models have a built-in mechanism to determine if an input is anomalous, whereas for others a proper anomaly threshold should be defined.

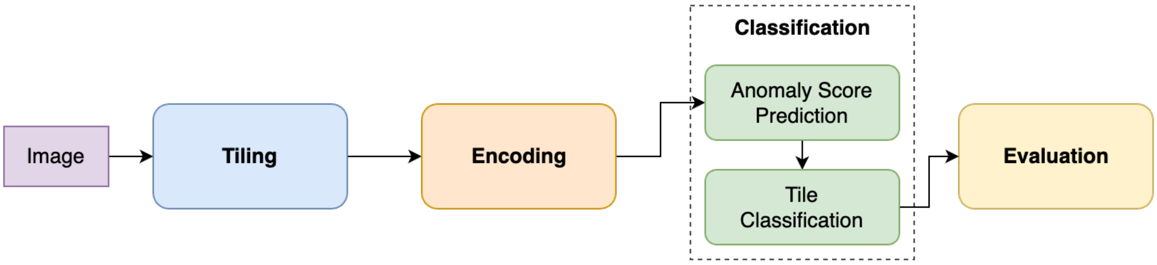


Figure 3.19: Pipeline for Deep Learning models with a multi-step classification phase.

LeNet-5 is one of the earliest convolutional neural networks trained to recognize handwritten numbers from small images of 32x32 pixels [82]. The original network comprises seven layers with three initial *Convolutional* layers to extract spatial features with *tanh* activation function, separated by *Average Pooling* layers to subsample and reduce the spatial dimensions. After the feature extractor, a *Fully Connected* network is used as a classifier with an output layer of 10 units to classify the number recognized from the input image. Despite the simple architecture with few layers, LeNet-5 has been used in many works in recent years [59, 88]. Adaptations of this network are implemented in many object recognition or image classification tasks, meaning that despite its simplicity and reduced parameters, it can learn useful and discriminating features.

In this thesis, major modifications have been applied to the original LeNet-5 model: the *Average Pooling* layers have been replaced by *Max Pooling* layers that introduce stronger non-linearity to the model, the *tanh* activation function have been substituted by the more efficient *Leaky ReLU* activation function and, finally, an additional *Convolutional* layer has been included to deal with bigger input images and reduce the number of trainable parameters. Figure 3.20 shows the final architecture.

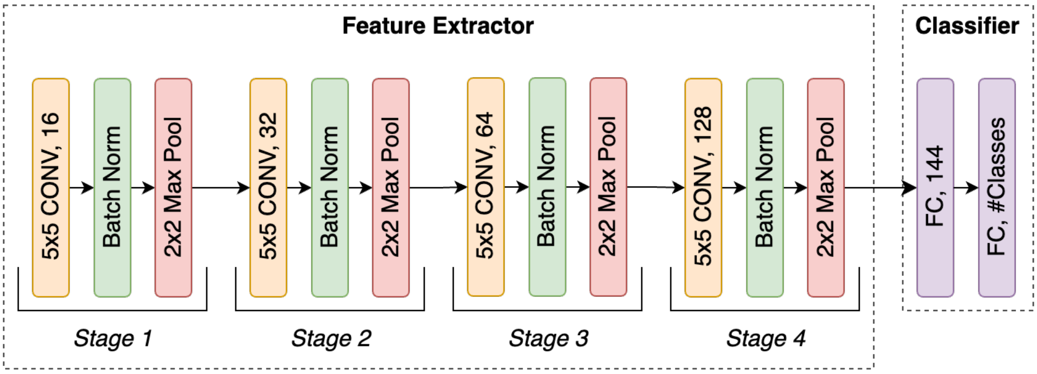


Figure 3.20: LeNet-type convolutional neural network architecture composed of a *feature extractor* and a *classifier*. The output layer has a variable number of neurons that depends on the number of classes defined by each task.

The implemented LeNet-type architecture mainly comprises a *feature extractor* composed of 4 similar convolutional stages which build a compact representation of the input image and a *classifier* which classifies the latent representation based on the specified task. The network input layer has a shape of 96x96x1 because it processes square tiles extracted from a grayscale thermal image.

Each stage of the feature extraction part is composed of the concatenation of three layers which are a *Convolutional* layer, a *Batch Normalization* layer, and a *Max Pooling* layer. The *Convolutional* layer has a kernel size of 5x5 and padding to maintain the input spatial dimension. The output of the *Convolutional* layer is normalized with the *Batch Normalization* layer to obtain a set of features with zero mean and standard deviation, followed by *Leaky ReLU* activation function. *Leaky ReLU* is a variation of the *ReLU* function with a small slope for negative values, instead of fixing them to 0, which allows back-propagation also for negative values of the activation function. The final *Max Pooling* layer has a kernel size of 2x2 and a stride of 2 pixels, halving the resolution of the feature maps. The number of filters in each *Convolutional* layer is doubled in each stage, starting from 16 filters in the first layer and 128 filters in the last. The output of the feature extractor subnetwork is flattened and connected to a *Fully Connected* layer of 144 units which is the latent representation of the input image. The final *Fully Connected* layer size depends on each specific task.

Each input tile is individually preprocessed with *Global Contrast Normalization (GCN)* [47] to normalize the contrast across the entire data set. By normalizing the amount of contrast in each tile, this preprocessing step removes a source of variation across the data samples. In the context of Deep Learning, the contrast of an image refers to the standard deviation of the pixels in the image. GCN aims to prevent images from having varying amounts of contrast by subtracting the mean from each image, then rescaling it so that the standard deviation across its pixels equals the unitary value. Given an input image $X \in \mathbb{R}^{r \times c}$, GCN produces a normalized output image X' as defined by Equation 3.8.

$$X'_{i,j} = \frac{X_{i,j} - \overline{X}}{\sqrt{\frac{1}{rc} \sum_{i=1}^{r} \sum_{j=1}^{c} \left(X_{i,j} - \overline{X}\right)^2}} \tag{3.8}$$

The resulting pixel values are rescaled to the range [0,1] using *min-max normalization* by computing the maximum and minimum values across the entire training set.

3.4.2. Self-Supervised Learning

The encoder network described in Section 3.4.1 will be used as part of larger deep anomaly detection architectures. Since the encoder network will be used to extract relevant feature representations, there are three possibilities for initializing its weights: random initialization, transfer learning from another domain, and transfer learning from a *pretext task*. Due to the nature of the data set and the abundance of background images, the *pretext task* approach has been adopted.

Due to the intrinsic characteristics of the scenario of forest imagery captured by drones, some of the most popular Self-Supervised Learning techniques [5] cannot be adopted because they would not allow the model to learn good representative features. For example, predicting the rotation applied to an input sample is not a viable solution because the images captured from a drone do not have a specific orientation. Moreover, background images have a very high intra-class variability because the possible forest scenarios are very diversified and lack a specific structure. For this reason, reconstruction-based networks [85] or generative approaches [153] do not obtain good results and do not allow a model to learn features that properly characterize the normal data.

The implemented *pretext task* leverages the characteristics of the available data set, specifically the forest types (broadleaf, conifer, mixed) of the various flights on which the images were captured. The network architecture described in Section 3.4.1 has been adapted to solve the image classification problem by using an output classification layer with 3 units, one for each forest, and applying a *Softmax* activation function on the output. These normalized values indicate the probabilities of the input belonging to each class. The highest probability is then used to determine the predicted class of each input.

The data set used for the *pretext task* is composed only of background tiles without any visible target. However, to prevent the encoder from learning the same data used to train the anomaly detection task, every tile that overlaps for more than 30% with any tile in the training set of the anomaly detection data set is discarded. Figure 3.21 highlights two subsets of tiles extracted from the same image. Figure 3.21.a shows tiles used for the anomaly detection task, and Figure 3.21.b shows tiles used for the *pretext* forest classification task.

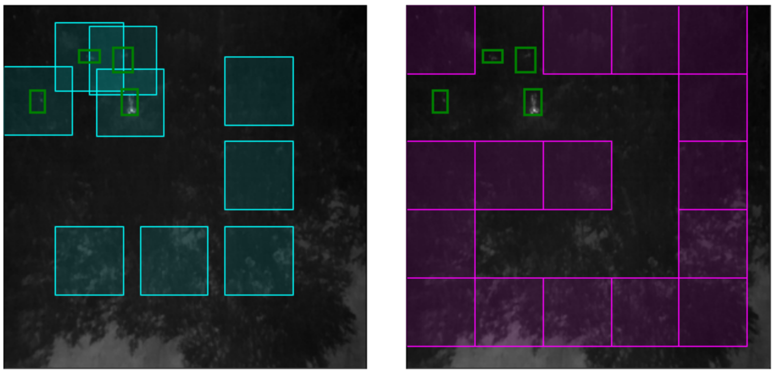

(a) tiles used for anomaly detection
(b) tiles used for pre-training

Figure 3.21: Background tiles used for the anomaly detection task (a) and background tiles used for the *pretext* forest classification task (b). The two sets of tiles are extracted from the same image, but the overlap between any pair of tiles selected respectively from the two sets is less than 30%.

Each extracted tile is annotated with a label that defines the forest type of the flight (broadleaf, conifer, mixed). All the labeled tiles are downsampled to obtain a balanced distribution of samples in each class. Since a fully-supervised forest classification task is performed on these images, the network learns relevant features of the forest textures that represent the normal class (or background) of the anomaly detection task.

After the Self-Supervised Learning pre-training, the final classification layer is dropped and the convolutional encoder network is used to initialize the weights of other anomaly detection models.

3.4.3. Unsupervised Anomaly Detection

The majority of Deep Learning approaches for anomaly detection involve networks trained to perform a different task such as image compression or generative models [4, 21, 132], which are then adapted for use in anomaly detection. The main limitation of these methods is that they are not trained on an anomaly detection-based objective [100].

A novel general approach to deep anomaly detection, *Deep SVDD*, presented in [121] is based on kernel-based one-class classification and minimum volume estimation. This model is inspired by the key concept of the Support Vector Data Descriptor (SVDD) [150] Machine Learning method which is an evolution of the OC-SVM. Although both methods define a boundary in the feature space to separate normal data from anomalies, the difference is that OC-SVM uses a hyperplane while SVDD exploits a hypersphere to enclose normal instances. Similarly to SVDD, the Deep SVDD technique trains a neural network to minimize the volume of a hypersphere that encloses the latent representations of the data to exclude anomalies, as shown in Figure 3.22. Minimizing the volume of the hypersphere, which encloses the features extracted from all the background samples, forces the network to learn the common factors of variation to map all the data points towards the center of the hypersphere.

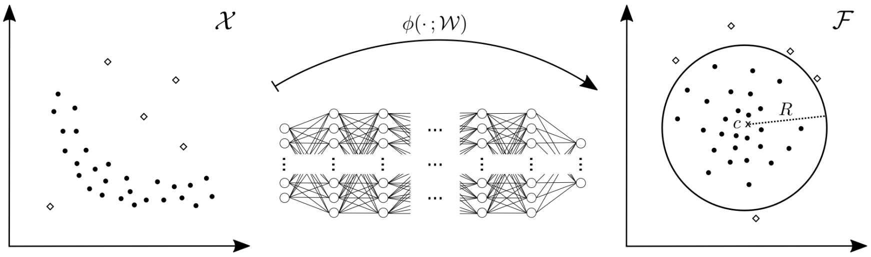


Figure 3.22: Deep SVDD learns a neural network transformation to map most of the data representations into a hypersphere of center **c** and radius R. Normal samples are mapped within the sphere, whereas anomalies are mapped outside. Figure taken from [121].

The objective of Deep SVDD is to jointly learn the network parameters $\mathcal{W}$ together with minimizing the volume of a hypersphere in the latent space that encloses the training data characterized by radius $R > 0$ and center $\mathbf{c}$. The *bias terms* $\mathbf{b}^l$ of the network layers are not used to prevent the network from learning a constant function that maps every input $x \in \mathcal{X}$ to the center, leading to hypersphere collapse. This could happen because, when using the bias terms, the network might learn to set the weights of a layer to zero ($\mathbf{W}^l = \mathbf{0}$), to obtain a constant output for every input $x \in \mathcal{X}$, and choose the bias term $\mathbf{b}^l$ (and the weights of subsequent layers) such that $\phi(\mathbf{x}; \mathcal{W}^*) = \mathbf{c}$ for every $x \in \mathcal{X}$. Consequently, the optimal solution becomes $R^* = 0$ and the hypersphere collapses.

The main advantage of Deep SVDD is that by jointly training the feature extraction network with the one-class classification objective, it can learn useful feature representations of the data and fine-tune the encoder to extract the most relevant characteristics to improve the anomaly detection performance.

The model can be trained using two objective functions: *Soft-Boundary* or *One-Class.*

Soft-Boundary

With this objective function, only the points outside the hypersphere are penalized, i.e., the samples with a distance to the center greater than the radius R. In this case, the volume of the hypersphere is minimized by minimizing R^2. The hyperparameter $\nu \in (0, 1]$ controls the trade-off between the sphere volume and the boundary violations, allowing some points to be mapped outside the sphere. Similarly to OC-SVM, it allows having outliers in the training set and the value of ν approximates the fraction of anomalous samples. Optimizing the objective function in Equation 3.9, the network learns parameters $\mathcal{W}$ such that data points are closely mapped to the center $\mathbf{c}$ of the hypersphere.

$$\min_{R,\mathcal{W}} \quad R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max\{0, \|\phi\,(x_i; \mathcal{W}) - \mathbf{c}\|^2 - R^2\} \tag{3.9}$$

Consequently, normal samples are mapped inside the sphere, whereas anomaly instances are mapped further away from the center and outside the hypersphere.

One-Class

The *One-Class* objective function, defined by Equation 3.10, is a simplified version of *Soft-Boundary* that can be used when most of the training data is assumed to be normal.

$$\min_{\mathcal{W}} \quad \frac{1}{n} \sum_{i=1}^{n} \|\phi\,(x_i; \mathcal{W}) - \mathbf{c}\|^2 \tag{3.10}$$

Differently from *Soft-Boundary*, the radius of a hypersphere is not explicitly defined with *One-Class* objective. In fact, the distance of all the sample representations to a central point in the latent space is penalized by employing a quadratic loss. Even in this case, the network must extract the common factors of variation to map data representations close to the center of the sphere.

After training, for a given test point $\mathbf{x} \in \mathcal{X}$, the anomaly score $s(\mathbf{x})$ for both variants of Deep SVDD can be defined as the distance $\phi(\mathbf{x}; \mathcal{W})$ of the point to the center of the hypersphere $\mathbf{c}$ (Equation 3.11):

$$s(\mathbf{x}) = \|\phi(\mathbf{x}; \mathcal{W}^* - \mathbf{c}\|^2 \tag{3.11}$$

where $\mathcal{W}^*$ are the network parameters of the trained model.

When the *soft-boundary* objective is used, the score can be adjusted by subtracting the radius R^* of the trained model such that anomalies mapped outside the boundary have positive scores, whereas normal data inside the hypersphere have negative scores.

For the *one-class* objective, all the predicted scores are positive numbers and there is no built-in function to differentiate between normal and anomaly samples. Consequently, a threshold should be defined from the predicted anomaly scores. The implemented solution is to compute the *Precision-Recall Curve* (PRC) that shows the trade-off between precision and recall at the variation of the anomaly threshold. The PRC is used instead of the *Receiver Operating Characteristic Curve* (ROC) because it is a better metric to assess prediction performance when the classes are imbalanced. After the computation of the PRC, a pre-established recall is selected which, in the use case of Search and Rescue missions, should be high (above 90%) to reduce the probability of missing people. The corresponding point on the curve is selected and the associated anomaly threshold is used for the final tile classification step.

3.4.4. Semi-Supervised Anomaly Detection

Typically, anomaly detection is approached as an unsupervised learning task because it is based on the assumption that only normal samples are available during training. In practice, however, in addition to a large collection of unlabeled samples that are assumed to be normal, a small subset of labeled samples may be available where each instance is annotated by some domain experts as being normal or anomalous [122]. Semi-supervised approaches aim to exploit such labeled samples. However, most methods are limited to including labeled normal samples, and only a few can take advantage of labeled anomalies.

A general end-to-end deep methodology for anomaly detection is *Deep SAD* [122], which can be considered a generalization of the unsupervised Deep SVDD model to the semi-supervised setting. This model exploits the anomaly samples directly in the objective function to better define a boundary that separates normal data and anomaly samples in the latent space.

The semi-supervised anomaly detection training set is composed of n unlabeled samples $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{X}$ with $\mathcal{X} \subseteq \mathbb{R}^D$ that are assumed to be normal and m labeled samples $(\tilde{\mathbf{x}}_1, \tilde{y}_1), \dots, (\tilde{\mathbf{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} = \{-1, +1\}$ where $\tilde{y} = +1$ denotes known normal samples and $\tilde{y} = -1$ known anomalies. If the number of labeled samples m is zero, the method is equivalent to the Deep SVDD model trained with an unsupervised data set. Similarly to Deep SVDD, also with Deep SAD removing the bias terms from the layers forces the network to learn a non-trivial solution that prevents the hypersphere collapse.

