Automatic detection of fiber-cement roofs in aerial images

Kevin Martín Fernández

Abstract

Asbestos is a fiber cement harmful to health. For this reason, almost all countries have laws to eliminate this material. Unfortunately, asbestos detection is a challenging task. The current procedures for identifying asbestos require human exploration, which is costly and slow. This has motivated the interest of governments and companies to develop automatic tools that can help to detect and classify these types of materials that are dangerous to the population. This paper explores multiple computer vision techniques based on deep learning for the automatic detection of asbestos in aerial images. Concretely, we trained and tested implementations of Mask-RCNN, ResNet and Embedding spaces, and we used data augmentation and weighted sampling to overcome the data scarcity for training. The obtained results are 86.55% average of correct asbestos detection in the testing set, which shows the potential of the computer vision deep learning techniques for the automatic detection of asbestos in aerial images.

Index Terms

Deep Learning, Segmentation, Fiber-cement, Detection, Data augmentation, Health, Roofs, Aerial images, Classification, Embedding spaces, Computer vision, Unbalanced data

I. Introduction

Asbestos is a component made of fiber cement, the material used in many buildings, concretely in rooftops, tiles, cements, fire retardant suits and other applications. Many years ago, it was popular in the industry due to its properties and low cost. It resists heat and flames and was used as an insulating material in the mechanical and chemical industries. Its characteristics and price caused an exponential increment of its installation.

Fiber cement is not harmful in its original conditions. However, deterioration of these materials causes the detachment of crystalline fibers. Fibers can be inhaled by humans and make micro-wounds in the lungs, causing health problems. In multiple articles, these health problems are related to the development of cancer diagnoses [1]. These issues can appear some years later to an exposition with fibers. For this reason, different countries made new laws to forbid this material in upcoming constructions. Due to this, governments and companies are interested in having an inventory of asbestos rooftops for control, prevention and removal with the objective of substituting the existing roofs by other ones made of a material that is not harmful to the health of citizens.

Specifically, Spain stopped selling it in December 2001 but did not remove it from the existing rooftops. Currently, we keep this danger in our society. Due to this fact, it is crucial to locate it and eliminate it. The generation of an asbestos inventory is hard using traditional methods (human exploration). Human exploration involves a high cost in time and money. This project provides a tool that allows detecting asbestos in images from aerial or satellite sources. These images are easier to get and give us the possibility to reach a classification of larger areas in a short time.

With the focus on the detection of asbestos, researchers in Poland implemented simple machine learning techniques to explore its detection using computer vision systems (Review in section II) [4]. In this paper, we explore different computer vision techniques based on Deep Learning as segmentation, detection, classification and embedding systems. The investigation started using advanced techniques and continued to simpler ones in order to demonstrate that sometimes a complex system is not needed to solve a problem (Review in section IV). Also, this paper investigates and tries to explain whether rooftop pictures are sufficient to determine if a rooftop is made of asbestos or not. It is known that deep learning models use colours and shapes to generate a series of more complex features. These features represent our data and are useful to determine if an input belongs to the class or not.

Challenge we have to address is how to work with sparse data. To fix it, we work with pre-trained networks on COCO [2] or ImageNet [3] that can improve training the models from scratch, even if the images provided by these datasets are so different from aerial imagery. Furthermore, we explore how to use data augmentation to improve the results. We use data augmentation

Author: Kevin Martín Fernández, kevinmf94@gmail.com
Advisor 1: Agata Lapedriza Garcia, eHealth Center, UOC
Advisor 2: Javier Borge Holthoefer, IN3, UOC
Thesis dissertation submitted: September 2022

only with the minority class to deem if this approach can be a solution to compensate unbalanced data. In this case, classic data augmentation techniques such as flip, shift, scale, illumination and contrast changes are used to generate variants of the original data. Finally, we assume that the data has a high bias and we apply weighted sampling to improve the learning process.

II. State of the art

In this section, we will see the current state of the art. For this, we can contemplate four categories of interest for this project: Asbestos location, Aerial imagery on machine learning, Deep learning models for Computer Vision and Data augmentation.

A. Asbestos location

The interest for automatically asbestos location began to be investigated in the recent past due to the new damage demonstration to humans. Since it is a relatively recent topic, we found few research papers. This is becoming a new field of investigation. One of the first works addressing automatic asbestos detection was presented by Krówczyńska et al. [4]. The asbestos database collected for [4] is not open to the public currently. The database contains the data from the 'Chęciny' commune city from Poland. On 2015, this city had 39% of its buildings made of asbestos. Urban areas had 369 buildings, and rural areas, 2755. It can be observed in its paper, that data samples were images whose center corresponded to the center of the rooftop to be classified. After the asbesto's detection, 8.3 tones of it were removed on 2019.

In [4], researchers used the dataset to train a simple CNN with the following model: (1) Two convolutional blocks consisting of one convolutional layer, activation layer (ReLu), Max pooling layer and Batch normalization; (2) Spatial dro out layer (5%); (3) Two blocks consisting of: Fully connected layer, Dropout layer (50%) and a batch normalization layer.

The output of the architecture used a layer with two neurons and a softmax output to predict if the image has asbestos or not. In this work, they compare using RGB against infrared images, searching for another spectral bandwidth to improve the performance. But this work does not try to use combined data or use both to feed a neural network. RGB experiment obtained a 93.62% of F-Score [5] and infrared obtained 95.08%.

Due to the fact that the dataset used for that paper is not available, we can not replicate the work made. Even though, considering that with a simple method they achieved a good performance, it is interesting to take into account in our work using simple architectures with few parameters.

Later, Wu et al. [6] study the detection of hazardous materials in rooftops. The paper is focused on detection of hazardous materials in rooftops. In this investigation, they try to predict asbestos pipe insulation in multifamily houses and PCB joints or sealants in school buildings in two major Swedish cities: Gothenburg and Stockholm. The objective of the work is to classify rooftops materials considering text and numeric data from the buildings. The information used to detect asbestos is construction year, floor area and city. They train different machine learning methods to resolve the asbestos location. The paper reached an accuracy of 89% and recall of 92% in its better methods (Random Forest and XGBoost). We find these results interesting for doing multimodal learning and combine them with aerial imagery methods.

Finally, Trevisiol et al. [7] study different approaches to classify roofing materials using satellite stereo-pairs. In [7], tries to resolve a problem that can be helpful for rooftop detection but does not detect asbestos directly. The categories defined are Clay tiles, Sheath, Metal sheets, Gravel tiles, Gravel and Other materials. The detection of these materials may help improve asbestos detection if asbestos materials were detected in one of the defined categories. They used data provided from multi-spectral imagery and combined them using pre-processing data to feed a classification model. As a machine learning model, they used an SVM model to classify the samples. The results obtained in the research are 89.5% of accuracy.

B. Aerial imagery on Machine Learning

Aerial imagery includes images taken by aircraft, drones and other flying devices with an onboard camera. These pictures are useful to locate a collection of objects, like streets, buildings, pedestrians, or events, such as fires, traffic or jams. Therefore, the development of Machine Learning methods to classify these types of images is an active research topic.