The Deep SAD objective is defined as:

$$\min_{\mathcal{W}} \quad \frac{1}{n+m}\sum_{i=1}^{n}\|\phi\left(x_i;\mathcal{W}\right)-c\|^2+\frac{\eta}{n+m}\sum_{j=1}^{m}\left(\|\phi\left(\tilde{x}_j;\mathcal{W}\right)-\mathbf{c}\|^2\right)^{\tilde{y}_j} \tag{3.12}$$

The same loss term as Deep SVDD (first part of Equation 3.12) is used for the unlabeled data, which implies that assuming most of the unlabeled data to be normal is still valid. The loss term for the labeled data (second part of Equation 3.12) is similar to the unlabeled term and minimizes the mean distance of all data representations to the center of the hypersphere. For the labeled normal samples ($\tilde{y} = +1$) the quadratic loss on the distances of the mapped points to the center is penalized to learn a latent representation that concentrates the normal data near the center. For labeled anomalies ($\tilde{y} = -1$), the *inverse* of the distances is penalized so that anomalies are mapped further away from the center. The loss term of the labeled data is weighted with the hyper-parameter $\eta > 0$ which controls the balance between the labeled and the unlabeled term on the objective loss. Setting $\eta > 1$ places more emphasis on the labeled data, while setting $\eta < 1$ emphasizes the unlabeled samples.

The anomaly score for a point $\mathbf{x}$ is computed as the distance $\phi(\mathbf{x};\mathcal{W})$ of its representation in the latent space to the center $\mathbf{c}$ of the hypersphere (Equation 3.13):

$$s(\mathbf{x}) = \|\phi(\mathbf{x};\mathcal{W}^* - \mathbf{c}\|^2 \tag{3.13}$$

where $\mathcal{W}^*$ are the network parameters of the trained model.

Similarly to Deep SVDD with *One-Class* objective, Deep SAD predicts a positive anomaly score for each input tile. An anomaly threshold is then computed from the predicted anomaly scores by selecting the point associated with a pre-established recall value on the *Precision-Recall Curve* (PRC). The associated threshold is then used to classify a tile as normal or anomaly based on the predicted anomaly score.

Semi-Supervised Training Set

In the case of semi-supervised learning, the training set is different from the data used in the unsupervised setting because some anomaly samples can be included. For this reason, the training data generation has been updated to remove the constraint of having only background data and to introduce a controlled number of anomaly samples. In particular, the Deep SAD training set comprises three subsets of data: unlabeled normal, labeled normal, and labeled anomaly. The ratio between these sets is defined by two parameters *ratio_known_normal* and *ratio_known_anomaly* which respectively control the number of labeled normal and labeled anomaly samples while the rest of the data $(1 - ratio\_known\_normal - ratio\_known\_anomaly)$ is composed by unlabeled samples that are assumed normal. Equation 3.14 shows the computation of the amount of labeled and unlabeled data based on these two parameters and the total number of normal samples (n_normal). All the normal samples used in the unsupervised training set are also used within this set and the only difference is that a subset of them is labeled $(\tilde{y} = +1)$ while the rest is unlabeled $(\tilde{y} = 0)$. Moreover, a specific quantity of anomaly instances $(\tilde{y} = -1)$ is added to the training set.

$$\begin{cases} x_{labeled\_anomaly} = \frac{ratio\_known\_anomaly}{1 - ratio\_known\_anomaly} * n_{normal} \\ x_{labeled\_normal} = \frac{ratio\_known\_normal}{1 - ratio\_known\_anomaly} * n_{normal} \\ x_{unlabeled\_normal} = n_{normal} - x_{labeled\_normal} \end{cases} \tag{3.14}$$

4 | Evaluation

This chapter discusses a quantitative and qualitative analysis of the implemented models. First, the evaluation procedures for Machine Learning and Deep Learning models are introduced. Then, all the models are compared to identify the best methods that could be useful for realistic missions. Finally, some examples of correct detection and model confusion are presented to identify the strength and weaknesses of the analyzed anomaly detection techniques. The chapter concludes with examples of anomaly heatmaps generated on images with targets and background images.

4.1. Evaluation Procedure

Anomaly detection models have been evaluated on the tiles extracted from the images in the anomaly detection data set (Section 3.1). The training set for unsupervised anomaly detection models is composed only of normal samples (background tiles) thus, all the anomaly tiles are discarded. Table 4.1 shows the distribution of anomaly and background tiles used for training and evaluating the unsupervised models. The validation set comprises 26% of anomaly tiles, while the test set has 31% of anomalous samples. The large number of available anomalies in the validation and test sets enables assessing the performance of models on targets with various degrees of occlusion.

Tile class	**Training**	**Validation**	**Test**
Anomaly	0	791	1,349
Background	6,519	2,252	2,987
Total	**6,519**	**3,043**	**4,336**

Table 4.1: Distribution of anomaly and background tiles among the data set.

The performance of a model trained with a set of hyper-parameter values is evaluated on the validation set. After the training phase, the model predicts the labels for all the samples in the validation set, and the performance is assessed by computing the *Precision*, *Recall*, and *F1-Score* metrics. For anomaly class, the *Precision* metric measures the ratio of correctly classified anomaly samples among all the predicted anomaly tiles. High

Precision indicates that the model predicts only a few background tiles as anomalies. The *Recall* metric measures the ratio of classified anomaly tiles among all the actual anomaly samples. High Recall indicates that only a small number of targets are missed by the model. The *F1-Score* metric is the harmonic mean of Precision and Recall and provides relevant information for an imbalanced data set because it better measures the incorrectly classified cases.

The selection of the best model configuration is different for Machine Learning and Deep Learning methods because the former techniques directly classify tiles, whereas deep models predict an anomaly score for each input tile without explicitly assigning a label.

For each feature extractor and Machine Learning classifier combination, the best configuration of hyper-parameter values is selected by assessing the performance on the validation set. Specifically, the configuration that reaches the highest F1-Score on the anomaly class of the validation set is selected as the best model.

For the Deep Learning models, the performance obtained by the tested hyper-parameter configurations is ranked by the Area Under Precision-Recall Curve (AUPRC) metric computed on the validation set. Then, for the configuration with the highest AUPRC, a point in the Precision-Recall Curve is selected with a good trade-off between precision and recall. Specifically, the point where the model obtains at least 90% recall of anomaly class is selected. In the Search and Rescue scenario, obtaining a high recall is crucial to reduce the chances of missing the identification of some people. At the same time, false alarms are not extremely critical because rescuers can manually inspect the highlighted areas and discard False Positives. The threshold corresponding to the selected point is used as the *anomaly threshold* for identifying anomalies. The anomaly detection performance of the model is finally assessed by computing the Precision, Recall, and F1-Score on the test set.

4.2. Quantitative Evaluation

Performance on the validation and test sets could be very different because they are composed of different distributions of forest types that have different characteristics, as discussed in Section 3.1.2. The distributions of forest types among the training, validation, and test sets are shown in Figure 4.1. The training set contains tiles from all the forest types, the validation set is composed only of mixed forest tiles and the test set contains conifer and broadleaf tiles. Since the validation and test sets are composed of two disjoint selections of forest types, the performance of a model evaluated on the two sets may be significantly different. Despite this imbalance of forest types among the data set, it was impossible to define a better split of images with a more balanced distribution of forest type, for the reasons explained in Section 3.1.2.

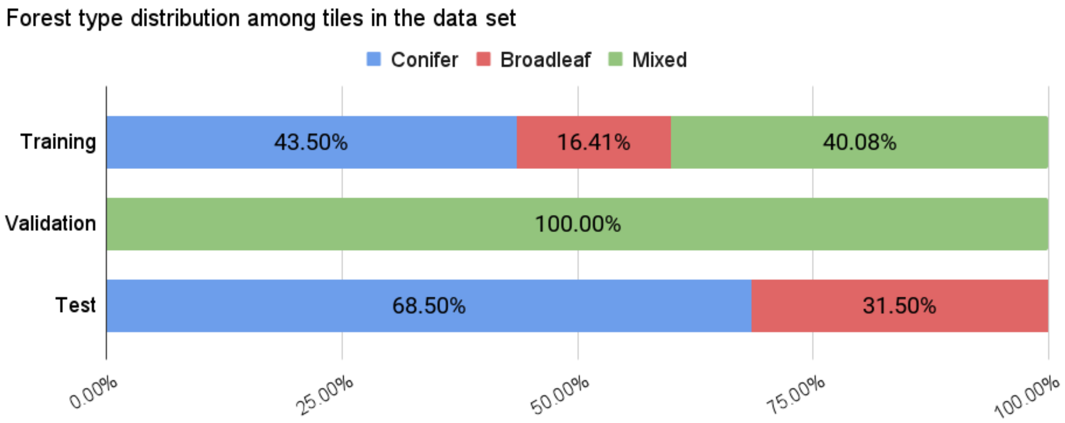


Figure 4.1: Distribution of forest types among tiles in the data set.

Machine Learning models

The hyper-parameter tuning of Machine Learning models is performed with a grid search strategy on the parameter space to guide the definition of a set of experiments. When a model relies on several parameters (high-dimensional parameter space), the adopted strategy is to run the first series of experiments with a coarse sampling on a wide range of values and then perform other runs with a more fine-grained sampling targeted on the most promising range of values.

Deep Learning models

A similar approach for the definition of experiments has also been adopted for the training of Deep Learning models. A grid search exploration of the hyper-parameter space of Deep Learning models has been conducted.

Deep Learning models have been trained for 100 epochs using a batch size of 256 to have a good trade-off between model generalization and GPU usage. It was possible to use a large batch size thanks to the small dimensions of input tiles (96x96 pixels). The initial learning rate is set to 10^{-4} and after 50 epochs it is reduced to 10^{-5} for the remaining epochs. Data augmentation techniques have been adopted to reduce the risk of overfitting, specifically vertical and horizontal random flips with a probability of 50%. Other augmentations such as random rotations are not effective for this task because the available data samples are already characterized by changes in target orientations due to the intrinsic nature of aerial images which do not have a specific direction.

4.2.1. Parameter Selection

In this section, a review of the influence of the parameters adopted by each method is presented along with a discussion of the discriminating capabilities of the representation generated by each feature extractor.

Tile size

Machine Learning models have been tested with different tile sizes to study the influence of the variation of context given to the classifier. The tested tile sizes range from 96 to 240 pixels. The upper bound on the maximum dimension is defined by the required level of granularity on the heatmaps that are generated for the target localization task. Using larger tiles would result in too coarse anomaly heatmaps that would provide imprecise localization information for identifying injured people.

One-Class SVM

One parameter, ν, has been studied for OC-SVM classifier, with values ranging from 0.05 to 0.75. Increasing the value of the ν parameter of OC-SVM allows for a greater number of wrongly classified training samples, reducing the risk of overfitting the normal samples while providing better generalization on new data. As can be seen in Figure 4.2, with low values of ν the boundary tends to overfit the training set, whereas higher values provide better generalization by defining a relaxed boundary.

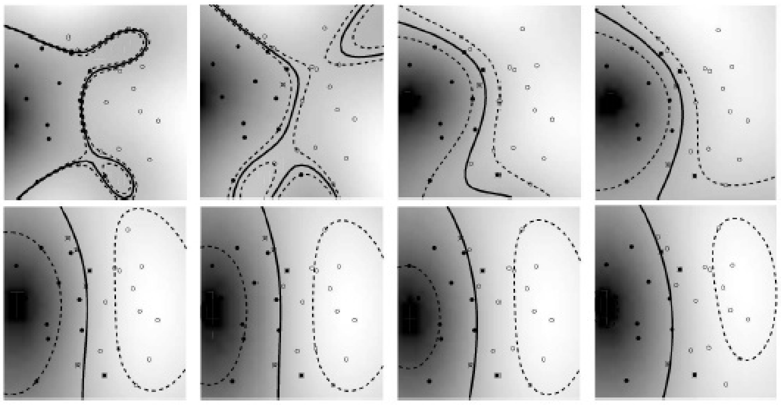


Figure 4.2: Influence of the parameter ν on the OC-SVM boundary with values from 0.1 (top left) to 0.8 (bottom right). Figure taken from [64].

Lower values of ν in the range 0.05-0.1 should provide better performance because the boundary is better fitted around the normal training data, reducing the risk of including many anomaly samples. When the training samples have high variability and cannot be modeled with the same distribution, OC-SVM tends to increase the ν to enclose as much training data as possible with the drawback of including many anomaly instances.

Obtaining the best performance using a high value of ν, confirms that the extracted features do not accurately discriminate between normal and anomaly samples. Moreover, using a high ν value may suggest that the feature extractor cannot model all the common factors of variation in normal data.

Isolation Forest

Isolation Forest is mainly characterized by three parameters: the number of trees (estimators) in the ensemble, the subset of data used for training each estimator, and the amount of contamination in the data set (proportion of outliers in the training set). The tested number of estimators ranges from 100 (default value proposed in the paper [90]) to 1,000 trees but the influence on the performance is negligible. The reason is that even a few estimators are sufficient because most instances can be easily isolated due to the high variability in the extracted features. Estimators have been trained with a variable number of samples from 20% to 80% of the training set but, also in this case, there is no significant influence on the performance. For all the methods, better results are obtained when training with 60-80% of normal samples, indicating that fewer instances are insufficient to model the normal data. This confirms that the data is characterized by high variability in the extracted features. The contamination value controls the threshold for the decision function when a scored instance should be classified. This parameter has a similar effect to the ν of OC-SVM. Contamination values in the range from 1% to 50% have been tested. Best results have been obtained with a contamination value ranging from 5% to 20%. Higher contamination values suggest strong variability in the normal data which induces Isolation Forest to consider many normal samples as anomalies.

Baseline: Histogram of pixel intensities

The only parameter of the baseline approach is the number of bins that compose the quantized histogram of pixel gray-level values. Several bins between 16 and 40 have been tested, but the effect on performance is negligible.

The tile dimension is the parameter that mostly influences the features extracted by the baseline method. Figure 4.3.a shows the feature means (the mean heights of the bins) of the anomaly and normal classes extracted from tiles of 240 px. The right-most bins (brighter pixels) have higher frequencies for anomaly instances with respect to normal samples. Therefore, the extracted features are effectively discriminating between the two classes. On the other hand, Figure 4.3.b shows that the means of anomaly features are more similar to the normal features when extracted from smaller tiles. The motivation is that there is a higher chance of including many targets when using bigger tiles. Therefore, in large tiles, the occurrences of brighter pixels (right-most bins) are much higher for anomaly tiles with respect to background ones, mostly composed of darker pixels.

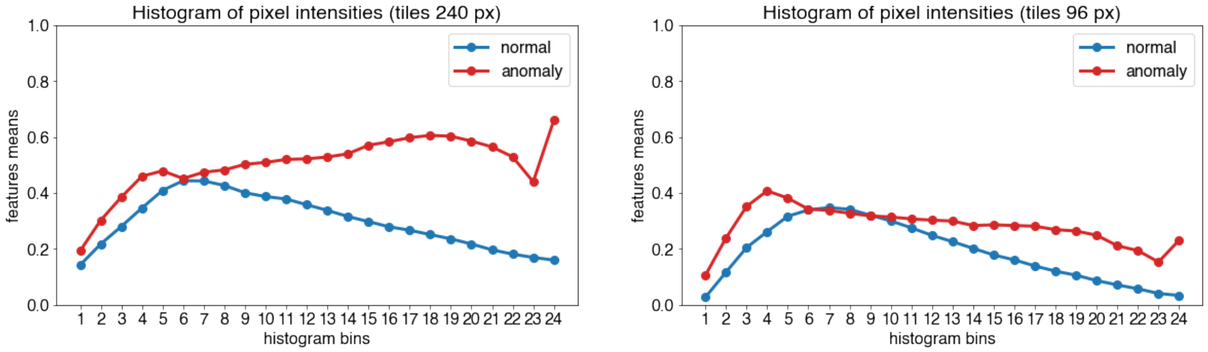


Figure 4.3: Baseline method features: mean heights of anomaly and normal histogram bins. Using bigger tiles (a), occurrences of brighter pixels *(right-most bins)* are more frequent in anomaly (red) tiles with respect to normal (blue) samples. Using smaller tile sizes (b), the pixel intensity occurrences are similar for both classes.

The best results with OC-SVM and IFOR are obtained with tiles of 216/240 pixels (the maximum size tested in the experiments). When histograms are computed from smaller tiles, OC-SVM confuses the similar pixel distributions of anomaly and normal samples, obtaining poor results, as shown in Table 4.2. On the other hand, Isolation Forest is less influenced by the dimension of the tiles and obtains similar results in both cases, as shown in Table 4.3. However, with smaller tiles, the similarity between normal and anomaly features induces Isolation Forest to mispredict some anomalies. Conversely, with the more distinctive features extracted from bigger tiles, Isolation Forest is capable of identifying more anomalies and increases the recall by 12%.

Tile size	**Bins**	ν	**Prec**	**Rec**	**F1**
240	24	0.1	64.4%	69.8%	**67.0%**
96	24	0.1	56.5%	54.6%	55.6%

Table 4.2: Influence of the tile size on the results for the baseline method with OC-SVM. Both precision and recall drop when using smaller tiles.

Tile size	**Bins**	**Num trees**	**Max samp**	**Contam**	**Prec**	**Rec**	**F1**
216	32	1,000	60%	5%	63.9%	80.7%	**71.3%**
96	32	1,000	60%	5%	69.8%	68.5%	69.2%

Table 4.3: Influence of the tile size on the results for the baseline method with Isolation Forest. When using smaller tiles, the F1-Score is similar because the decrease in the recall is counterbalanced by an increased precision.

The generalization capability of this baseline method is poor because the generated histograms are sensitive to changes in the intensity distributions which can vary for different forest types. This method relies on many nearby targets captured together in larger tiles. Consequently, performance could be much worse on another more realistic data set where only a few targets are visible.

The results on the validation set for the best configurations of the baseline method with OC-SVM and Isolation Forest are presented in Table 4.4.

Model	**Tile size**	**Bins**	ν	**Num trees**	**Max samp**	**Contam**	**Prec**	**Rec**	**F1**
OC-SVM	240	24	0.1	-	-	-	64.4%	69.8%	67.0%
IFOR	216	32	-	1,000	60%	5%	63.9%	80.7%	71.3%

Table 4.4: Best configurations of the baseline method with OC-SVM and Isolation Forest evaluated on the validation set (results of anomaly class).

Haralick features

The Haralick method has only one parameter, the *Haralick distance* which determines the neighbor pixel used to compute the co-occurrence of gray-level values. A set of experiments have been performed with distance values in the range from 1 to 21. Better results are obtained using smaller values but the influence on the performance is marginal. Another important aspect that has been tested is the selection of the subset of the most discriminating features. As can be seen from Figure 4.4, some features are similar for anomaly and normal tiles, meaning that they are not relevant and can introduce noise. Only a small subset is significantly different for the two classes and can be exploited for the detection of anomalies.

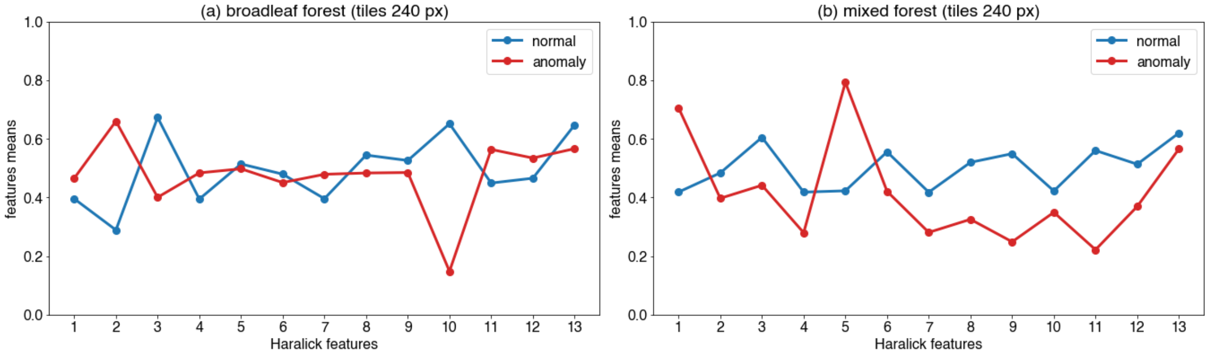


Figure 4.4: Haralick features: means for a flight of broadleaf forest (a) and a flight of mixed forest (b). Some features are similar for both normal (blue) and anomaly (red) tiles. The subset of most distinctive features may change for different flights.

Table 4.5 shows that OC-SVM obtains better results when considering only the subset of the three most discriminating features. When adding other features, which are less discriminating, OC-SVM gets confused by the introduced noise. On the other hand, Isolation Forest obtains weak results when classifying the subset of the three most discriminating features. The reason is that with little information about the data points, Isolation Forest is not capable of accurately discriminating anomalies from the normal data. Consequently, it tends to predict most of the samples as normal. However, selecting more features does not linearly improve the performance. Table 4.6 shows that the best performance is obtained when using the top five features.

Tile size	**Dist**	**Num feats**	ν	**Prec**	**Rec**	**F1**
240	1	3	0.1	51.2%	77.2%	**61.6%**
240	1	5	0.1	48.1%	71.3%	57.4%
240	1	13	0.1	39.5%	55.9%	46.3%

Table 4.5: Influence of the number of selected features on results for Haralick with OC-SVM. Best results obtained with a small subset of features.

Tile size	**Dist**	**Num feats**	**Num trees**	**Max samp**	**Contam**	**Prec**	**Rec**	**F1**
192	1	3	100	80%	5%	45.1%	20.2%	27.9%
192	1	5	100	80%	5%	72.1%	61.8%	**66.6%**
192	1	13	100	80%	5%	71.0%	59.4%	64.7%

Table 4.6: Influence of the number of selected features on the results for Haralick with Isolation Forest. Best results obtained by selecting the 5 most distinctive features. With a smaller set, the performance drastically decreases.

The main issue related to the feature selection step is that the subset of the most discriminating features may change based on the flight types. As an example, Figure 4.4 shows that Haralick features computed on broadleaf and mixed forests are considerably different. The subset of most distinctive features for a broadleaf forest is (2,3,10) whereas for a mixed forest is (5,9,11).

The tile size parameter significantly influences the results of Haralick features. Using larger tiles improves the performance because the global statistics are computed on a wider area of the image containing more contextual information.

The results on the validation set for the best configurations of Haralick features with OC-SVM and Isolation Forest are presented in Table 4.7.

Haralick features are ineffective for discriminating background tiles from anomalies be-

Model	Tile size	Dist	Num feats	ν	Num trees	Max samp	Contam	Prec	Rec	F1
OC-SVM	240	1	3	0.3	-	-	-	51.2%	77.2%	61.6%
IFOR	192	1	5	-	100	80%	5%	72.1%	61.8%	66.6%

Table 4.7: Best configurations of Haralick features with OC-SVM and Isolation Forest evaluated on the validation set (results of anomaly class).

cause the global information extracted is too generic to identify a small target in a large tile. This is confirmed by the best configuration of OC-SVM, which requires a very high value of $\nu = 0.3$ to obtain the best performance, as shown in Table 4.7. The motivation is that the high variability of normal features in the training set forces the definition of a loose boundary that includes many anomaly samples. Isolation Forest cannot detect a large percentage of anomalies because they have features similar to normal samples seen in the training set. The model confusion is confirmed by the low recall of 61.8% on the anomaly class.

SIFT BoW

The only parameters that can be tuned with the SIFT BoW technique are the maximum number of keypoints extracted from a tile and the number of words in the vocabulary. For the maximum number of keypoints, a range of values between 20 and 300 keypoints per tile have been tested. This parameter does not influence the results because, on average, a limited number of keypoints are extracted from each tile, especially on smaller ones. Examples of detected keypoints are shown in Figure 4.5. Results with a variable number of keypoints per tile are shown in Table 4.8.

The vocabulary size has been tested with 10 to 300 words and the best vocabulary size is 25. Using a bigger vocabulary could lead to generating a very sparse Bag-of-Words. In this case, due to the excessive sparsity, the classification models struggle to identify the normality distribution, as shown in Table 4.9. On the other hand, a smaller vocabulary does not generate discriminating features because a wide variety of keypoints with diverse characteristics would be grouped into a small set of clusters with low intra-class similarity. Examples of different vocabulary sizes are shown in Figure 4.6.

The results on the validation set for the best configurations of SIFT with OC-SVM and Isolation Forest are presented in Table 4.10.

Similarly to Haralick features, OC-SVM with SIFT BoW requires a high value of $\nu = 0.3$ to improve the results. This indicates that the classification model cannot easily distinguish normal and anomaly features due to the high variance of normal samples. This behavior is

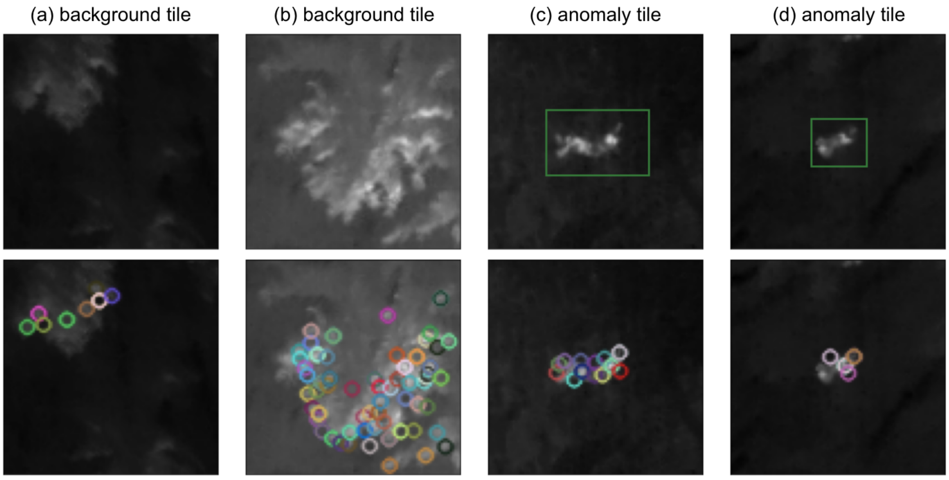


Figure 4.5: Examples of SIFT keypoints detected on background (a,b) and anomaly tiles (c,d). Usually, a few keypoints are extracted from small tiles (a,c,d). In some cases, a larger number of keypoints may be detected when the background is articulate (b).

confirmed by Isolation Forest which has a contamination ratio of 20%. Both models fail to learn an accurate representation of the normality distribution which results in predicting many normal tiles as anomalies. The result is a high recall of anomalies with a very low precision indicating that a high ratio of False Positives is generated. Almost 50% of the anomaly predictions are False Positives, as shown in Table 4.10.

The Bag-of-Words technique is usually adopted for supervised classification tasks where a model learns a vocabulary composed of words representing all the available classes. In an unsupervised setting, there is no word associated with the anomaly class but all the clusters are computed from normal samples. Consequently, the performance is poor because all the keypoints detected on anomaly tiles are associated with the nearest word which, by construction, represents a cluster of normal keypoints.

LETRIST

The implemented version of LETRIST adopts the default parameter values proposed in the original paper [143]. LETRIST introduces a step to transform the extracted features into a scalar texture code to finally produce a histogram representation. The proposed quantization levels were already fine-tuned by the authors and were not further investigated in this work.

Differently from Haralick features, applying the feature selection step to the extracted LETRIST features does not significantly influence the performance, as shown in Table 4.11. This is an indication that even if only a small subset of features are effectively relevant for discriminating between normal and anomaly samples, the other features do

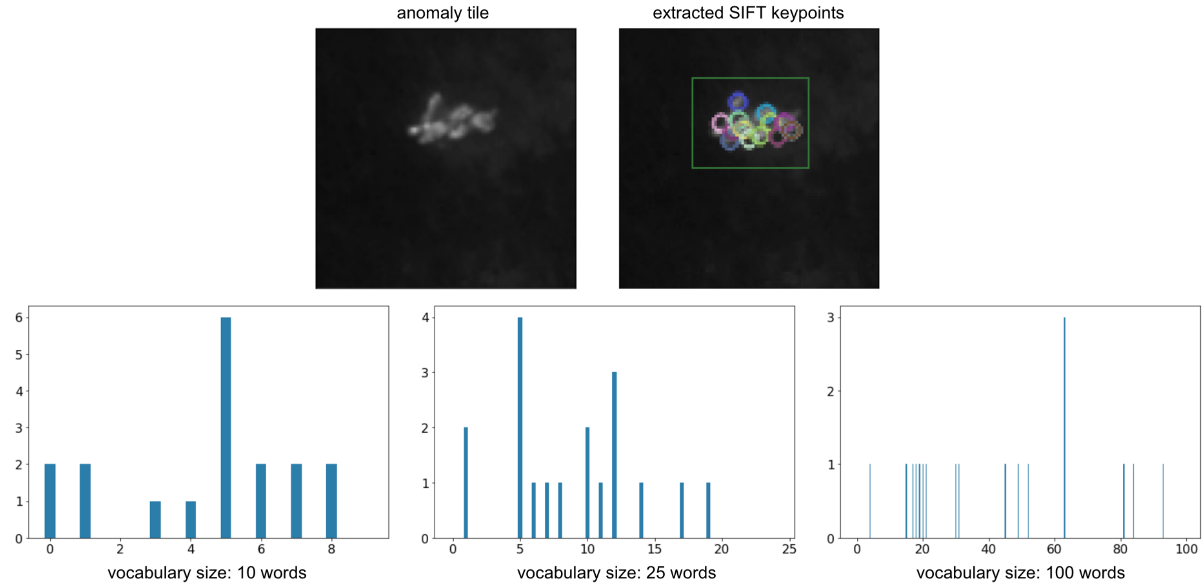


Figure 4.6: Example of anomaly tile with different sizes of SIFT keypoints vocabulary. The best results are obtained with 25 words. Fewer words generate too generic clusters of keypoints while a bigger vocabulary leads to the generation of a sparse Bag of Words.

not introduce much noise.

The results on the validation set for the best configurations of LETRIST with OC-SVM and Isolation Forest are presented in Table 4.12.

LETRIST features are not effective for this anomaly detection task because, similarly to SIFT, the high recall of the anomaly class is associated with a low value of precision. This has the effect of generating a high number of False Positives when using both OC-SVM and Isolation Forest. With OC-SVM, there is a lot of confusion when testing on new data, meaning that features extracted from other images are different from the ones generated from the training set. Consequently, many normal samples are mispredicted as anomalies, as can be seen in Table 4.12 where OC-SVM has only 56.9% of precision on the anomaly class. In contrast, Isolation Forest considers many training samples as outliers since the contamination is 20% and, as a result, most of the samples are predicted as anomalies. However, also in this case, the rate of false alarms is very high, as demonstrated by a low precision of 53.8% on the anomaly class.

HGM

The HGM feature extractor does not have any parameters to be tuned. The original paper [138] defines the histogram descriptor with 16 bins, therefore the same value has been adopted in this work. The key aspect of HGM is that the range of values used to build the histogram is not defined beforehand but is derived from the maximum and

Tile size	Kps tile	Num words	ν	Prec	Rec	F1
96	80	25	0.3	54.3%	90.4%	67.8%
96	100	25	0.3	54.3%	90.5%	**67.9%**
96	200	25	0.3	54.1%	90.5%	67.7%

Table 4.8: Influence of the number of keypoints per tile on the results for SIFT with OC-SVM. This parameter is not influential because, on average, a few keypoints are detected.

Kps tile	Num words	ν	Prec	Rec	F1
100	25	0.3	54.3%	90.5%	**67.9%**
100	50	0.3	25.9%	99.8%	41.2%

Table 4.9: Influence of the vocabulary size on the results for SIFT with OC-SVM. The best performance is obtained with a small vocabulary.

Model	Tile size	Kps tile	Num words	ν	Num trees	Max samp	Contam	Prec	Rec	F1
OC-SVM	96	100	25	0.3	-	-	-	54.3%	90.5%	67.9%
IFOR	96	80	25	-	100	60%	20%	54.1%	90.8%	67.8%

Table 4.10: Best configurations of SIFT with OC-SVM and Isolation Forest evaluated on the validation set (results of anomaly class).

minimum gradient magnitudes computed from each image. This approach for building the histogram always generates a few occurrences of the largest gradient magnitudes because, in a grayscale image, there are statistically fewer pixels with the highest intensity. Consequently, since a small number of pixels has the highest intensity in both normal and anomaly tiles, few occurrences are counted by the histogram bins associated with brighter pixels, as shown in Figure 4.7.

Therefore, the main difference between the features of normal and anomaly tiles resides in the distribution of bins associated with darker pixels. In background tiles without targets, the pixel intensities are typically homogeneous and the computed gradients have a smaller range of magnitudes. Thus, the histograms generated from these tiles are characterized by a homogeneous distribution of magnitude values among a wider set of bins, as shown in Figure 4.7. Differently in anomaly tiles, large gradient magnitudes on the border of the targets generate a larger range of values. Therefore background gradients, which have lower magnitudes with respect to target gradients, are grouped in a reduced set of histogram bins that have a higher count of occurrences, as shown in Figure 4.7.

Tile size	Num feats	Num trees	Max samp	Contam	Prec	Rec	F1
96	25	1,000	60%	20%	53.8%	87.1%	66.5%
96	100	1,000	60%	20%	50.6%	84.2%	63.2%
96	200	1,000	60%	20%	49.1%	84.9%	62.3%
96	413	1,000	60%	20%	53.8%	87.1%	**66.5%**

Table 4.11: Influence of the number of selected features on the results for LETRIST with Isolation Forest. Similar results are obtained independently from the subset of selected features.