In this field, we found some datasets, but we can highlight three:

Among the multiple aerial image datasets publicly available, we highlight the following three:

- **DOTA [8]**: DOTA is a large-scale dataset for object detection in aerial images. It can be used to develop and evaluate object detectors in aerial images. The images have been taken from different sensors and platforms. Each image of the collection has sizes between 800×800 to $20,000 \times 20,000$ pixels and contains objects exhibiting a wide variety of scales, orientations and shapes.
- **LandCover.ai [9]**: The LandCover.ai (Land Cover from Aerial Imagery) dataset is used in automatic mapping of buildings, woodlands, water and roads from aerial images. It has 33 orthophotos with $25cm$ per pixel resolution ($\sim 9000x9500px$) and 8 orthophotos with 50cm per pixel resolution ($\sim 4200x4700px$). Total area of 216.27 km^2.

- **AID [10]**: AID is a new large-scale aerial image dataset that collects sample images from Google Earth imagery. The dataset is made of the following 30 aerial scene types: airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks and viaduct. All the images are labeled by specialists in the field of remote sensing image interpretation.

We can highlight three papers that use aerial imagery:

1) TorchGeo: Deep Learning With Geospatial Data [11]: TorchGeo explores the usage of Geospatial data for deep learning models. This paper made an implementation of different samplers for geospatial data and analyzed the computer performance in the training process. Also, the paper analyzes the effect into accuracy of using pre-training with ImageNet. Datasets used for the experiments are provided from different satellital sources. This paper does not focus on what is detected, but on the improvement of computer performance.

The most interesting result for us is the effect of pre-train models with datasets that are not in the domain of aerial imagery. They compare four weighted initializations: Random, ImageNet (+ random), Cityscapes and In domain. It is interesting to explain "in-domain" initialization. The concept in-domain refers to the pre-training method in Neumann et al. [12]. "In-domain" consists in training models to ImageNet, then further training on remote sensing (dataset in the domain of aerial imagery) and finally fine-tuning with the actual target dataset.

We compared the results of pre-training with ImageNet and In-Domain. The accuracy performance obtained using ResNet50 [13] are 96.86%, 99.20% and 99.61% using the datasets RESIC45 [14], EuroSAT [15] and UC Merced [16] respectively. RESIC 45 is used for classification of scenes, it has 45 scenes with 700 images for each scene. EuroSAT is based on Sentinel-2 images using 13 hyperspectral bands; it has 10 classes with a total of 27000 images labeled and geo-referenced. UC Merced is a 21 class land using remote sensing image dataset, with 100 images per class. Fine-tuning in other domains is helpful to improve performance in the target domain. For this reason, we chose ResNet for this work.

2) Learning Multi-grain Instance Representation for Aerial Scene Classification [17]: The paper explains the influence of the objects depending on the point of view, height of capturing and more. We can find objects (in our case buildings) in different sizes depending on the height of capture and other variables of the camera. Larger object sizes provoke the division of them in multiple images. This division cause the neural network to have a partial view of it.

This work proposed a multi-grain detection model (Detection on different size levels) named AGOS with a comparable performance close to the actual state of the artx. AGOS is a plug&play module that can be combined with other models. It is based on a bag of words approach and the framework consists on three components after the CNN backbone. To be specific, the multi-grain perception module implements a differential dilated convolution on the convolutional features to get a discriminative multi-grain representation. We consider that this paper is interesting to resolve the size problems produced by aerial imagery.

3) Augmentation Invariance and Adaptive Sampling in Semantic Segmentation of Agricultural Aerial Images [18]: This work focuses in semantic segmentation for agricultural aerial imagery. They expose what they consider the two main problems in this type of segmentation. They put forward a solution based on invariance and adaptive sampling. The problems they found are the following:

- Top-down perspective implies that the model cannot rely on a fixed semantic structure of the scene, because the same scene may be experienced with different rotations of the sensor.
- There can be a strong unbalance in the distribution of semantic classes because the relevant objects of the scene may appear at extremely different scales.

C. Deep learning models for Computer Vision

This section summarizes the different deep learning models that we use in this work. There are popular architectures for computer vision problems such as classification, instance detection and instance segmentation. We will go through them in order of difficulty from high difficulty tasks (Segmentation) to simple tasks (Classification).

Instance segmentation is an area of computer vision that whose aim is to detect and segment objects pixel-by-pixel. The scenes can have multiple items of the same class in a single image (instance detection). To complete this task, we found some models in the current state of the art trained in COCO dataset as Swin Transformer V2 (SwinV2) [19], Vision Transformers Adapter for Dense Predictions (ViT-Adapter) [20], Mask-RCNN [21] and more.

The last improvement considering state of the art is the one based in Transformers models family [22]. These models implement a module named attention. Attention mechanism is a powerful module for different fields of research, not only in image applications. This method consists of the generation of a weighted representation reflecting the importance of the data for each

other. The importance is defined in different ways such as self-attention, cross-attention and others. You can find implementations of Transformers for audio, natural language processing, images and more. Some models based on Transformers are SwinV2 and ViT-Adapter and get AP results of 54.4% and 52.1% respectively (trained in COCO).

On the other hand, Mask-RCNN is a computer vision model composed for multiple modules with the objective of predicting multiple instances with its corresponding bounding box, segmentation and confidence. Mask-RCNN is an extension of Faster-RCNN [23]. It adds a new branch to the model to predict segmentation masks. The results obtained with Mask-RCNN reaches 40.3% of mAP. An important topic when we use a model is the computation power requirement. In this case, Mask-RCNN uses fewer computational resources compared to the resources needed to train transformers models.

In most cases, reduction of the task complexity is an improvement in the results. If a problem is faced using segmentation and it is transformed to only bounding box detection, the number of parameters and its complexity is reduced. For this reason, it is necessary to consider simpler models to handle the issue from the correct perspective.

Finally, we reviewed the simplest models whose aim is to classify objects that have been created so far. CNN can classify images in two types: if it has a class or not (binary classification) and what is the present class in the image (multi-class classification). In this area, we found as a state of the art similar works as ViT-G/14 [24], ResT V2 [25], ConvNeXt [26], all based on transformers. Also, we can highlight models such as ResNet* [27] or VGG [28] that are very important nowadays because the computer power needed is more affordable for small centers of research.

D. Data augmentation

Deep learning models for computer vision have the prerequisite of training models with a large quantity of data. Data necessity is due to the high number of parameters used for deep learning. The desired behavior of deep learning models is to generalize the solution to a problem to have a good performance with new inputs. For this generalization, the quantity of data that the model needs is very large. When the data is not enough, we need to use some techniques to acquire more data.

We used data augmentation to increment our data. Data augmentation consists in using actual data to generate new data with some differences that are useful to have more samples. This allows models to improve their performance with an initial small dataset. One study about its effectiveness is the paper “The Effectiveness of Data Augmentation in Image Classification using Deep Learning” [29]. The most common transformations are shift pixels, flips, rotations, change of illumination, noise and more.