Model	Tile size	ν	Num trees	Max samp	Contam	Prec	Rec	F1
OC-SVM	144	0.05	-	-	-	56.9%	71.0%	63.2%
IFOR	96	-	1,000	60%	20%	53.8%	87.1%	66.5%

Table 4.12: Best configurations of LETRIST with OC-SVM and Isolation Forest evaluated on the validation set (results of anomaly class).

The results on the validation set for the best configurations of HGM with OC-SVM and Isolation Forest are presented in Table 4.13.

Model	Tile size	ν	Num trees	Max samp	Contam	Prec	Rec	F1
OC-SVM	96	0.1	-	-	-	74.4%	79.4%	76.8%
IFOR	96	-	400	80%	20%	70.8%	67.0%	68.8%

Table 4.13: Best configurations of HGM with OC-SVM and Isolation Forest evaluated on the validation set (results of anomaly class).

The best OC-SVM configuration has a low value of $\nu = 0.1$, indicating that the normal samples can be accurately enclosed inside the boundary. On the other hand, Isolation Forest uses high contamination of 20%, indicating that many normal features are isolated from the main distribution.

Deep SVDD

Deep SVDD with *soft-boundary* objective has been tested with different values of ν in the range [0.05, 0.3]. The best performance is obtained with $\nu = 0.2$, while with lower or higher values the AUPRC metric decreases, as shown in Table 4.14.

The ν parameter controls the trade-off between the volume of the sphere and the number of points mapped outside the boundary. Similarly to OC-SVM, obtaining high performance with a lower ν means that the model is able to tighten the boundary (a hypersphere,

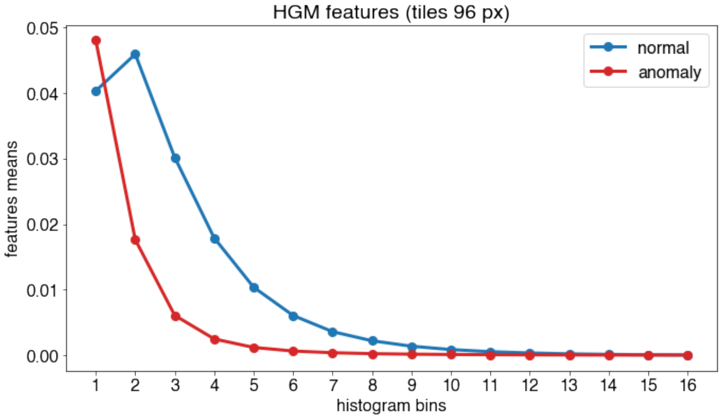


Figure 4.7: HGM features: mean occurrences of gradient magnitudes for anomaly (red) and normal (blue) tiles. The main difference between the two classes resides in the bins that count the smaller values of magnitudes *(left-most bins)*.

Tile size	ν	**AUPRC**
96	0.1	83.47%
96	0.15	83.63%
96	0.2	**83.85%**
96	0.25	83.75%
96	0.3	83.66%

Table 4.14: Influence of the parameter ν of Deep SVDD with *soft-boundary* objective on the AUPRC metric. Best results obtained with a value of $\nu = 0.2$.

in the case of Deep SVDD) around the normal distribution. The optimal configuration identified from the experiments has a relatively high value indicating that many instances of the training set are considered anomalies and thus are mapped outside the boundary. The reason for this behavior is that the encoder cannot learn an accurate representation of normal samples and tends to map many instances in the latent space far from the center of the sphere. By learning a hypersphere that excludes a portion of the normal samples, the recall on the anomaly class is very high, 85.2%, because there is a high probability for an anomaly to be mapped outside of the sphere. Table 4.15 shows that the precision on anomaly class obtained by Deep SVDD is only 56.4%, indicating that many false alarms may be triggered since many normal samples are mispredicted as anomalies.

Deep SVDD with *one-class* objective does not have any parameter to be tuned, such as *soft-boundary*, because it minimizes the distance of every training sample to a central point in the representation space. Therefore, since a hypersphere is not defined such as with *soft-boundary*, an anomaly score based on the distance from the center is computed without effectively predicting a label. The anomaly threshold is defined by choosing a

Objective	Tile size	ν	Prec	Rec	F1
soft-boundary	96	0.2	56.4%	85.2%	67.8%
one-class	96	-	69.2%	80.0%	74.2%

Table 4.15: Best configurations of Deep SVDD evaluated on the validation set (results of anomaly class).

point on the Precision-Recall Curve computed on the validation set.

Given the Precision-Recall Curve for the *one-class* objective in Figure 4.8, it is clear that choosing a recall above 90% (which should the target recall for the use case of this work) is not feasible. Selecting this target recall (red point in Figure 4.8), the corresponding precision would be below 50% obtaining a False Positives Rate that would be excessively high. Consequently, choosing a lower recall but obtaining a higher precision is preferable and it is an effective solution to limit the False Positives Rate.

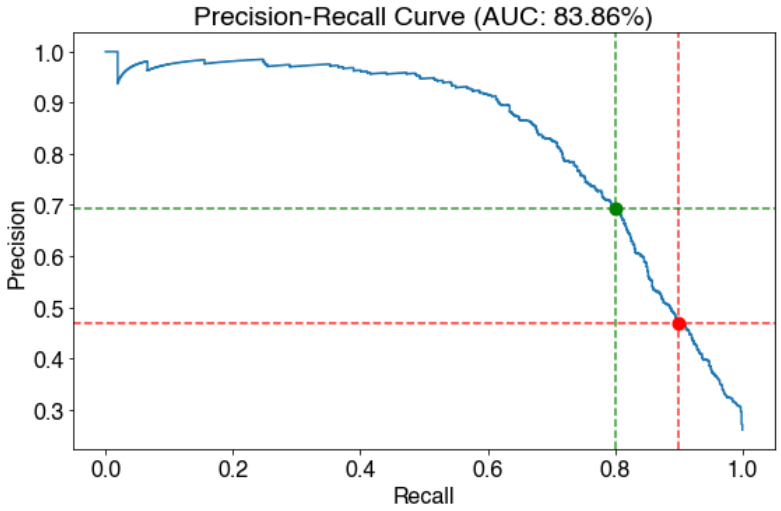


Figure 4.8: Precision-Recall Curve of the best Deep SVDD configuration with *one-class* objective. Selecting 90% recall would lead to very low precision *(red point)*. Reducing the recall to 80% allows obtaining a precision of 70% *(green point)*.

The trade-off is chosen by decreasing the recall value down to 80% (green point in Figure 4.8) and gaining 20% of precision which goes up to 70%. The chosen lower recall corresponds to an increment in the False Negatives Rate. For this reason, this is the minimum acceptable recall value in the use case of Search and Rescue missions because otherwise too many targets would be potentially missed.

The best results obtained with *one-class* objective are shown in Table 4.15.

Deep SAD

Deep SAD has three parameters to be fine-tuned which are the ratio of labeled normal samples ($ratio\_known\_normal$), the ratio of labeled anomaly samples ($ratio\_known\_anomaly$), and the parameter η which defines the weight between the labeled and the unlabeled terms in the loss function.

By increasing the η parameter, more weight is given to the subset of labeled samples composed of both anomaly samples and a number of normal samples defined by *ratio_known_normal*. If η is greater than 1, the set of labeled normal samples has a larger weight than the rest of the unlabeled normal instances. Consequently, Deep SAD tends to define a boundary that better fits the subset of labeled normal instances. A set of experiments with a value of $ratio\_known\_normal$ in the range [0%, 100%] has been performed. In Figure 4.9.a, fixing $\eta = 25$ and training without anomalies, the best results are obtained when all the normal samples have the same weight which are the edge cases of all normal unlabeled ($ratio\_known\_normal = 0\%$) and all normal labeled ($ratio\_known\_normal = 100\%$). By labeling only 25% of the samples, performance decreases because the model is overfitting the subset of labeled samples while the rest of the training set, having a lower weight, has less influence on the loss.

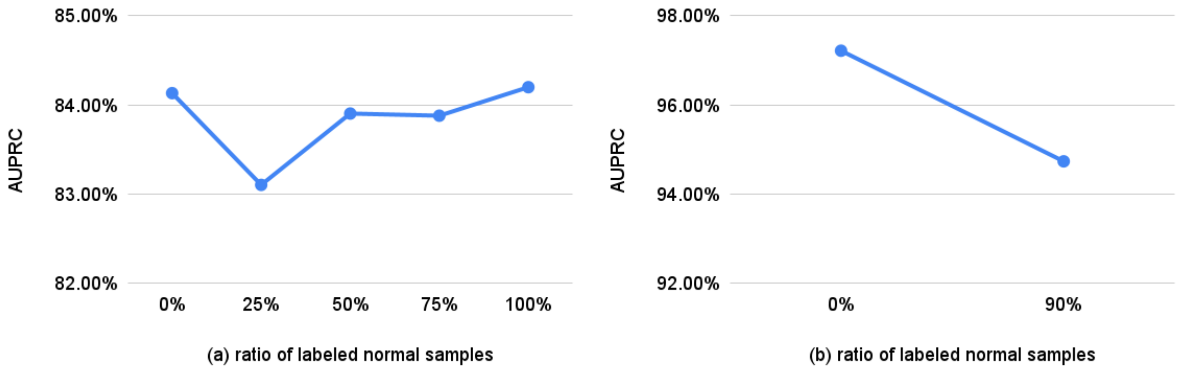


Figure 4.9: Effect of the number of labeled normal samples on Deep SAD training. AUPRC with $\eta = 25$ and 0% anomalies (a): best results obtained with all normal unlabeled ($ratio\_known\_normal = 0\%$) and all normal labeled ($ratio\_known\_normal = 100\%$) because all normal samples have equal weight. AUPRC with $\eta = 25$ and 10% anomalies (b): best results obtained with all normal data unlabeled ($ratio\_known\_normal = 0\%$) because anomaly samples have a higher weight with respect to normal samples.

The difference between having every normal sample labeled or all training set unlabeled is relevant when a set of anomaly instances is introduced in the training set as depicted in

Figure 4.9.b with $\eta = 25$ and 10% of anomalies. When the training set is composed of 90% labeled normal data and 10% labeled anomaly samples ($ratio\_known\_normal = 90\%$, $ratio\_known\_anomaly = 10\%$), the η parameter is not effective because both normal and anomaly data have the same weight on the loss. In this scenario, it is not possible to assign a higher weight to anomaly instances with respect to normal data. On the other hand, when all normal samples are unlabeled ($ratio\_known\_normal = 0\%$), the η parameter can be used to assign higher importance to the labeled anomaly samples suggesting to Deep SAD which areas of the latent space contain anomaly instances and therefore should be excluded from the hypersphere.

By increasing the value of η, anomaly samples have a greater influence on the loss which improves performance because normal samples are densely mapped in the latent space near the center $\mathbf{c}$ of the hypersphere. Values of η in the range [0.1, 50] have been tested but better results are obtained with values greater than 1. As seen in Figure 4.10, increasing the value of η improves the results until $\eta = 25$ and, above this value, the gain in performance is marginal.

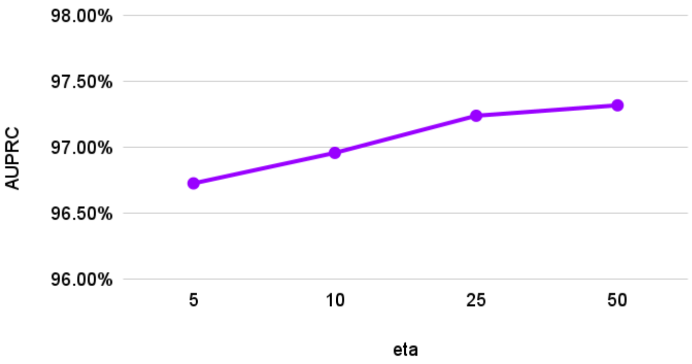


Figure 4.10: Effect of the parameter η on Deep SAD training. AUPRC with all unlabeled normal samples and 5% anomalies: increasing the value of η improves the performance until $\eta = 25$.

The main reason for the strong performance of Deep SAD is the ability to exploit some anomaly samples during the training phase without relying on supervised techniques. Indeed this model can learn useful features of anomaly data only from a small number of instances because the main objective is still to enclose all the normal training samples inside a sphere of minimum volume. When Deep SAD is trained without anomalies ($ratio\_known\_anomaly = 0\%$), the performance is similar to Deep SVDD. This result is in line with the expectations since the models share the same architecture and, when only normal samples are used for training, Deep SVDD and Deep SAD have similar loss functions. During the training of Deep SAD, anomalies are used to better fit the sphere around normal data forcing it to exclude the available anomaly examples. Values of *ratio_known_anomaly* in the range [1%, 20%] have been tested. As can be seen from Figure 4.11, introducing a small number of anomaly instances in the training set, strongly improves the anomaly detection capability with respect to unsupervised techniques. Adding more than 5% of anomaly data in the training set does not significantly increase the performance.

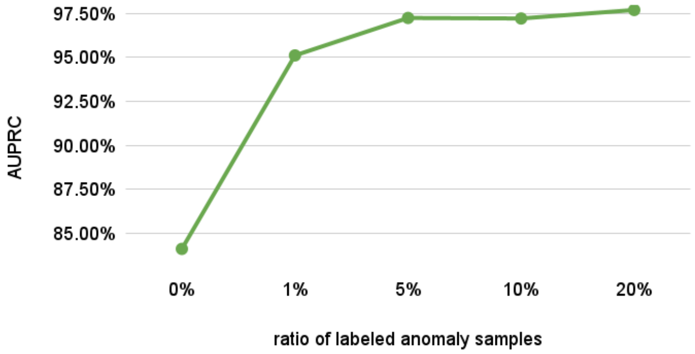


Figure 4.11: Effect of the number of anomaly samples on Deep SAD training. AUPRC with $\eta = 25$ and all unlabeled normal samples: introducing anomaly instances in the training set significantly improves the performance. Adding more than 5% of anomaly samples does have relevant benefits.

The configuration which obtains the best performance on the validation set is presented in Table 4.16.

Known normal	**Known anomaly**	**Tile size**	η	**Prec**	**Rec**	**F1**
0%	5%	96	25	93.7%	90.0%	**91.8%**

Table 4.16: Best configuration of Deep SAD evaluated on the validation set (results of anomaly class).

The best configuration is trained only with unlabeled normal data and exploits only 5% of anomaly samples in the training set which correspond to 343 tiles containing targets. Obtaining this amount of anomaly samples is feasible by performing a small number of data collection flights.

However, when lowering the number of anomaly data in the training set from 5% to 1%, the performance on the test set is worse. Especially the recall of anomaly class becomes significantly lower, as shown in Table 4.17. This result indicates that by training on fewer anomaly samples, the model is still capable of learning a good representation of normal data but the anomalies are not enough to accurately learn the separation between normal and anomaly data. The motivation for this result is that the learned hypersphere includes many anomaly areas of the latent space leading to mispredicting many anomalous instances as normal.

Known normal	**Known anomaly**	**Tile size**	η	**Prec**	**Rec**	**F1**
0%	1%	96	25	99.3%	77.9%	87.3%
0%	5%	96	25	95.4%	90.0%	**92.6%**

Table 4.17: Influence of the parameter *ratio_ known_ anomaly* of Deep SAD on the results of the test set (anomaly class). Worse performance is obtained by reducing the number of anomalies in the training set from 5% to 1%.

4.2.2. Methods Comparison

Deep SAD outperforms all the other analyzed methods both in terms of precision and recall metrics, as shown in Figure 4.12, proving that leveraging the information of a few anomalous samples enables learning high-level features that better separate anomalies from the normality distribution. On the other hand, Deep SVDD with both objectives (blue and pink crosses in Figure 4.12) have performance on the test set that is similar to the results obtained by Machine Learning models. The main advantage of this deep model, especially with *one-class* objective (blue cross in Figure 4.12), is the balance between precision and recall values whereas other Machine Learning techniques usually obtain high values for a metric but low value on the other. For example, HGM with Isolation Forest (orange pentagon in Figure 4.12) has higher precision than Deep SVDD but a lower recall, while SIFT with both classifiers (red and brown triangles in Figure 4.12) obtains a very high recall which is however paired with the lowest precision among all the methods.

The performance of Machine Learning methods is mainly dependent on the discriminating power of the extracted textural features. Haralick features (blue and yellow downward triangles in Figure 4.12), the baseline and LETRIST with OC-SVM (respectively the light blue plus and green circle in Figure 4.12) have weak performance on the test set due to low values both in precision and recall on the anomaly class. The poor performance indicates that these models make a lot of confusion between background and anomaly tiles meaning that they would generate many false alarms while missing a high percentage of targets.

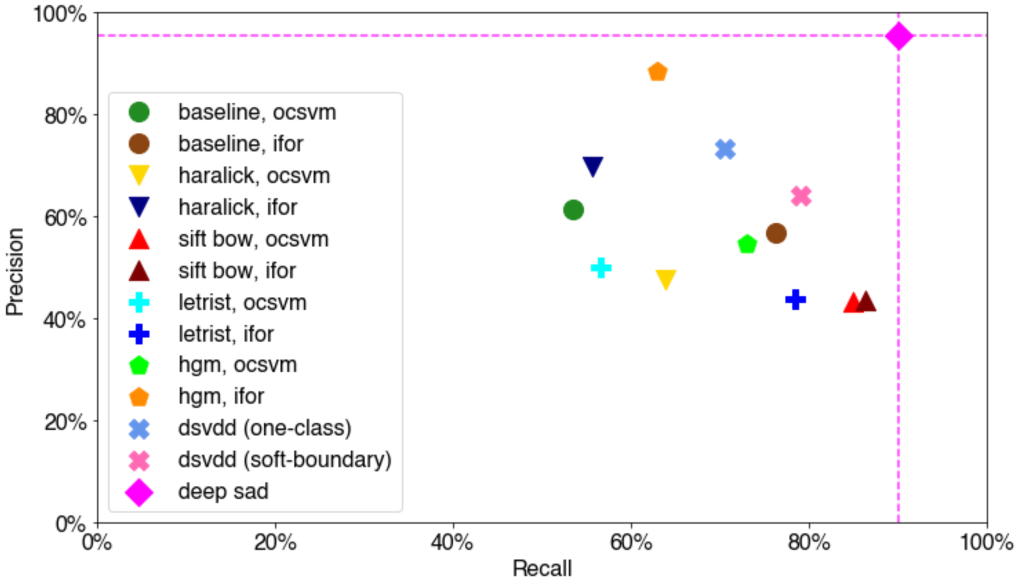


Figure 4.12: Precision and Recall values on the anomaly class of the test set for the best configuration of all the models. Deep SAD (purple diamond) outperforms the other methods both in precision and recall.

Table 4.18 shows the performance obtained by the various combinations of textural feature extractors and Machine Learning classifiers along with the Deep Learning models. The Precision, Recall, and F1-Score are reported separately for the anomaly and normal (background) classes. For each method, also the tile size of the best configuration is reported to give an indication of the level of granularity that can be obtained when generating anomaly heatmaps. Deep Learning models, despite relying on a simple LeNet-type architecture, have better performance than traditional Machine Learning methods accordingly to the F1-Score on anomaly class, as shown in Table 4.18. Only HGM with Isolation Forest obtains a higher F1-Score (73.5%) than Deep SVDD but it is limited by a lower recall (62.9%) indicating that many targets are missed.

Methods such as SIFT with both classifier and LETRIST with Isolation Forest, have a high value of recall (85-86% for SIFT and 78% for LETRIST) but a low precision below 50%. These models are not able to correctly discriminate between background and anomaly tiles and, despite being able to identify many targets, they would generate a high number of False Positives. Due to the high number of False Positives that are generated, these models should be avoided for the use case of rescue missions because the extremely high number of false alarms would not help rescuers focus their attention only on the most important areas of the images.

Machine Learning models and Deep SVDD have worse performance on the test set with

Model	Tile size	Test set, anomaly			Test set, background		
		Precision	Recall	F1	Precision	Recall	F1
Baseline, OC-SVM	240	61.3%	53.5%	57.2%	80.2%	84.8%	82.4%
Baseline, IFOR	216	56.8%	76.3%	65.1%	87.3%	73.8%	80.0%
Haralick, OC-SVM	240	47.6%	63.8%	54.5%	80.7%	68.3%	74.0%
Haralick, IFOR	192	69.8%	55.6%	61.9%	81.6%	89.1%	85.2%
SIFT BoW, OC-SVM	96	43.3%	85.0%	57.4%	88.0%	49.7%	63.6%
SIFT BoW, IFOR	96	43.5%	86.4%	57.8%	88.9%	49.2%	63.4%
LETRIST, OC-SVM	144	49.9%	56.5%	53.0%	92.4%	90.3%	91.3%
LETRIST, IFOR	96	43.7%	78.4%	56.1%	84.8%	54.3%	66.2%
HGM, OC-SVM	96	54.6%	73.0%	62.5%	85.6%	72.6%	78.6%
HGM, IFOR	96	88.4%	62.9%	73.5%	85.2%	96.3%	90.4%
DSVDD one-class	96	73.2%	70.6%	71.9%	87.0%	88.3%	87.6%
DSVDD soft-bound	96	64.2%	79.1%	70.9%	89.5%	80.1%	84.5%
Deep SAD	96	**95.4%**	**90.0%**	**92.6%**	**95.6%**	**98.1%**	**96.8%**

Table 4.18: Evaluation results on the test set for all the models. Metrics are computed separately for the background and anomaly classes. The tile size of the best configuration is indicated for each model.

respect to the results obtained on the validation set. The only exception is HGM with Isolation Forest which gains 5% of F1-Score when evaluated on the test set. The reason, valid for both different behaviors, resides in the different distributions of forest types in the validation and test sets, as explained in Section 4.2. Machine Learning models are not powerful enough to extract features that can be generalized on different forest types characterized by diverse visual appearances. Instead, Deep SVDD despite being capable of learning a better representation of normal data, may suffer from the existing knowledge of the pre-trained feature encoder. On the other hand, Deep SAD, which shares the same architecture as Deep SVDD, obtains similar performance when evaluated on validation and test sets. This proves the capability of this model to learn high-level features that can be generalized on other forests with different textural characteristics.

Finally, Deep SAD demonstrates that the semi-supervised setting allows for significant improvements in anomaly detection performance despite the model being trained with few examples of targets. When fewer anomaly samples are available for training, the results are slightly worse because the anomaly instances are insufficient for the model to learn a good separation between background and anomaly tiles.

A detailed analysis of the anomaly detection performance based on the number of targets visible in a tile has been performed using the model evaluation framework called *ODIN* [152]. For each tile, a meta-annotation is computed to indicate the number of targets that are (at least partially) visible. Then, *ODIN* allows disaggregating the evaluation metrics individually for each value of the meta-annotation. The following analysis reports the recall of anomaly class at the variation of the number of visible targets.

Regarding the amount of context captured by a tile, except for the baseline method and Haralick features, all the other Machine Learning methods obtain better results when processing smaller tiles. SIFT, LETRIST, and HGM are local feature extractors and, therefore, these methods can extract a representation that better captures the target features when the surrounding context is reduced. The reason is that with smaller tiles, the extracted local features are not dominated by large background areas but the targets have a higher influence.

Conversely, the baseline method and Haralick features compute global statistics over the entire input data. These methods have lower recall when few targets are visible in a large tile because background pixels dominate the distribution of intensities, as seen in Figure 4.13.

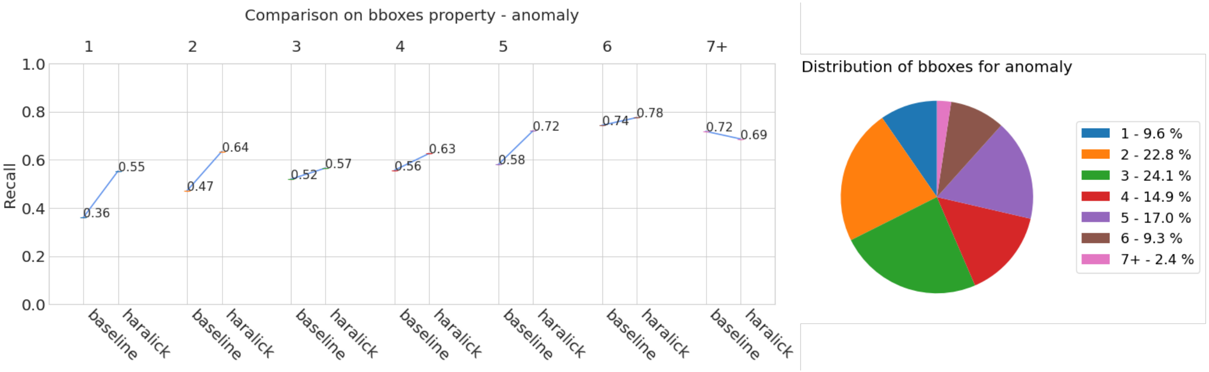


Figure 4.13: Recall on anomaly class for baseline method and Haralick features in relation to the number of targets *(bboxes)* in a tile of 240 px. On the right, the distribution of targets per tile in the test set.

Using bigger tiles, there is a higher probability that many targets are included in the same tile, therefore the influence of target pixels in the global statistics becomes more relevant. Figure 4.13 shows that a high recall is obtained when there are 5 or more visible targets which, for tiles of 240 px, represents the 29% of the anomaly tiles in the test set. Conversely, less than 6% of tiles with a size of 96 px capture more than two targets, as shown in Figure 4.14. The improved performance when many targets are visible motivates the best results obtained by these methods on larger tiles.

For Deep SAD, Deep SVDD (*soft-boundary* objective) and HGM with Isolation Forest, the influence on the anomaly detection performance caused by the number of visible targets has been analyzed. Deep SAD and HGM obtain consistent recall independently from the number of targets visible on a tile, as shown in Figure 4.14. HGM can detect only 60-70% of anomaly targets in all the cases, confirming that this model could potentially miss many people, especially when only a person is visible on a tile (60% recall).

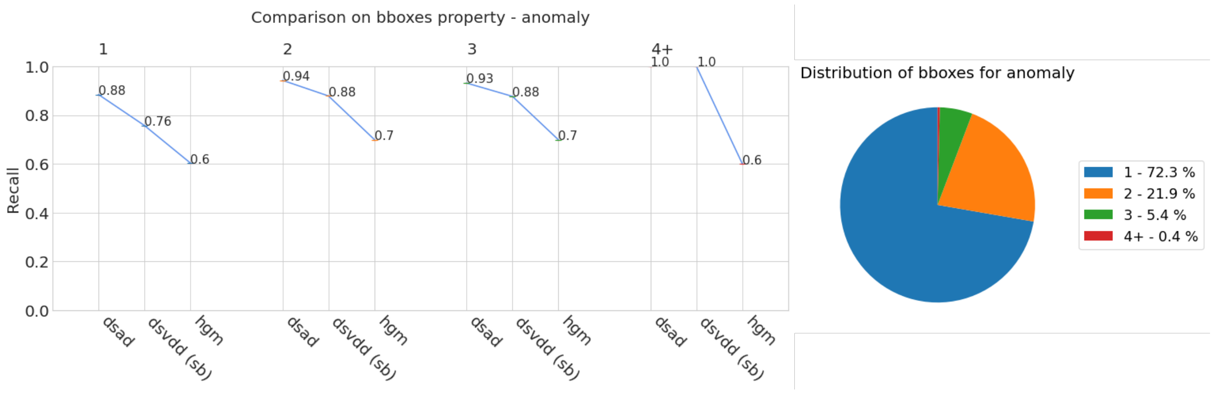


Figure 4.14: Recall on anomaly class for the top three methods (Deep SAD, Deep SVDD *soft-boundary*, HGM) in relation to the number of targets *(bboxes)* in a tile of 96 px. On the right, the distribution of targets per tile in the test set.

Deep SVDD loses 12% recall on tiles with a single target compared to the case when 2-3 targets are visible. Since 72% of the test set is composed of tiles with only one target, this is the main performance bottleneck of Deep SVDD. The reason is that tiles containing two or more targets have a higher likelihood of being different from normal samples and being mapped far from the hypersphere in the latent space. The higher distance from the sphere center allows Deep SVDD to accurately detect the presence of an anomaly. All the tiles containing 4 or more targets are correctly detected both by Deep SVDD and Deep SAD. However, this result is not relevant in the global anomaly detection performance because it represents only 0.4% of the test set. Moreover, during a real rescue mission, it is very unlikely that a large number of nearby people are visible from the same image.

When small tiles of 96x96 px are used, 94.2% of tiles in the test set contain only one or two targets, as shown by the distribution in Figure 4.14. Especially Deep SAD, being trained with anomaly examples, is able to detect 88% of the tiles that contain a single target. Therefore, the major influence on anomaly detection performance depends on the capability of a model to correctly identify the tiles that contain only a single target, since this case represents 72% of the test set.

Obtaining high recall when a single target is visible on a tile is a promising result because, in a real mission, the majority of anomaly tiles would contain only a single person. The motivation is that in most rescue scenarios, only a few people get lost or injured and they might not be all visible from a single image when they are not very close to each other. In these scenarios, Deep SAD which is capable of correctly identifying most of the anomaly tiles that contain a single target reduces the risk of missing some people.

4.3. Qualitative Evaluation

After the quantitative evaluation, a visual inspection of the performance has been conducted. Firstly, some examples of tile predictions are analyzed to discuss the possible sources of confusion for the various methods. Then, the anomaly heatmaps on a set of images with targets and a set of background images are generated to further inspect the anomaly detection performance of the tested models.

Only the top three methods, ranked by the F1-Score on the test set, are selected for the prediction analysis. These methods are Deep SAD, Deep SVDD, and HGM with Isolation Forest which has the highest F1-Score among Machine Learning models. For Deep SVDD, the variant with *soft-boundary* objective is chosen because it has a 9% higher recall with respect to *one-class* objective despite having the same value of F1-Score.

Firstly, some examples of False Negatives, False Positives, and True Positives predicted by all three chosen models are presented. Then, a set of targets identified only by Deep SAD, and missed by the other two models, are shown to prove the advantages that a semi-supervised setting provides over unsupervised techniques.

Figure 4.15 shows examples of False Negatives, which are targets missed by all the top three models. The identification of some targets, such as examples in Figure 4.15 (B,D,E,G), is challenging even for a human operator that might confuse them as tree foliage. Other examples, such as in Figure 4.15 (A,C,F,H,I), are clearly visible to the human eye but still also the best model (Deep SAD) predicts them as background. Especially the example in Figure 4.15 (J) contains two large non-occluded targets but all the models confuse it as background. The reason for these mispredictions could be that the models confuse these targets with similar shapes of tree branches seen during training. To overcome this issue, a more discriminative representation should be learned to better differentiate these anomaly instances from the distribution of background tiles.

Figure 4.16 shows examples of False Positives, which are background tiles confused as anomalies. Some examples, especially in Figure 4.16 (A,C,D,E), contain textures that may resemble the shapes of a human body partially occluded. However, these brighter areas depict tree branches warmed by sunlight that models confuse as targets. An interesting example of model confusion is Figure 4.16 (B) which is composed of dark homogeneous

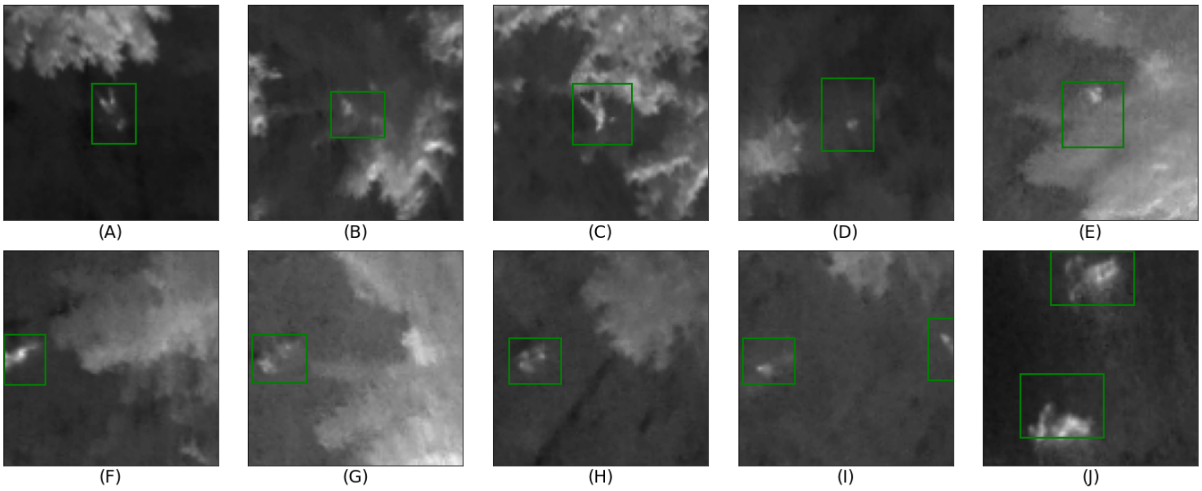


Figure 4.15: Examples of False Negatives by all the top three methods. These anomaly tiles are wrongly classified as background. During a real mission, these targets would be missed. The green boxes enclose the targets.

background but still is classified as an anomaly. The motivation for this error is in the nature of the anomaly detection task. The analyzed models do not learn how to recognize specific shapes or textures associated with the target class but, instead, learn how to model the background data and detect any sample that does not belong to the normality distribution. The empty tile in Figure 4.16 (B) may be characterized by a set of features that are sufficiently distant from the normal distribution and thus the models mispredict it as being an anomaly.