We can consider synthetic data as data augmentation. In some papers like KITTI-CARLA [30] or Synthetic Data to Improve Deep Learning Models [31], we can see how the synthetic data helps to train deep learning models. Generally, the use of synthetic data requires complementing it with few samples of real data to make the model work properly with real world input.

DatasetGAN [32] is presented in 2021 developed by NVIDIA, the University of Toronto, Vector Institute, the University of Waterloo and MIT. DatasetGAN is a procedure that aims to generate a new synthetic dataset using GAN [33]. DatasetGAN models are trained with popular datasets of segmentation generating a latent space. We need only a few images with its segmentation ground truth to adapt the model to our problem, thanks to the latent space. GAN works like a factory. With it, we generate some new synthetic data with its respective segmentation masks. This approach is one of the most efficients to feed deep learning models that need a large quantity of segmented data. We consider this approach very useful to generate more segmented samples of asbestos.

There are two new papers published related to this work: the first one is BigDatasetGAN [34], which extends this work using ImageNet. It uses five segmented images of each class for the training step. The second one, "Application of DatasetGAN in medical imaging: preliminary studies" [35], which applies DatasetGAN for its research in medical applications.

III. Methodology

In this section we explore the dataset used in this work. We describe the data peculiarities and how we use these data to resolve the problem of asbestos detection.

A. Data

The data used is a collection of aerial images from the following towns of Cataluña: Badalona, Sant Adria del Besos, Bages, Castellbisbal, Cubelles, Gava, Viladecans, Ginestar, Hostalric, La Verneda, Montornes del Valles, Vilanova and Zona Franca. The data is provided by DetectA [36]. Detecta is a company that works on creating a solution to the first major challenge of asbestos eradication: to identify where it is and who is responsible for removing it.

DetectA segments the rooftops on the aerial imagery and classifies them into three groups: Asbestos, Non-Asbestos (Hard negative) and Non-Asbestos (Soft negative). Soft negatives mean that it is not guaranteed that it does not belong to asbestos.

Then, the aerial imagery is associated with its corresponding image provided by Google Earth, and some extra classes like Roads and Green Spaces are added to the annotations.

The original aerial images of towns are sliced into N images of 5000x5000 pixels in RGB format, saved in TIFF format to preserve the original quality and id classes in the mask. Then, these blocks of N images are processed to split the original images and generate a dataset to feed different models used in the section III-B.

Also, DetectA generates a specific dataset for testing that improves the quantity of asbestos correctly labeled. In summary, we have two groups of masks and their corresponding classes:

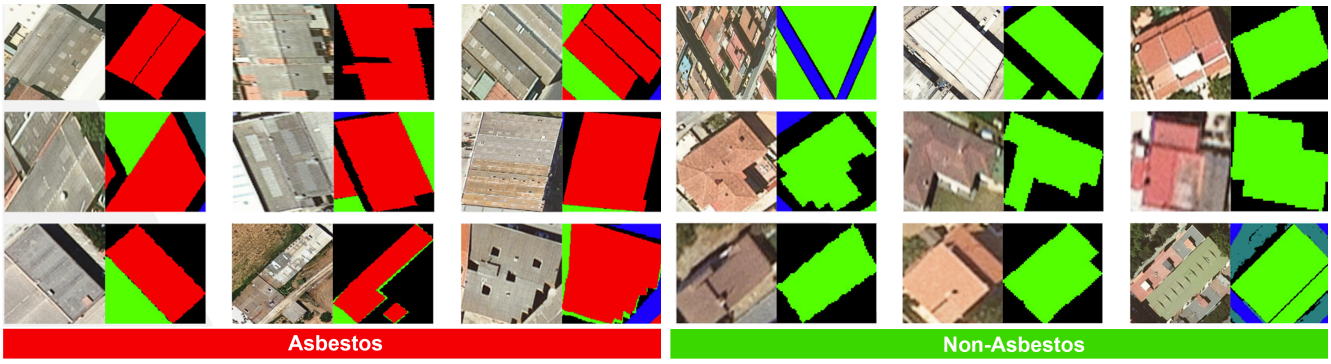
- **Training dataset**: Asbestos (1), Non-Asbestos (2), Streets (3), Green Spaces (4) and Others that not belong to any of the previously classes (5). This dataset has more data, but some of the rooftops labelled as non-asbestos are asbestos that were not correctly labelled because these rooftops were not checked.
- **Test dataset**: Asbestos (1), Non-Asbestos (Hard negative) (2), Non-Asbestos (Soft Negative) (3) and Others that not belong to any of the previously classes (4). This dataset has less data but has more rooftops correctly labelled as asbestos. Therefore, this dataset allows us to know the most accurate performance of our experiments.

More concretely, there are 7302 instances, of which 1022 are asbestos (14%), and 6280 are non-asbestos (86%). You can see the full summary of asbestos presence by region in Table I. The dataset is divided into three sets to guarantee the quality of the results. Training set had 6044 instances (82.77%), containing all the data except Bages and Zona Franca, which are reserved for validation and testing. Validation set had 250 instances (3.42%), and testing set, 1008 (13.80%).

Category/Zone	Badalona / Sant Adria	Castellbisbal	Cubelles	Gava / Viladecans	Ginestar	Hostalric	La Verneda	Montornes	Zona Franca	Bages	Total
Asbestos	121	21	33	63	39	29	36	21	79	580	**1022**
Non-Asbestos	582	443	1777	1326	171	220	455	707	38	561	**6280**

TABLE I: Towns included in our dataset along with the number of instances per town. Table shows the unbalanced data between asbestos and non-asbestos. Zona Franca and Bages are testing towns and their balance is better than others.

Notice that, the dataset has a high bias. The most represented class is non-asbestos. The representation of asbestos is very small compared with the quantity of non-asbestos images, and the distribution in the towns is very unbalanced. Low asbestos presence does not necessarily mean less danger for humans. The consequences of the exposure to this material are still dangerous for our health.



(a) Examples of rooftops with asbestos

(b) Examples of non-asbestos rooftops.

Fig. 1: Random selection of rooftop images with corresponding segmentation masks. Classes for the segmentation masks: Asbestos (Red), Non-Asbestos (Green), Streets (Blue), Dark green (Green Spaces) and Others (Black)

We observe a random collection of samples in the Figure 1. We can observe that asbestos rooftops have some shared characteristics. In some cases, the structures are of gray colours, shapes are commonly rectangles with a middle line and you can see how some of them have little rectangles as windows on the rooftop. On the other hand, houses or buildings that are non-asbestos in some cases have a brick or colourful colours. At this point we only have a simple analysis of the asbestos structures. Also, consider that asbestos could be painted in another colour and camouflaged by plants or other furniture. One purpose of this work is to understand if deep learning models can detect some pattern that allows the identification of asbestos samples.

B. Proposed methods

In this section, we will see how we pre-processed the imagery data. Finally, we will see the deep learning methods used to resolve the problem, the loss functions needed for the learning process and the metrics used to evaluate the model.