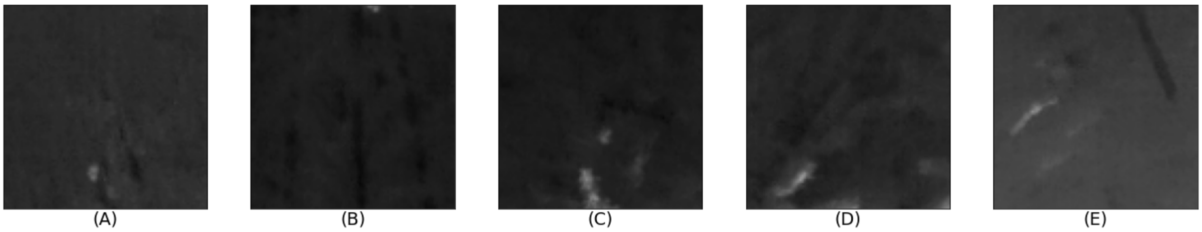


Figure 4.16: Examples of False Positives by all the top three methods. These background tiles are wrongly classified as anomalies. During a real mission, these mispredictions would trigger false alarms that should be manually filtered by rescuers.

Figure 4.17 shows examples of True Positives. Some cases are easily identifiable also by the human eye, such as cases in Figure 4.17 (B,G,H,I). Other examples, such as in Figure 4.17 (E,J), have a higher degree of occlusions that could induce a rescuer to miss the targets or confuse them with the tree foliage. Finally, examples such as in Figure 4.17 (A,C,D,F), demonstrate the power of an anomaly detection model over supervised techniques. These tiles contain targets that are only visible in a few pixels but still, all the models are able to correctly detect the presence of an entity that does not belong to the background. In these cases, supervised models trained to recognize specific features and shapes of a person could miss these instances because they do not resemble the usual aspect of samples learned during training.

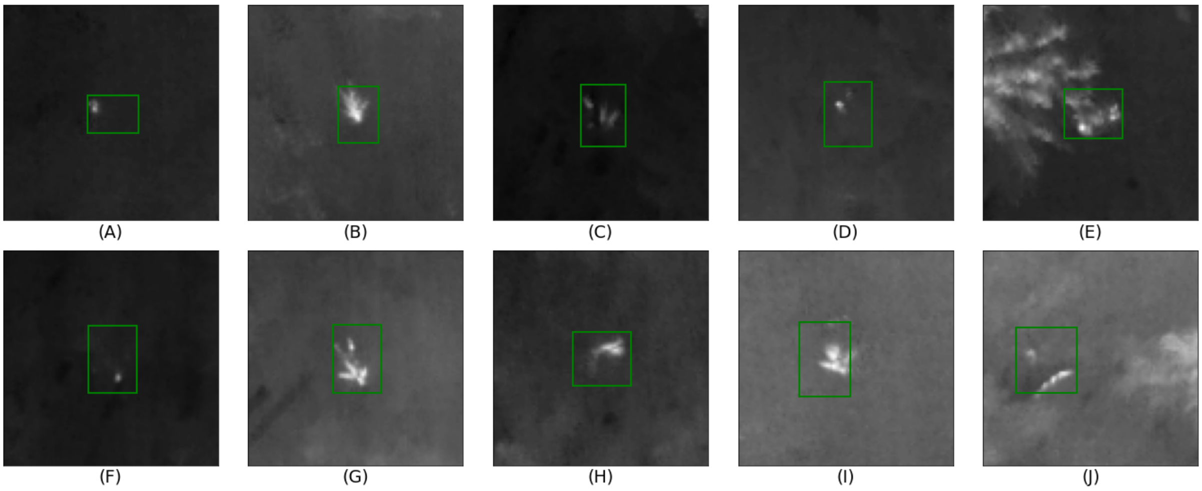


Figure 4.17: Examples of True Positives by all the top three methods. The targets are correctly identified by all the methods despite some instances being visible only for a small fraction (A,C,D,F). The green boxes enclose the targets.

Finally, Figure 4.18 shows some anomaly tiles correctly detected by Deep SAD but missed by the other two models, Deep SVDD and HGM. In some samples, such as Figure 4.18 (A,C,F,H), targets are visible to the human eye but still, some models cannot identify them. The reason is that the features extracted are not discriminative enough to differentiate those anomalies from background data. Examples in Figure 4.18 (B,D) can be confused with the tree foliage by a rescuer but Deep SAD can identify some characterizing features that indicate the presence of a target. Finally, examples such as Figure 4.18 (E,G,I,J), contain weakly visible targets confused with background by HGM and Deep SVDD. In these cases, Deep SAD can detect the strongly occluded targets thanks to the accurate representation learned from both normal and anomaly tiles during training. These results prove that only with a few examples of anomaly data available during the training process, the model can learn more discriminating features that improve the ability to detect weakly visible targets which are instead confused as background by unsupervised methods.

Finally, the anomaly heatmaps for a set of test images are generated for all the implemented models. Each heatmap is generated by combining the predictions of all the tiles

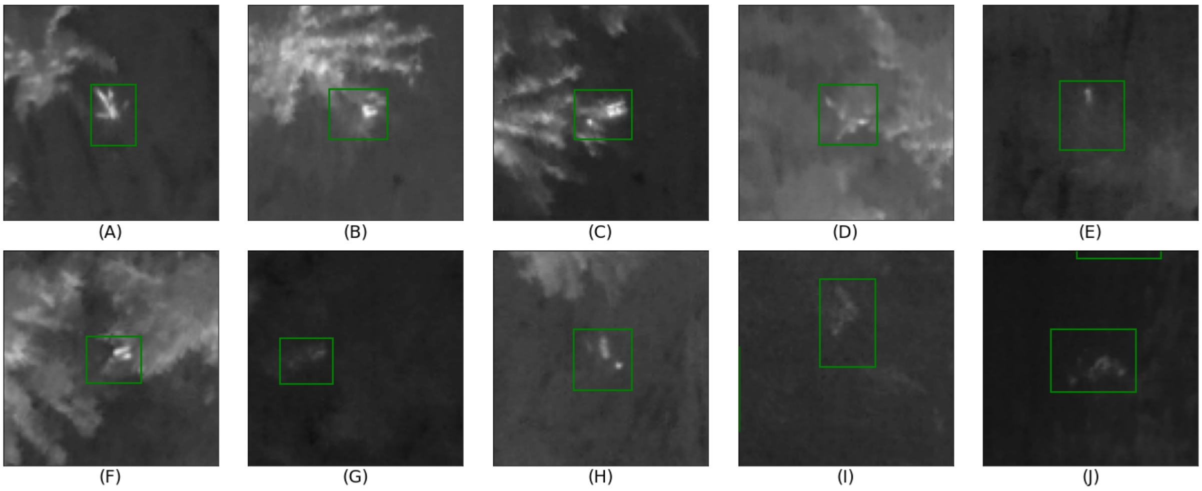


Figure 4.18: Examples of tiles correctly classified as anomalies by Deep SAD but wrongly predicted as background by Deep SVDD and HGM. Deep SAD can identify more difficult targets with respect to the other two methods. The green boxes enclose the targets.

extracted from an image. The tiles are extracted using a sliding window with a size defined by each model. The stride is fixed at 25% of the tile size to obtain a fine-grained heatmap thanks to the overlapping of neighboring tiles. Each tile predicted as an anomaly is assigned a weight equal to +1 (to all its pixels), while tiles predicted as background have a negative weight of -1 (to all the pixels). The weighted sum of the predictions is computed where tiles are overlapping. Using this approach, also tiles predicted as normal are considered to limit the influence of False Positives. For example, if a background tile is misclassified as an anomaly (False Positive), but the nearby tiles are correctly predicted as normal, the False Positive is canceled out in the weighted sum by the correct predictions of the neighboring tiles. After the computation, all the negative weights are discarded and each pixel is characterized by an anomaly score that is higher if many tiles that contain that pixel are classified as an anomaly.

Figure 4.19 shows examples of heatmaps generated on 5 images with targets. The baseline method and Haralick use tiles respectively of 216 px and 192 px that generate coarse heatmaps. The other methods, with tiles of 96 px, enable more accurate localization of the identified targets thanks to the generation of a fine-grained heatmap. Haralick and SIFT often confuse the background areas containing brighter tree foliage warmed by sunlight as being anomalies. In particular, SIFT performs well when the background is homogeneous, but when warmer trees are present, this model tends to incorrectly classify many background areas. HGM, which is the best technique among Machine Learning models, generates anomaly heatmaps that are very similar to the maps generated by

Deep SAD. However in Figure 4.19 (E), Deep SAD is able to identify all the 5 targets, while HGM misses the central one. Deep SVDD has similar results to HGM and Deep SAD but, in general, can confuse a wider range of tiles, as the example in Figure 4.19 (A).

Figure 4.20 shows examples of heatmaps generated on 5 background images. The Haralick method is the worst model on background images because it mispredicts many background areas. This method often confuses the presence of warmer trees with anomalies as shown in Figure 4.20 (H,I). Other models, such as the baseline method, SIFT, LETRIST, and Deep SVDD, correctly predict many tiles but still can get confused by some branches with a visual appearance that resembles target shapes, as the examples in Figure 4.20 (F,G,I) show. HGM and Deep SAD have the best performance even on background tiles since they are rarely confused and correctly predict most of the background tiles. Particularly Deep SAD, in the proposed examples, misclassifies only a few tiles in Figure 4.20 (F), while in all the other background images, it correctly does not detect any anomaly.

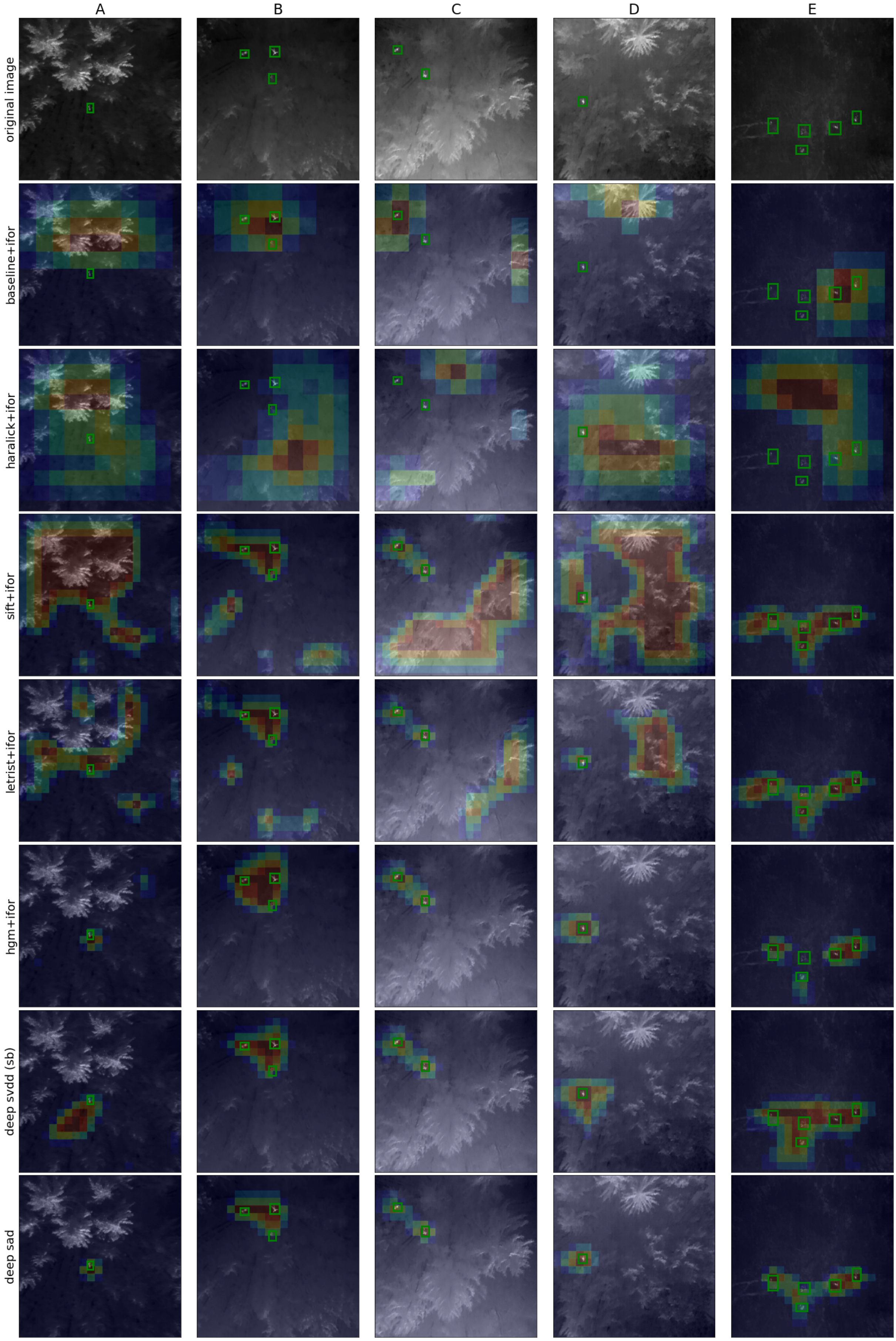


Figure 4.19: Anomaly heatmaps generated on 5 images with targets *(green boxes)*.

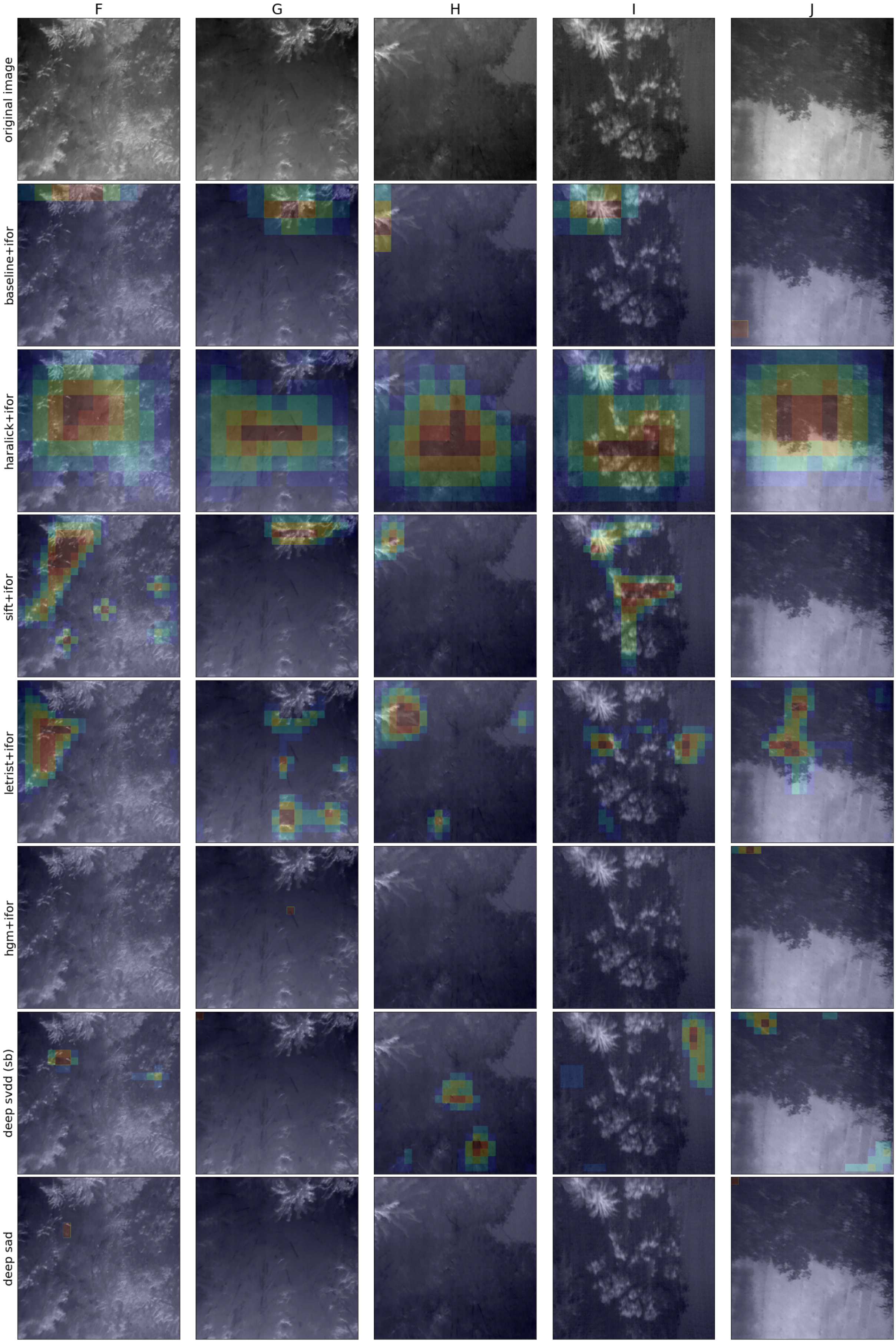


Figure 4.20: Anomaly heatmaps generated on 5 background images.