1) Data pre-processing: Pre-processing data is necessary because the original images are very big ($5000 \times 5000px$). We have to resize the dimensions of the pictures to the neural networks requirements. To achieve it, we considered two approaches (Python scripts can be observed in appendix A):

- **Patches**: The original image has a size of 5000×5000. We divided this data into patches of the same size (e.g.: 5000×5000 divided by 10 generates $10 \times 10 = 100$ patches of size 500×500). After this step, we resized the images to the required size by the neural network.
- **Centered roof top**: We only generate images with the building centered based on the segmentation mask. For this, we process the data for each class and mask, applying a blob detector that generates the location of each building. After this step, we resized the images to the required size by the neural network. Buildings have different sizes, which causes that small and large images are generated. Then, we need to resize to an equal dimension (upscale small pictures and downscale large pictures). The difference in sizes generates a dataset with multiple qualities. We can interpret inequality as a good feature in the data. The aerial imagery data can be provided from different sources, each source having a different quality, size, elevation, angle and other characteristics. We assume that this inequality can exist when receiving aerial images.

Resize of the masks is a critical point. We applied an interpolation method named "nearest pixel" to prevent incoherent value changes on the masks. Other interpolation methods combine the closer values and generate a weighted value. These methods would generate a different value from the expected one for the classes defined in the masks.

Which classes to use in the training process is another configuration to have in mind. It is interesting to explore how the other categories (Streets and Green Spaces) affect the decision to classify a rooftop as asbestos. For this reason, it is a great experiment to use these classes and compare them with the version of binary classification of asbestos. This configuration is only possible when the problem is segmentation.

2) Deep learning models: For the asbestos detection task, we considered different models to analyze the problem behavior depending on the model used. We have focused on three kinds of model: segmentation/detection, classification and triplet networks [37] (Python scripts can be observed in appendix A).

- **Mask-RCNN** is used for instance detection/segmentation. We used grid crop and building-centered crop with this model. Mask-RCNN is based on Faster-RCNN adding one branch to resolve the instance segmentation task. For our experiments we made fine-tuning with a Mask-RCNN, configured with a ResNet50 as backbone and pre-trained with COCO dataset. For all the experiments we used the Detectron2 implementation [38] that provides an environment for managing segmented datasets. The pre-trained model has average precision [39] results of 41.0% in detection and 37.2% in segmentation with COCO dataset. Between ResNet101 and ResNet50, we chose the ResNet50 as backbone because of the number of parameters. We assumed that using a lightweight model could be auspicious for learning a simpler task and reducing the computer performance needed.
- **ResNet** is the most common CNN model integrated into many new models because of its performance and lightweight compared to AlexNet [40] (ResNet50 has 26,6M of parameters, whilst AlexNet has 61,1M of parameters). ResNet implements modules of type residual blocks. In the end, residual blocks aggregate the output generated with the original input of the block. This behavior allows the network to avoid vanishing problems, helping train deep networks. The main task of ResNet is multi-category classification, but it is very common to use this network as base for other networks. For our problem, we need to adapt ResNet to a binary problem modifying the last layers. We changed the last layer to apply a sigmoid function. For this work, we used different ResNet implementations from the PyTorch framework [41] pre-trained in ImageNet.
- **Triplet Networks** are an extension of Siamese networks and part of the deep metric learning techniques. The objective of this model is to generate an embedding space where the samples of different categories are distinguished with a distance metric that can be, for example, the euclidean distance. Also, we can use this embedding space as a feature to feed other models. Siamese networks have two branches, negative and positive. Instead, Triplet Networks have three paths: anchor (sample randomly selected), positive (sample randomly selected from the same category that the anchor) and negative (sample randomly selected but does not belong to the anchor category). A Triplet Network combined with a Triplet Loss [42] optimizes the embedding space (anchor and positive samples stay together and negative samples are away from the positive ones). It can be observed as an example in the Figure 2, that a representation feature of 4 digits is generated. Triplet Loss puts together images that belong to number one group and keeps away the ones that belong to number two group.

  To detect asbestos using an embedding space we propose two different approaches. The first approach is to investigate if the latent space is enough to classify our data correctly. The second one is to check if using the latent space as a feature for a SVM [43] suffices to distinguish between the categories.

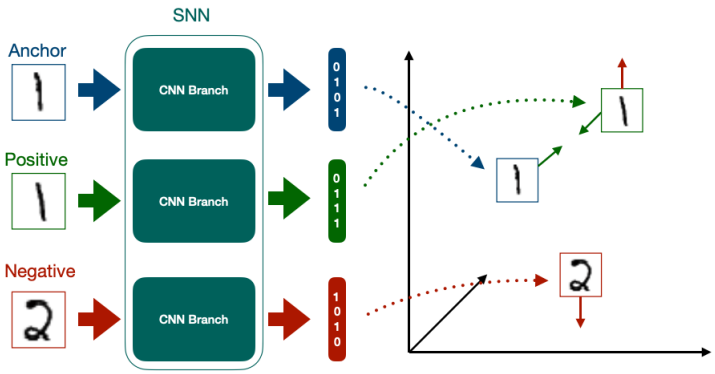


Fig. 2: Behavior of a Triplet Network. 1) Generation of an embedded representation for each input (anchor, positive, negative); 2) Optimization using the Triplet Loss. Anchor and positive optimized to be close to each other. Negatives are optimized to be away from anchor/positive.

In the experiments, we used ResNet50 (from PyTorch pre-trained in ImageNet) as a model. We used it for the three paths (shared weights) and used Triplet Loss to train the model. The output of ResNet50 is modified to generate a custom output of configurable size, which represents the features extracted from the images in the new embedding space.

3) Loss functions and evaluation metrics: To reach the best optimization of the deep learning models we chose these following loss functions:

- Mask-RCNN for segmentation (Detectron2): Combines multiple losses depending on the output branch. We used the default losses implemented in detectron2.
- ResNet50 for binary classification (PyTorch): Binary Cross Entropy Loss (BCELoss)
- ResNet50 for embedding (PyTorch): Triplet Loss (Constrastive Loss)

The metrics used for evaluation are average precision (AP), accuracy (ACC), sensitivity, specificity and confusion matrix. Aiming to compare our results with the classification results of other papers, we had to implement a final script to evaluate asbestos detection on Mask-RCNN. The code made a comparison of the outputs with the groundtruth. If the model detects any bounding box as asbestos, this is classified as asbestos to generate the metrics and confusion matrix.

We highlight that, the most important metrics are sensitivity and specificity. These two metrics measure how good the prediction of a concrete class is in binary problems. Sensitivity corresponds to the asbestos class and specificity to non-asbestos.

IV. Experiments and Results

In this section, we explain the performed experiments, their configurations, and the obtained results.

A. Experimenting using Mask-RCNN

For these experiments, we used the Detectron2 implementation that incorporates Model Zoo (Manager of models and its weights). We chose the **Mask-RCNN_R50_FPN_3x** configuration of Model Zoo. This configuration uses a ResNet + FPN backbone with standard convolutional and fully-connected heads for mask and box prediction, respectively. We considered results and performance in COCO. This configuration is the most balanced one and obtains the best speed/accuracy tradeoff.

Detectron2 configuration has many hyper-parameters to configure all the modules involved in the network. We only want to highlight some of the parameters from the original configuration:

- LR_SCHEDULER_NAME: Default value is “WarmupMultiStepLR”. In experiments, we could see how the learning rate starts from a little number and increase to our objective learning rate.
- IMS_PER_BATCH: Commonly named batch size, Detectron can play only with the batch size and calculate the number of epochs with the formula $Epochs = Iterations \times Batch\ size$. We set it to 45 for our dataset.
- ROI_HEADS.BATCH_SIZE_PER_IMAGE: This parameter corresponds to the number of regions of interest per batch during training. We set it to 256 for all the experiments.

(a) Image. (b) Mask. (c) Mask data over the image.

Fig. 3: Example image corresponding to Zona Franca. Colour legend for image 3c: Red are asbestos rooftops, blue are non-asbestos rooftops.

1) Patches as inputs:

The first approach to resolve this problem is using patches (see section III-B1 for the details), generating slices of the original images. The pictures before being sliced can be observed in the Figure 3. In this approach, we have in mind to check if the rest of the classes as Streets or Green Spaces can help to improve asbestos detection.

The results of asbestos detection in these experiments are closer to 10% of accuracy for the asbestos class. In the experiments using extra classes, we could not see any improvement. Finally, we discarded this approach to focus the effort on the centered roof tops. Anyways, we consider this one a research line to explore in future works.

2) Centered roof top as inputs:

This approach generates an image centered in the building. An example can be observed in Figure 1. The main idea of this experiment is to center the focus of the deep learning methods on the main object, removing noisy objects and simplifying the problem.

- **No Data Augmentation:** In the first experiment, we want to show how the original problem has a high bias. Due to the sparsity of asbestos, the dataset has a high unbalanced data issue. We will have to address this point to improve our work. We analyse the confusion matrix generated by the experiment. The detection of non-asbestos has an specificity of 100% (480 of 480 non-asbestos detected). This is an excellent result for detecting non-asbestos, whilst the asbestos category was detected in 10 of 518 samples (1.89%). It is a significantly wrong result for our objective category. The loss graphs of the Figure 4 show us that the model can learn the data but only focus on the non-asbestos class. We tested the model in some points of the training process when asbestos learning is upper, but the results are bad due to the high quantity of non-asbestos. The model learns to predict that all are non-asbestos, meaning that the dataset needs more asbestos to be representative.

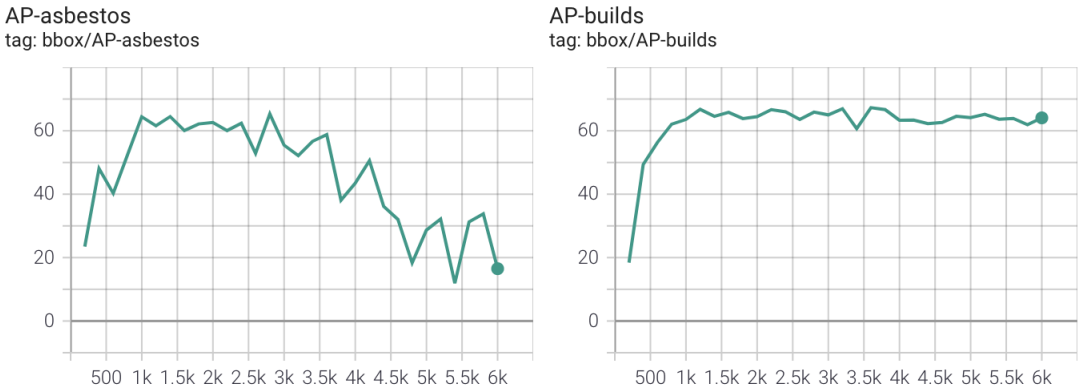


Fig. 4: Average Precision (AP) for asbestos (left) and non-asbestos (right) classes using Mask-RCNN without data augmentation and with centered rooftops. We observe how the asbestos class does not learn, instead of the non-asbestos. The results of the experiment output generates a 100% of specificity (non-asbestos detection) and 1.89% of sensitivity (asbestos detection). X-Axis corresponds to the number of iterations and Y-Axis to the number of AP.

The detection of asbestos is unsatisfactory. We can observe this in Figure 5. In order to improve it, we need more data to segment correctly the buildings. This data needs to be independent of its category.

- **With Data Augmentation:** As we could see in previous experiments, the category asbestos is hard to learn, the principal hypothesis for it is because of the sparsity of asbestos samples on the dataset. To avoid the problem, we considered applying data augmentation to the minority category. It allows the system to see more different samples of asbestos. The objective is to make equivalent the number of examples for both categories. In our tests, we implemented two approaches. Both approaches used Albumentations [44] framework to generate the data augmentation.

  In the first approach, we implemented a classical data augmentation technique consisting in augmenting data during the sampling process. We apply data augmentation when the samples are loaded from the hard disk. Even if the same sample is loaded, we generate a random variance of it. We had to develop custom components of Detectron2 (**DatasetMapper**) to apply the data augmentation only to the asbestos category. All the implementations can be observed in the code repository (Appendix A).

  Using this approach, we detected a problem. The issue is that, compared to the epoch without data augmentation, in the epoch in which we used this technique the number of samples belonging to each category is the same (e.g. in the epoch without data augmentation, it noticed 400 asbestos and 5000 non-asbestos, while in the epoch with data augmentation, the model saw the same amount for each category). To fix this, we also implemented our custom sampler to apply weighted sampling (**AsbestosWeightedSampler)**. The custom implementation allows us to define what samples are more likely to be chosen for this iteration. In this problem, we increment the probability of the asbestos class.

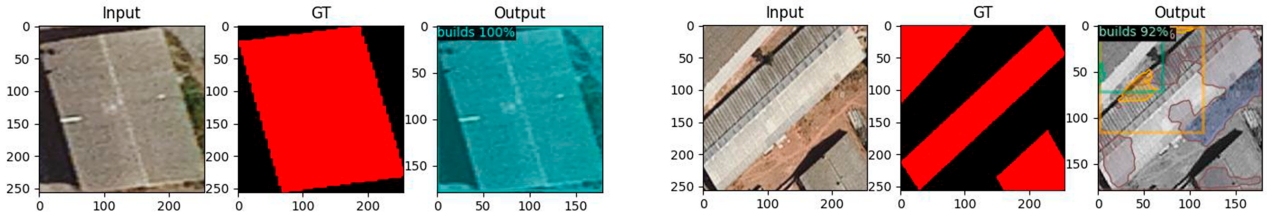
  For this experiment, we apply the next augmentations: HorizontalFlip, VerticalFlip, ShiftScaleRotate, RandomBrightness, and RandomContrast with a random probability of 0.5 and a weighted sampler. The sampler generates this distribution in each epoch 3970 (65.68%) for asbestos and 2074 (34.31%) for non-asbestos. We focus the analysis on the bounding boxes because segmentation is limited by bounding box detection. Results for asbestos classification resulted in sensitivity of 13.64% against the specificity of 98.12%. We concluded that this approach does not work for Mask-RCNN.