5 | Conclusions and Future work

This thesis addresses the problem of people detection from drone imagery in a Search and Rescue scenario as an anomaly detection task. This approach implies that models are trained mostly with background images.

First, an extensive review of the state-of-the-art techniques for anomaly detection in images has been conducted to identify the most promising methods that could be applied to the use case under study. Then, an anomaly detection pipeline has been proposed to extract smaller overlapping tiles from an image that are individually classified. Each tile is labeled as an anomaly if the model detects traces of a person that does not belong to the forest background.

The models have been trained and evaluated using a publicly available data set that simulates Search and Rescue missions in forest scenarios [130]. The original data set has been preprocessed, as described in Section 3.1.1, to filter redundant images and clean noise from the ground truth. However, the data set still presents some inherent issues that cannot be addressed and therefore must be taken into consideration when evaluating the anomaly detection models. These issues have been extensively discussed in Section 3.1.3.

The major bias of the data set is the placement of many nearby people on the ground that are captured from the same image. This scenario is not representative of a realistic mission because usually only a few people can be seen in an image during a rescue operation. When many targets are visible in a tile, models have higher chances of identifying at least one person and correctly predicting the tile as an anomaly. During the performance assessment, this aspect has been accurately analyzed to study the effectiveness of the implemented models at the variation of the number of visible people.

Several combinations of feature extractors and Machine Learning classifiers have been tested. For each technique, the influence of the parameters on the results has been accurately studied to determine the configuration that obtains the best performance. Two Deep Learning models for anomaly detection have been tested obtaining a higher performance with respect to shallow Machine Learning models. Especially Deep SAD [122], which is trained in a semi-supervised setting, leverages a limited number of anomaly samples during training to accurately define the separation between background tiles and normal tiles. This strategy allows Deep SAD to outperform all the other implemented techniques both in terms of precision and recall on anomaly class.

This result proves that Deep Learning models are more powerful than Machine Learning techniques and, moreover, the end-to-end training of deep models enables the learning of features that are specifically fine-tuned for the anomaly detection task. Moreover, it demonstrates that the availability of a few anomaly examples during training can significantly improve anomaly detection performance.

A quantitative and qualitative evaluation of the implemented models has been performed to compare the results of all the techniques. The *ODIN* evaluation framework [152] has been exploited to inspect the results based on the number of targets visible in a tile. The predictions of the models have also been visually analyzed to better understand the strengths and weaknesses of the implemented techniques.

In conclusion, Deep SAD is able to achieve 92.6% F1-Score on the anomaly class. This result is characterized by a high recall which indicates that only a few targets might be missed during a real rescue operation. Furthermore, during a real mission, the model is not required to correctly identify the targets from every frame but it should be able to detect the presence of a person in at least one image to alert the rescue team. Therefore, the likelihood of localizing all the injured people in the scene is higher because targets may be visible from at least one frame, thanks to the variability of the ground occlusion that changes from different viewpoints. The results are very promising and prove that approaching the problem with an anomaly detection task is suitable for the detection of people from drone imagery in a Search and Rescue mission.

Despite the high performance obtained with the best model, a lot of directions can be explored to further improve the results. Future work will concentrate on:

- **Realistic Data Set**: the data set used for training and evaluation of the tested models has some intrinsic biases, discussed in Section 3.1.3, that prevent accurate estimation of the model performance during rescue operations. Building a data set composed of images captured during real missions could improve the results by allowing proper training of the models on realistic data.

- **RGB Images**: this thesis has been focused on detecting people from thermal images since the RGB frames in the available data set rarely contain visible people. However, color cameras are often used during rescue operations performed in daylight conditions. Collecting a larger amount of target samples captured with a color camera could enable the training of the proposed models on RGB images. Results could improve since color images may contain more information with respect to grayscale thermal images.

- **Deep SAD Encoder**: Deep SAD has been tested using a simple LeNet-style encoder [83] that in the literature has been outperformed by other more powerful models. The current Deep SAD encoder could be replaced with models such as ResNet [53], VGG [142], AlexNet [73] that are characterized by a deeper architecture. However, these more complex networks are defined by a larger number of parameters that could negatively impact the real-time performance required for fast victim localization.

- **Deep Learning Architectures**: the tested deep models, Deep SVDD and Deep SAD, share a similar architecture. Therefore, totally different architectures could be tested to study the results of different approaches. Other deep anomaly models such as AnoGAN [132], GANomaly [4], and FCDD [92] may be implemented that are able to perform anomaly localization directly on the entire image. Consequently, the complexity of generating a set of tiles from each image could be removed.

Bibliography

[1] E. E. A. Abusham and H. K. Bashir. *Face Recognition Using Local Graph Structure (LGS)*. 2011. doi: 10.1007/978-3-642-21605-3_19.

[2] A. L. Adams, T. A. Schmidt, C. D. Newgard, C. S. Federiuk, M. Christie, S. Scorvo, and M. DeFreest. Search is a time-critical event: When search and rescue missions may become futile. *Wilderness & Environmental Medicine*, 18(2):95–101, 2007. ISSN 1080-6032. doi: https://doi.org/10.1580/06-WEME-OR-035R1.1. URL `https://www.sciencedirect.com/science/article/pii/S1080603207702181`.

[3] M. Ahmadi, M. Sabokrou, M. Fathy, R. Berangi, and E. Adeli. Generative adversarial irregularity detection in mammography images. In I. Rekik, E. Adeli, and S. H. Park, editors, *Predictive Intelligence in Medicine*, pages 94–104, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32281-6.

[4] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In C. V. Jawahar, H. Li, G. Mori, and K. Schindler, editors, *Computer Vision – ACCV 2018*, pages 622–637, Cham, 2019. Springer International Publishing.

[5] S. Albelwi. Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging. *Entropy*, 24(4):551, Apr 2022. ISSN 1099-4300. doi: 10.3390/e24040551. URL `http://dx.doi.org/10.3390/e24040551`.

[6] A. Aldayri and W. Albattah. Taxonomy of anomaly detection techniques in crowd scenes. *Sensors*, 22(16), 2022. ISSN 1424-8220. doi: 10.3390/s22166080. URL `https://www.mdpi.com/1424-8220/22/16/6080`.

[7] R. J. Amala Arokia Nathan, I. Kurmi, D. C. Schedl, and O. Bimber. Through-foliage tracking with airborne optical sectioning. *Journal of Remote Sensing*, 2022, 2022.

[8] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, and B. Schiele. Vision based victim detection from unmanned aerial ve-

hicles. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1740–1747, 2010. doi: 10.1109/IROS.2010.5649223.

[9] D. Avola, L. Cinque, G. L. Foresti, N. Martinel, D. Pannone, and C. Piciarelli. A uav video dataset for mosaicking and change detection from low-altitude flights. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50:2139–2149, 6 2020. ISSN 2168-2216. doi: 10.1109/TSMC.2018.2804766.

[10] A. R. Backes, W. N. Gonçalves, A. S. Martinez, and O. M. Bruno. Texture analysis and classification using deterministic tourist walk. *Pattern Recognition*, 43:685–694, 3 2010. ISSN 00313203. doi: 10.1016/j.patcog.2009.07.017.

[11] A. R. Backes, A. S. Martinez, and O. M. Bruno. Texture analysis using graphs generated by deterministic partially self-avoiding walks. *Pattern Recognition*, 44: 1684–1689, 8 2011. ISSN 00313203. doi: 10.1016/j.patcog.2011.01.018.

[12] D. Bank, N. Koenigstein, and R. Giryes. Autoencoders, 2021.

[13] M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani. A convolutional neural network approach for assisting avalanche search and rescue operations with uav imagery. *Remote Sensing*, 9(2), 2017. ISSN 2072-4292. doi: 10.3390/rs9020100. URL `https://www.mdpi.com/2072-4292/9/2/100`.

[14] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

[15] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1): 303–336, 2014. doi: 10.1109/SURV.2013.052213.00046.

[16] M. Bonetto, P. Korshunov, G. Ramponi, and T. Ebrahimi. Privacy in mini-drone based video surveillance. pages 1–6. IEEE, 5 2015. ISBN 978-1-4799-6026-2. doi: 10.1109/FG.2015.7285023.

[17] I. Bozcan and E. Kayacan. Uav-adnet: Unsupervised anomaly detection using deep neural networks for aerial surveillance. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1158–1164, 2020. doi: 10.1109/IROS45743.2020.9341790.

[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof. pages 93–104. ACM, 5 2000. ISBN 1581132174. doi: 10.1145/342009.335388.

[19] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3), jun 2011. ISSN 0004-5411. doi: 10.1145/1970392.1970395. URL `https://doi.org/10.1145/1970392.1970395`.

[20] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. 1 2019.

[21] R. Chalapathy, A. K. Menon, and S. Chawla. Robust, deep and inductive anomaly detection. In M. Ceci, J. Hollmén, L. Todorovski, C. Vens, and S. Džeroski, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 36–51, Cham, 2017. Springer International Publishing. ISBN 978-3-319-71249-9.

[22] R. Chalapathy, A. K. Menon, and S. Chawla. Anomaly detection using one-class neural networks, 2019.

[23] T. Chalumeau, L. D. F. Costa, O. Laligant, and F. Meriaudeau. Complex networks : application for texture characterization and classification. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 7:93, 4 2009. ISSN 1577-5097. doi: 10.5565/rev/elcvia.247.

[24] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL `https://doi.org/10.1145/1541880.1541882`.

[25] W. Chen, F. Kong, F. Mei, G. Yuan, and B. Li. A novel unsupervised anomaly detection approach for intrusion detection system. In *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, pages 69–73, 2017. doi: 10.1109/BigDataSecurity.2017.56.

[26] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun. Uav-based surveillance system: an anomaly detection approach. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2020. doi: 10.1109/ISCC50000.2020.9219585.

[27] CNSAS. Il soccorso alpino diffonde i dati dell'attività 2018: è record di interventi, aumentano gli incidenti in montagna, 2019. URL `https://news.cnsas.it/il-soccorso-alpino-diffonde-i-dati-dellattivita-2018-e-record-di-interventi-aumentano-gli-incidenti-in-montagna/`. Online; accessed 08 April 2023.

[28] CNSAS. I dati 2022 delle attività del soccorso alpino e speleologico, 2023. URL `https://www.cnsas.it/2023/03/16/dati-2022/`. Online; accessed 08 April 2023.

[29] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967. ISSN 1557-9654. doi: 10.1109/TIT.1967.1053964.

[30] M. Crosier and L. D. Griffin. Using basic image features for texture classification. *International Journal of Computer Vision*, 88:447–460, 7 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0315-0.

[31] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. pages 886–893. IEEE, 2005. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.177.

[32] T. Defard, A. Setkov, A. Loesch, and R. Audigier. Padim: A patch distribution modeling framework for anomaly detection and localization. In A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, H. J. Escalante, and R. Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475–489, Cham, 2021. Springer International Publishing. ISBN 978-3-030-68799-1.

[33] J. del Cerro, C. Cruz Ulloa, A. Barrientos, and J. de León Rivas. Unmanned aerial vehicles in agriculture: A survey. *Agronomy*, 11(2):203, Jan 2021. ISSN 2073-4395. doi: 10.3390/agronomy11020203. URL `http://dx.doi.org/10.3390/agronomy11020203`.

[34] Q. N. Do, M. A. Lewis, A. J. Madhuranthakam, Y. Xi, A. A. Bailey, R. E. Lenkinski, and D. M. Twickler. Texture analysis of magnetic resonance images of the human placenta throughout gestation: A feasibility study. *PLOS ONE*, 14(1):1–11, 01 2019. doi: 10.1371/journal.pone.0211060. URL `https://doi.org/10.1371/journal.pone.0211060`.

[35] P. Doherty and P. Rudol. A uav search and rescue scenario with human body detection and geolocalization. In *Australasian Joint Conference on Artificial Intelligence*, pages 1–13. Springer, 2007.

[36] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning, 2017.

[37] T. Ehret, A. Davy, J.-M. Morel, and M. Delbracio. Image anomalies: A review and synthesis of detection methods. *Journal of Mathematical Imaging and Vision*, 61 (5):710–743, Jun 2019. ISSN 1573-7683. doi: 10.1007/s10851-019-00885-0. URL `https://doi.org/10.1007/s10851-019-00885-0`.

[38] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Deep learning

for medical anomaly detection – a survey. *ACM Comput. Surv.*, 54(7), jul 2021. ISSN 0360-0300. doi: 10.1145/3464423. URL `https://doi.org/10.1145/3464423`.

[39] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *Computer*, 28(9):23–32, 1995. doi: 10.1109/2.410146.

[40] W. T. Freeman, E. H. Adelson, et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[41] A.-J. Gallego, A. Pertusa, P. Gil, and R. B. Fisher. Detection of bodies in maritime rescue operations using unmanned aerial vehicles with multispectral cameras. *Journal of Field Robotics*, 36(4):782–796, 2019. doi: https://doi.org/10.1002/rob.21849. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21849`.

[42] M. M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4:172–179, 6 1975. ISSN 0146664X. doi: 10.1016/S0146-664X(75)80008-6.

[43] H. Gao, L. Dou, W. Chen, and J. Sun. Image classification with bag-of-words model based on improved sift algorithm. In *2013 9th Asian Control Conference (ASCC)*, pages 1–6, 2013. doi: 10.1109/ASCC.2013.6606268.

[44] M. K. Ghalati, A. Nunes, H. Ferreira, P. Serranho, and R. Bernardes. Texture analysis and its applications in biomedical imaging: A survey. *IEEE Reviews in Biomedical Engineering*, 15:222–246, 2022. doi: 10.1109/RBME.2021.3115703.

[45] I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/5e62d03aec0d17facfc5355dd90d441c-Paper.pdf`.

[46] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[47] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[48] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020. ISSN 0001-0782. doi: 10.1145/3422622. URL `https://doi.org/10.1145/3422622`.

[49] M. A. Goodrich, B. S. Morse, C. Engh, J. L. Cooper, and J. A. Adams. Towards using unmanned aerial vehicles (uavs) in wilderness search and rescue: Lessons from field trials. *Interaction Studies*, 10(3):453–478, 2009. ISSN 1572-0373. doi: https://doi.org/10.1075/is.10.3.08goo. URL `https://www.jbe-platform.com/content/journals/10.1075/is.10.3.08goo`.

[50] D. Gudovskiy, S. Ishizaka, and K. Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 98–107, January 2022.

[51] S. Hamdi, S. Bouindour, H. Snoussi, T. Wang, and M. Abid. End-to-end deep one-class learning for anomaly detection in uav video stream. *Journal of Imaging*, 7(5): 90, 2021.

[52] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3:610–621, 11 1973. ISSN 0018-9472. doi: 10.1109/TSMC.1973.4309314.

[53] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[54] D. K. Hoai and N. V. Phuong. Anomaly color detection on uav images for search and rescue works. volume 2017-January, 2017. doi: 10.1109/KSE.2017.8119473.

[55] C. Huang, Z. Yang, J. Wen, Y. Xu, Q. Jiang, J. Yang, and Y. Wang. Self-supervision-augmented deep autoencoder for unsupervised visual anomaly detection. *IEEE Transactions on Cybernetics*, 52(12):13834–13847, 2022. doi: 10.1109/TCYB.2021.3127716.

[56] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[57] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1997. doi: 10.1109/CVPR.1997.609412.

[58] A. Humeau-Heurtier. Texture feature extraction methods: A survey. *Ieee Access*, 7:8975–9000, 2019.

[59] M. R. Islam and A. Matin. Detection of covid 19 from ct image by the novel lenet-5 cnn architecture. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5, 2020. doi: 10.1109/ICCIT51783.2020.9392723.

[60] J. Jabez and B. Muthukumar. Intrusion detection system (ids): Anomaly detection using outlier detection approach. *Procedia Computer Science*, 48:338–346, 2015. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2015.04.191. URL `https://www.sciencedirect.com/science/article/pii/S1877050915007000`. International Conference on Computer, Communication and Convergence (ICCC 2015).

[61] K. Jafari-Khouzani and H. Soltanian-Zadeh. Rotation-invariant multiresolution texture analysis using radon and wavelet transforms. *IEEE Transactions on Image Processing*, 14(6):783–795, 2005. doi: 10.1109/TIP.2005.847302.

[62] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996. ISSN 0031-3203. doi: https://doi.org/10.1016/0031-3203(95)00160-3. URL `https://www.sciencedirect.com/science/article/pii/0031320395001603`.

[63] X. Jiang, G. Xie, J. Wang, Y. Liu, C. Wang, F. Zheng, and Y. Jin. A survey of visual sensory anomaly detection. 2 2022.

[64] J. R. Juita S, H. Hidayati, and A. A. Gozali. Electronic product feature-based sentiment analysis using nu-svm method. *International Journal on Information and Communication Technology (IJoICT)*, 1(1):38–44, Mar. 2016. doi: 10.21108/IJOICT.2015.11.4. URL `http://socj.telkomuniversity.ac.id/ojs/index.php/ijoict/article/view/4`.

[65] Kamat, Pooja and Sugandhi, Rekha. Anomaly detection for predictive maintenance in industry 4.0- a survey. *E3S Web Conf.*, 170:02007, 2020. doi: 10.1051/e3sconf/202017002007. URL `https://doi.org/10.1051/e3sconf/202017002007`.

[66] Y. Karaca, M. Cicek, O. Tatli, A. Sahin, S. Pasli, M. F. Beser, and S. Turedi. The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. *The American Journal of Emergency Medicine*, 36(4):583–588, 2018. ISSN 0735-6757. doi: https://doi.org/10.1016/j.ajem.2017.09.025. URL `https://www.sciencedirect.com/science/article/pii/S0735675717307507`.

[67] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang. Real-time traffic flow parameter estimation from uav video based on ensemble classifier and optical flow. *IEEE*

Transactions on Intelligent Transportation Systems, 20(1):54–64, 2019. doi: 10.1109/TITS.2018.2797697.

[68] S. S. Khan and M. G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014. doi: 10.1017/S026988891300043X.

[69] J. Kim, S. Kim, C. Ju, and H. I. Son. Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications. *IEEE Access*, 7: 105100–105115, 2019. doi: 10.1109/ACCESS.2019.2932119.

[70] S. Kim, S. Yoo, J. Park, S. Cho, and T. Kim. Rapid disaster mapping through data integration from uavs and multi-sensors mounted on investigation platforms of ndmi, korea. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:291–294, 2018.

[71] R. J. Koester. Lost person behavior: A search and rescue. *dbs Productions LLC*, 2008.

[72] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL https://doi.org/10.1145/3065386.

[74] G. Kumar and P. K. Bhatia. A detailed review of feature extraction in image processing systems. In *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, pages 5–12, 2014. doi: 10.1109/ACCT.2014.74.

[75] I. Kurmi, D. C. Schedl, and O. Bimber. Airborne optical sectioning. *Journal of Imaging*, 4(8):102, 2018.

[76] I. Kurmi, D. C. Schedl, and O. Bimber. A statistical view on synthetic aperture imaging for occlusion removal. *IEEE Sensors Journal*, 19:9374–9383, 10 2019. ISSN 1530-437X. doi: 10.1109/JSEN.2019.2922731.

[77] I. Kurmi, D. C. Schedl, and O. Bimber. Thermal airborne optical sectioning. *Remote Sensing*, 11(14):1668, 2019.

[78] I. Kurmi, D. C. Schedl, and O. Bimber. Data: Autonomous drones for search and rescue in forests. 12 2020. doi: 10.5281/ZENODO.4349220. URL https://zenodo.org/record/4349220.

[79] I. Kurmi, D. C. Schedl, and O. Bimber. Combined person classification with airborne optical sectioning. *Scientific reports*, 12(1):1–11, 2022.

[80] H. Kwon and N. Nasrabadi. Kernel rx-algorithm: a nonlinear anomaly detector for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43 (2):388–397, 2005. doi: 10.1109/TGRS.2004.841487.

[81] S. Küçük and S. E. Yüksel. Comparison of rx-based anomaly detectors on synthetic and real hyperspectral data. In *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4, 2015. doi: 10.1109/WHISPERS.2015.8075504.

[82] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[83] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[84] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL `https://doi.org/10.1038/nature14539`.

[85] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. 4 2021.

[86] L. Li, C. S. Tong, and S. K. Choy. Texture classification using refined histogram. *IEEE Transactions on Image Processing*, 19:1371–1378, 5 2010. ISSN 1057-7149. doi: 10.1109/TIP.2010.2041414.

[87] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:18–32, 1 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.111.

[88] W. Li, X. Li, Y. Qin, W. Song, and W. Cui. Application of improved lenet-5 network in traffic sign recognition. In *Proceedings of the 3rd International Conference on Video and Image Processing*, ICVIP '19, page 13–18, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450376822. doi: 10.1145/3376067.3376102. URL `https://doi.org/10.1145/3376067.3376102`.

[89] F. Liu and R. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:722–733, 7 1996. ISSN 01628828. doi: 10.1109/34.506794.

[90] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. pages 413–422. IEEE, 12 2008. ISBN 978-0-7695-3502-9. doi: 10.1109/ICDM.2008.17.

[91] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2023. doi: 10.1109/TKDE.2021.3090866.

[92] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K.-R. Müller. Explainable deep one-class classification, 2021.

[93] L. Lopez-Fuentes, J. van de Weijer, M. González-Hidalgo, H. Skinnemoen, and A. D. Bagdanov. Review on computer vision techniques in emergency situations. *Multimedia Tools and Applications*, 77(13):17069–17107, Jul 2018. ISSN 1573-7721. doi: 10.1007/s11042-017-5276-7. URL `https://doi.org/10.1007/s11042-017-5276-7`.

[94] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 11 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94.

[95] R. Maani, S. Kalra, and Y.-H. Yang. Noise robust rotation invariant features for texture classification. *Pattern Recognition*, 46:2103–2116, 8 2013. ISSN 00313203. doi: 10.1016/j.patcog.2013.01.014.

[96] N. Malini and M. Pushpa. Analysis on credit card fraud identification techniques based on knn and outlier detection. In *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pages 255–258, 2017. doi: 10.1109/AEEICB.2017.7972424.

[97] B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:837–842, 1996. ISSN 01628828. doi: 10.1109/34.531803.

[98] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In T. Honkela, W. Duch, M. Girolami, and S. Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 52–59, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21735-7.

[99] R. Mehta and K. Egiazarian. Dominant rotated local binary patterns (drlbp) for texture classification. *Pattern Recognition Letters*, 71:16–22, 2 2016. ISSN 01678655. doi: 10.1016/j.patrec.2015.11.019.

[100] B. Mohammadi, M. Fathy, and M. Sabokrou. Image/video deep anomaly detection: A survey. 2021. doi: 10.48550/ARXIV.2103.01739. URL `https://arxiv.org/abs/2103.01739`.

[101] P. Molina, I. Colomina, P. Victoria, J. Skaloud, W. Kornus, R. Prades, and C. Aguilera. Drones to the rescue! *Inside GNSS*, July/August, 2012. URL `http://infoscience.epfl.ch/record/180464`.

[102] N. H. Motlagh, M. Bagaa, and T. Taleb. Uav-based iot platform: A crowd surveillance use case. *IEEE Communications Magazine*, 55(2):128–134, 2017. doi: 10.1109/MCOM.2017.1600587CM.

[103] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab. Machine learning for anomaly detection: A systematic review. *IEEE Access*, 9:78658–78700, 2021. doi: 10.1109/ACCESS.2021.3083060.

[104] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29:51–59, 1 1996. ISSN 00313203. doi: 10.1016/0031-3203(95)00067-4.

[105] L. P. Osco, J. Marcato Junior, A. P. Marques Ramos, L. A. de Castro Jorge, S. N. Fatholahi, J. de Andrade Silva, E. T. Matsubara, H. Pistori, W. N. Gonçalves, and J. Li. A review on deep learning in uav remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, 102:102456, 2021. ISSN 1569-8432. doi: https://doi.org/10.1016/j.jag.2021.102456. URL `https://www.sciencedirect.com/science/article/pii/S030324342100163X`.

[106] P. Oza and V. M. Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2019. doi: 10.1109/LSP.2018.2889273.

[107] G. Pajares. Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs). *Photogrammetric Engineering & Remote Sensing*, 81(4):281–329, 2015. ISSN 0099-1112. doi: https://doi.org/10.14358/PERS.81.4.281. URL `https://www.sciencedirect.com/science/article/pii/S0099111215300793`.

[108] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel. Deep learning for anomaly detection: A review. 54(2), mar 2021. ISSN 0360-0300. doi: 10.1145/3439950. URL `https://doi.org/10.1145/3439950`.

[109] E. Parzen. On estimation of a probability density function and mode. *The Annals*

of Mathematical Statistics, 33:1065–1076, 9 1962. ISSN 0003-4851. doi: 10.1214/aoms/1177704472.

[110] G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. In *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*, pages 96–102, 1996. doi: 10.1109/ACV.1996.572008.

[111] P. Perera, R. Nallapati, and B. Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[112] D. Ping Tian et al. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4):385–396, 2013.

[113] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[114] M. Plauth, W. Hagen, F. Feinbube, F. Eberhardt, L. Feinbube, and A. Polze. Parallel implementation strategies for hierarchical non-uniform memory access systems by example of the scale-invariant feature transform algorithm. In *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1351–1359, 2016. doi: 10.1109/IPDPSW.2016.47.

[115] U. Porwal and S. Mukund. Credit card fraud detection in e-commerce: An outlier detection approach, 2019.

[116] J. Proft, J. Suarez, and R. Murphy. Spectral anomaly detection with machine learning for wilderness search and rescue. In *2015 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pages 1–3, 2015. doi: 10.1109/URTC.2015.7563746.

[117] M. Półka, S. Ptak, and Łukasz Kuziora. The use of uav's for search and rescue operations. *Procedia Engineering*, 192:748–752, 2017. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2017.06.129. URL `https://www.sciencedirect.com/science/article/pii/S1877705817326759`. 12th international scientific conference of young scientists on sustainable, modern and safe transport.

[118] I. Reed and X. Yu. Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1760–1770, 1990. doi: 10.1109/29.60107.

[119] M. Rezapour. Anomaly detection using unsupervised methods: credit card fraud case study. *International Journal of Advanced Computer Science and Applications*, 10(11), 2019.

[120] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14318–14328, June 2022.

[121] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. volume 80, pages 4393–4402. PMLR, 3 2018. URL `https://proceedings.mlr.press/v80/ruff18a.html`.

[122] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection, 2020.

[123] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, Dec 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y. URL `https://doi.org/10.1007/s11263-015-0816-y`.

[124] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[125] M. Sabokrou, M. Khalooei, and E. Adeli. Self-supervised representation learning via neighborhood-relational encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[126] S. Sambolek and M. Ivasic-Kos. Automatic person detection in search and rescue operations using deep cnn detectors. *IEEE Access*, 9:37905–37922, 2021. doi: 10.1109/ACCESS.2021.3063681.

[127] S. Sambolek and M. Ivasic-Kos. Search and rescue image dataset for person detection - sard, 2021. URL `https://dx.doi.org/10.21227/ahxm-k331`.

[128] K. K. Santhosh, D. P. Dogra, and P. P. Roy. Anomaly detection in road traffic using visual surveillance: A survey. *ACM Comput. Surv.*, 53(6), dec 2020. ISSN 0360-0300. doi: 10.1145/3417989. URL `https://doi.org/10.1145/3417989`.

[129] D. C. Schedl, I. Kurmi, and O. Bimber. Search and rescue with airborne optical

sectioning. *Nature Machine Intelligence*, 2, 2020. ISSN 25225839. doi: 10.1038/s42256-020-00261-3.

[130] D. C. Schedl, I. Kurmi, and O. Bimber. Data: Search and rescue with airborne optical sectioning. 6 2020. doi: 10.5281/ZENODO.4024677. URL `https://zenodo.org/record/4024677`.

[131] D. C. Schedl, I. Kurmi, and O. Bimber. An autonomous drone for search and rescue in forests using airborne optical sectioning. *Science Robotics*, 6(55), 2021.

[132] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In M. Niethammer, M. Styner, S. Aylward, H. Zhu, I. Oguz, P.-T. Yap, and D. Shen, editors, *Information Processing in Medical Imaging*, pages 146–157, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59050-9.

[133] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. ISBN 978-3-540-69620-9.

[134] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 7 2001. ISSN 0899-7667. doi: 10.1162/089976601750264965.

[135] V. Sehwag, M. Chiang, and P. Mittal. Ssd: A unified framework for self-supervised outlier detection, 2021.

[136] F. Seits, I. Kurmi, and O. Bimber. Evaluation of color anomaly detection in multispectral images for synthetic aperture sensing. *Eng*, 3(4):541–553, 2022. ISSN 2673-4117. doi: 10.3390/eng3040038. URL `https://www.mdpi.com/2673-4117/3/4/38`.

[137] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634, 2019. doi: 10.1109/ACCESS.2019.2909530.

[138] M. Sharma and H. Ghosh. Histogram of gradient magnitudes: A rotation invariant texture-descriptor. pages 4614–4618. IEEE, 9 2015. ISBN 978-1-4799-8339-1. doi: 10.1109/ICIP.2015.7351681.

[139] A. D. Shieh and D. F. Kamm. Ensembles of one class support vector machines. In J. A. Benediktsson, J. Kittler, and F. Roli, editors, *Multiple Classifier Systems*,

pages 181–190, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02326-2.

[140] B. Siegel. Industrial anomaly detection: A comparison of unsupervised neural network architectures. *IEEE Sensors Letters*, 4(8):1–4, 2020. doi: 10.1109/LSENS.2020.3007880.

[141] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge. Multipurpose uav for search and rescue operations in mountain avalanche events. *Geomatics, Natural Hazards and Risk*, 8(1):18–33, 2017. doi: 10.1080/19475705.2016.1238852. URL `https://doi.org/10.1080/19475705.2016.1238852`.

[142] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. URL `https://arxiv.org/abs/1409.1556`.

[143] T. Song, H. Li, F. Meng, Q. Wu, and J. Cai. Letrist: Locally encoded transform feature histogram for rotation-invariant texture classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 28:1565–1579, 7 2018. ISSN 1051-8215. doi: 10.1109/TCSVT.2017.2671899.

[144] J. J. P. Suarez and P. C. Naval. A survey on deep learning techniques for video anomaly detection, 2020. URL `https://arxiv.org/abs/2009.14146`.

[145] W. Sultani and M. Shah. Human action recognition in drone videos using a few aerial training examples. *Computer Vision and Image Understanding*, 206:103186, 2021. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2021.103186. URL `https://www.sciencedirect.com/science/article/pii/S1077314221000308`.

[146] J. Sun, B. Li, Y. Jiang, and C.-y. Wen. A camera-based target detection and positioning uav system for search and rescue (sar) purposes. *Sensors*, 16(11), 2016. ISSN 1424-8220. doi: 10.3390/s16111778. URL `https://www.mdpi.com/1424-8220/16/11/1778`.

[147] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi: 10.1609/aaai.v31i1.11231. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11231`.

[148] D. S. Tan, Y.-C. Chen, T. P.-C. Chen, and W.-C. Chen. Trustmae: A noise-resilient defect classification framework using memory-augmented auto-encoders with trust regions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 276–285, January 2021.

[149] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak. Deep learning for unsupervised anomaly localization in industrial images: A survey. *IEEE Transactions on Instrumentation and Measurement*, 71:1–21, 2022. doi: 10.1109/TIM.2022.3196436.

[150] D. M. Tax and R. P. Duin. Support vector data description. *Machine Learning*, 54: 45–66, 1 2004. ISSN 0885-6125. doi: 10.1023/B:MACH.0000008084.60811.49.

[151] D. M. J. Tax and K.-R. Müller. Feature extraction for one-class classification. In O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, editors, *Artificial Neural Networks and Neural Information Processing — ICANN/ICONIP 2003*, pages 342–349, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-44989-8.

[152] R. N. Torres, F. Milani, and P. Fraternali. *ODIN: Pluggable Meta-annotations and Metrics for the Diagnosis of Classification and Localization*, pages 383–398. 2022. doi: 10.1007/978-3-030-95467-3_28.

[153] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, and N.-M. Cheung. An improved self-supervised gan via adversarial training, 2019.

[154] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis. Attention guided anomaly localization in images. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 485–503, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58520-4.

[155] J. Yang, R. Xu, Z. Qi, and Y. Shi. Visual anomaly detection for images: A survey. 9 2021.

[156] J. Yang, R. Xu, Z. Qi, and Y. Shi. Visual anomaly detection for images: A systematic survey. *Procedia Computer Science*, 199:471–478, 2022. ISSN 18770509. doi: 10.1016/j.procs.2022.01.057.

[157] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows, 2021.

[158] V. Zavrtanik, M. Kristan, and D. Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 4 2021. ISSN 00313203. doi: 10.1016/j.patcog.2020.107706.

[159] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar. Efficient gan-based anomaly detection, 2018. URL `https://arxiv.org/abs/1802.06222`.

[160] J. Zhang, J. Liang, and H. Zhao. Local energy pattern for texture classification using

self-adaptive quantization thresholds. *IEEE Transactions on Image Processing*, 22: 31–42, 1 2013. ISSN 1057-7149. doi: 10.1109/TIP.2012.2214045.

[161] Z.-H. Zhou. *Machine learning*. Springer Nature, 2021.

[162] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge, 2018. URL `https://arxiv.org/abs/1804.07437`.

[163] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021. doi: 10.1109/JPROC.2020.3004555.

List of Figures

List of Tables