  A negative point of data augmentation in live is that it can mask the real samples. This impacts directly the learning process because Mask-RCNN has a harder problem to optimise its weights. We fixed it by generating a fixed number of data augmentation previous to the training step. With this method, we guarantee that the real data and augmentation data are learnt. Also, we ensure that all the data is processed in each training epoch.

  For this, we implemented a script to generate the data with the Albumentations framework. In this approach, the number of transformations are chosen previously. The number of samples generated can be calculated (e.g. if it has 8 images of asbestos and it is defined to apply 3 transformations, the script generates $8 \times 3 = 24$ images of augmentation). We define that all the transformations need to be applied with a probability of 100.0% - always. The intensity of transformations is kept random (e.g. the level of brightness to be increased or reduced is randomly chosen).

  There are 363 images of asbestos and 5681 of non-asbestos. We can highlight a series of experiments modifying the quantity of asbestos generated by data augmentation. In these experiments, as it can be observed in Table II, the augmentation of the number of samples helps to reduce the sparsity problem and improves the detection of the weak class. In consequence, it has a performance downgrade in the non-asbestos class. We chose static augmentation with 11 transformations as the best result for asbestos detection.

  Horizontal, Vertical and ShiftScaleRotate transformations help the model to be invariant to position, rotation and size as it can be observed in Figure 6. In aerial imagery, there are a big variety of buildings. For this reason, the generation of these



(a) Asbestos as non-asbestos with a correct segmentation.

(b) Asbestos as non-asbestos with wrong segmentation

Fig. 5: Qualitative examples of centered roof tops approach using Mask-RCNN without data augmentation. a) In the first image, asbestos is classified as a building, but the segmentation is accurate. b) In the second one, asbestos is detected, but the segmentation has a shattered representation of the building. This segmentation is not enough to know where pixels are asbestos.

	Generated images	Sensitivity	Specificity	Accuracy
Without augmentation	-	1.89%	100.00%	48.61%
Dynamic augmentation	-	13.64%	98.12%	53.86%
Static augmentation (7 transformations)	2541	40.53%	78.75%	58.73%
Static augmentation (11 transformations)	4356	**66.85%**	53.75%	60.61%
Static augmentation (16 transformations)	5808	45.07%	**82.70%**	**62.99%**

TABLE II: Summary of accuracies of asbestos detection using Mask-RCNN. It can be observed the baseline result (without data augmentation), dynamic augmentation result, and the comparison of the best approach using static augmentation. Static augmentation has a comparison changing the number of transformations and it can be seen the number of generated images for each one. All the experiments are generated with a score threshold of 0.7.

invariants can help the training process. RandomBrightness and RandomContrast make illuminance and tone changes. Aerial imagery has an uncontrolled light environment (light depends directly on the Sun) and images are acquired at different times and seasons. The diversity of illumination in the images can help the training process.

B. Experiments using ResNet

Mask-RCNN is a complex model to use with data sparsity. However, we get some reasonable results. That indicates that we can find a better solution with a simpler task. In this way, we chose an easier task, a binary classification. We experimented with ResNet50, ResNet34 and ResNet18. Finally, we decided to use ResNet18 for two reasons: the results with this model are equal or better in performance and, in consequence, we consider that the usage of a model with fewer parameters can help the training process.

Generally, in all the experiments, we used the following configuration:

- Loss function: BCELoss
- Optimizer: SGD
- Epochs: 50
- Batch size: 100
- Learning rate: 0.01
- Network output: Linear(512, 1) + Sigmoid
- Pre-trained: ImageNet

In ResNet, we only implemented the live generation of data augmentation since the requirements for data for a ResNet are less. Because of it, we focused our tests using this approach. We used the implementation of ResNet inside the PyTorch framework. Using PyTorch implies the definition of a class that represents our data. PyTorch, for images, has a class named

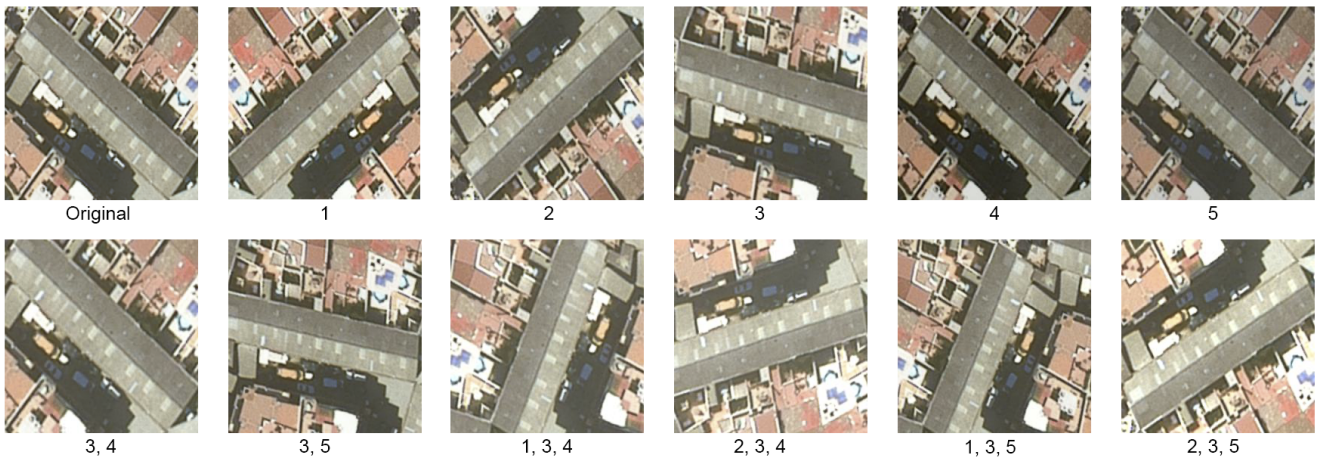


Fig. 6: Example of augmentations applied to one asbestos roof top. Numbers are transformations applied to the image. Horizontal Flip (1), Vertical Flip (2), ShiftScaleRotate (3), RandomBrightness(4), RandomContrast(5). We can see in these images how data augmentation generates images with multiple rotations, illumination and colour changes. This behaviour is the expected. Acquisiton of aerial images could be provided from differents rotations and differents light conditions. We consider these augmentations helpful to generalize the objective task.

"VisionDataset". We need to implement VisionDataset and modify it to generate data augmentation only for asbestos class using the Albumentations frameworks, as in previous experiments. Like we made in Mask-RCNN, we also implement a weighted sampling. In the case of PyTorch, there are existing classes already that help to manage this weighted sampling, so we didn't have to implement them.

For output metrics, we used Torch Metrics [45] as framework to generate all the results. Torch Metrics are an implementation of the most popular metrics friendly to PyTorch framework. The code is available in the appendix A.

	Aug. Prob.	Weighted [Non-Asb, Asb]	Sensitivity (Asbestos)	Specificity (Non-Asbestos)	AP	Accuracy
a)	-	-	17.61%	**98.96%**	84.05	56.35%
b)	0.3	-	32.39%	96.46%	82.39	62.90%
c)	0.3	[0.1, 0.9]	65.15%	83.33%	85.04	73.81%
d)	0.3	[0.1, 1.5]	74.43%	78.54%	86.66	76.39%
e)	0.3	[0.1, 2.0]	76.52%	74.79%	86.49	75.69%
f)	0.3	[0.1, 4.0]	86.55%	68.33%	**87.41**	**77.88%**
g)	0.3	[0.1, 8.0]	**91.86%**	55.62%	86.12	74.60%
h)	0.5	[0.1, 2.0]	81.63%	72.50%	85.37	77.28%
i)	0.5	[0.1, 4.0]	90.34%	55.83%	84.34	73.91%
j)	0.7	[0.1, 2.0]	78.60%	67.92%	83.81	73.51%

TABLE III: Results of the most relevant experiments using ResNet18 with a classification threshold of 0.5. "Aug. Prob. ": Probability of applying a transformation - all transformations have the same probability."Weighted": the first number corresponds to non-asbestos importance, second one to asbestos importance. b) The use of data augmentation increases the performance of sensitivity in 1.83x. e) The use of data augmentation with weighted sampling allows us to find a balanced result. f) We chose it as the best result even if the non-asbestos suffered a downgrade. We had an improvement of 4.34x in sensitivity. g) It has the best performance for sensitivity but in consequence the worst result in specificity.

We can see how the usage of data augmentation without weighted sampling gives us an improvement of 1.83x in sensitivity against asbestos (Table III-B). We thought that this could be improved by using weighted sampling. Therefore, we used it to manage the importance of asbestos samples. The increment of asbestos in the sampling weights makes the sensitivity to increase. Consequently, the specificity downgrades its performance as we can see in the Table III. We generated a graph to visualize the influence of weighted sampling, as it can be seen in Figure 7. Also, we analyzed the graph to find the best performance. We considered that the best performance for both categories is found in the intersection of sensitivity vs specificity (Table III-E).

We took into account that asbestos detection is more important for the purpose of this work. We needed to define some criteria to choose the best model. One possible criteria is to allow some performance downgrade in the non-asbestos category. To solve our task, we could tolerate that some non-asbestos building is detected as asbestos. Since an expert in asbestos can, in posterior steps, discard it with a visual exploration of the roof top, the priority is to detect the most asbestos rooftops as possible.

We chose experiment of Table III-F as the best. The high sensitivity (86.55%) and the moderate downgrade of non-asbestos to 68.33% reaches 87.41 of AP. It is so far the best model with an improvement of 4.34x respect to the baseline.

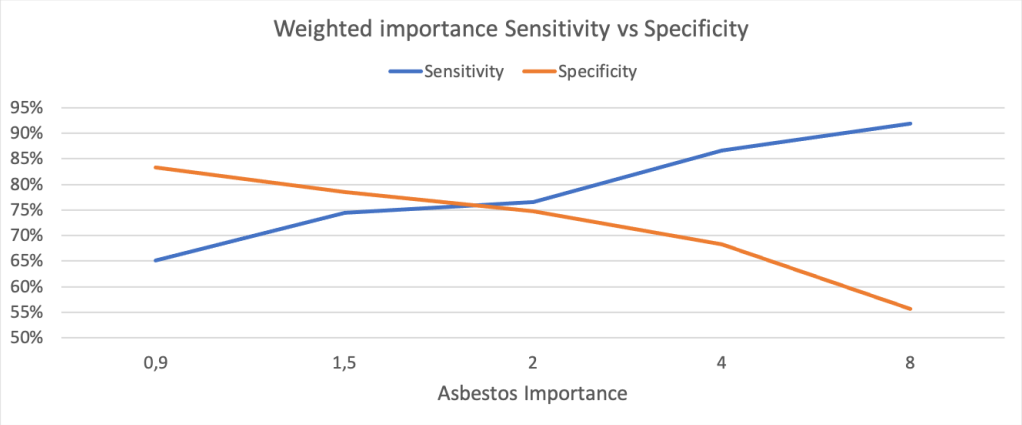


Fig. 7: Influence of asbestos weighted importance in terms of Sensitivity (Asbestos detection) vs Specificity (Non-Asbestos detection). X-Axis: importance of asbestos probability (not necessary 0-1), Y-Axis: Sensitivity or Specificity (0-100%). We can see in the graph how the increment of asbestos importance increases the sensitivity and downgrades the specificity. We need to find a balanced performance for our project.

C. Experiments using Triplet network

Some rooftops labelled as non-asbestos are, actually, asbestos. We thought about other methods that are more robust to not-accurate labelling. For this topic, we wanted to test embedding spaces. The hypothesis of these experiments is to check if a contrastive loss could generate discriminative features that separate our samples. Finally, we analyzed if the representation generated by an embedding space is better than a CNN feature.

1) Embedding space + t-SNE:

To create our embedding space, we used PyTorch, and we reused the implementation made for ResNet with some modifications. We needed to implement a Dataset that returns three images (anchor, positive and negative) for training, as it can be seen in the loading schema of Figure 8. The anchor should be a random sample, the positive should be a random sample belonging to the same class as the anchor and the negative should be a random sample of the opposite class. When the anchor is asbestos, instead of searching for an asbestos sample we generate a positive class using data augmentation.

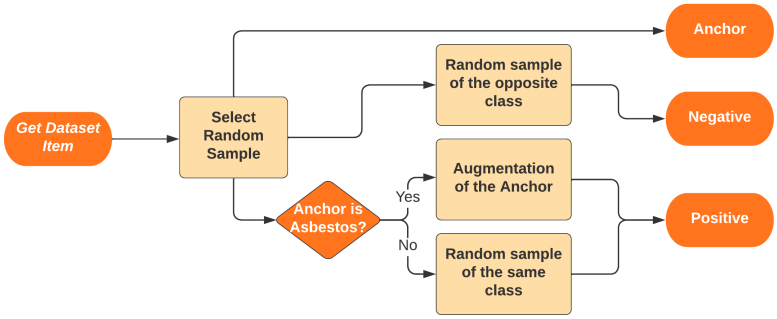


Fig. 8: Workflow to load one database item for training the triplet network. 1) Choose a random sample as Anchor, 2) Choose another random sample of the opposite class as Negative, 3) If the anchor is asbestos, generate a random data augmentation of the anchor. If it is non-asbestos, get a random sample of non-asbestos. At this point, we have the three samples needed to optimise the neural network with a triplet loss.

For training, we used the same hyper-parameters of the best ResNet approach and configured the neural network output to generate a feature of size N. In this case, we change the fully-connected layer to set a linear layer of the desired output size. In the experiments, we tested with output sizes of 128 and 256. To analyse the results of a triplet network output we used t-SNE [46], which makes a dimensional reduction, allowing us to display the embedding space in a 2D representation.

Qualitatively, the results visualizing t-SNE are very similar between the two experiments made. For this reason, we only analysed the output size of 128. t-SNE has two main hyper-parameters: perplexity and iterations. We tested some perplexity values and we set the iterations to 5000 for all the experiments. As we can see in Figure 9, the separation between classes is not enough to train a k-NN or a clustering representation in this dimension. Therefore, we decided to test in a bigger dimension using a SVM as it can be seen in the next section.

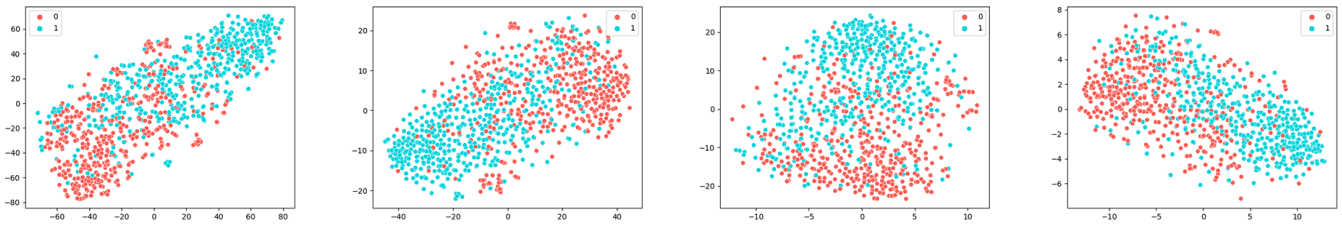


Fig. 9: t-SNE representation for different values of perplexity of the dataset (Iterations = 5000). Red (0): Non-Asbestos, Blue (1): Asbestos. X-Axis and Y-Axis are the locations in the latent space after applying t-SNE. As it can be seen in the images, the distance between classes is not enough to work with this dimension.

2) Embedding + SVM:

In the previous section, we tested to generate an embedding space and visualise it. We determined that it is not enough to represent our problem. In this section, we train a SVM model with the output features. We used the training dataset to generate

embedding representations and then train an SVM. For the training process, we tested with different hyper-parameters. The hyper-parameters we changed are: the type of kernel, class weights and change the regularisation parameter C.

	ResNet output	Kernel	Degree	C	Sensitivity	Specificity	Accuracy
a)	256	RBF	-	1	39.43%	**95.32%**	66.92%
b)	256	RBF	-	1	49.13%	91.75%	70.90%
c)	256	Poly	3	1	66.81%	78.39%	72.50%
d)	256	Poly	4	1	73.70%	72.38%	**73.05%**
e)	256	Poly	5	1	78.87%	65.03%	72.07%
f)	256	Poly	6	1	82.32%	59.68%	71.19%
g)	256	Poly	7	1	87.06%	55.45%	71.52%
h)	256	Poly	8	1	88.79%	49.66%	69.55%
i)	128	Poly	3	1	70.04%	75.95%	72.94%
j)	128	Poly	4	1	76.72%	68.60%	72.72%
k)	128	Poly	5	1	81.89%	60.80%	71.52%
l)	128	Poly	6	1	87.06%	54.12%	70.86%
m)	128	Poly	7	1	88.79%	47.88%	68.67%
n)	128	Poly	8	1	**90.73%**	41.65%	66.59%
o)	128	Poly	4	0.25	83.62%	59.02%	71.52%
p)	128	Poly	5	0.25	88.14%	49.22%	69.00%
q)	128	Poly	5	0.50	85.77%	56.12%	71.19%

TABLE IV: Results of training a SVM with the features of the embedding space. We used a threshold of 0.5 for classification. All the experiments except the experiment a) have a weighted sampling. The weighted sampling configurations are 0.1 for non-asbestos and 2 for asbestos. We can highlight how the increment of sensitivity downgrades the specificity, as in previous experiments. Parameter C has an improvement into the sensitivity, but increments the distance between sensitivity and specificity. The increment of the degree, improves the sensitivity performance and downgrades in specificity are lower than the ones with C parameter.

For training the SVM, we kept using weighted sampling, as in previous experiments. All the experiments used class weights configured with 0.1 for non-asbestos and 2 for asbestos, except the experiment Table IV-A.

We can highlight that in RBF, we could not reach the expected results. The polynomial implementation of SVM (poly) allows us to play with the degrees of freedom. Low degrees keep off the model to fit the data (underfitting) and high degrees provoke fitting the data (overfitting). We can see how the increment of the degree improves the results in Table IV. In the same way as in previous methods, the increment of asbestos detection downgrades non-asbestos detection.

If we search for comparable results with the ResNet approach, we can not find a balance. When results are closer to 86% in terms of sensitivity, the downgrade of specificity is deeper than in ResNet. With the same criteria as in previous methods, we chose the experiment IV-E as the best result for embedding spaces with SVM.

V. Results comparison

In this section, we show a summary of best results for each experiment configuration and we compare it with the baseline. We also show qualitative results of the best Mask-RCNN and ResNet models. Finally, we discuss the common characteristics shared by the models that produce the best results.

A. Quantitative results

We can highlight how all methods have a high sensitivity to unbalanced data. This fact makes it easier for the models to learn to detect all the buildings in the pictures, converting them into a good building detectors, even if at the moment they are bad asbestos detectors. Also, all the methods have a coherent behaviour - the increment of weighted sampling generates a downgrade in the specificity and a better performance in the sensitivity.

We have to search the trade-off between loss of accuracy in non-asbestos and detection of asbestos. We have to keep in mind that our main objective in this work is the detection of asbestos. As we can see in Table V, ResNet baseline has better results, due to the fact that the task to resolve is an easier one. In the final results, the best one is ResNet using data augmentation and weighted sampling. ResNet has the best improvement of sensitivity (+68.94%) with the smallest downgrade in specificity (-30.63%).

If we design a workflow with these models, it is possible to generate an automatic detection of asbestos pipeline. An example workflow with these results could be: 1) We use Mask-RCNN in its baseline configuration to detect all the buildings (asbestos are buildings in any case). Mask-RCNN segments images and, in consequence, can detect more than one building per image; 2) Then, we use the ResNet that has the best performance in asbestos detection to determine its presence.

Model	Sensitivity	Specificity	Accuracy
Baseline Mask-RCNN	1.89%	100.00%	48.61%
Best Mask-RCNN	66.85% (+64.96%)	53.75% (-46.25%)	60.61% (+12.00%)
Baseline ResNet	17.61%	98.96%	56.35%
Best ResNet	**86.55%** (+68.94%)	**68.33%** (-30.63%)	**77.88%** (+21.53%)
Best Embedding space + SVM	78.87% (+61.26%)	65.03% (-33.93%)	72.07% (+15.72%)

TABLE V: Selection of the best results from each method. We can highlight how all the methods have an increment in sensitivity and a downgrade in specificity. The best model is ResNet with 77.88% of accuracy. All the best methods used data augmentation and weighted sampling. While green numbers represent the improvement with respect to the baseline, red numbers represent the downgrade.

B. Qualitatively results

In this section, we check the outputs of the best models for Mask-RCNN and ResNet. It is important to understand what happens inside our machine learning systems.

1) Segmentation:

To analyse the segmentation results, we chose six samples that represent the most common behaviours we found. In the first instance, we show a correct example. Asbestos is detected and the mask precision is very accurate. We can see it in Figure 10a. In spite, we want to search the limits when the model fails. Fails generate more interesting knowledge than hits, and make us search for better solutions.

We highlight the following behaviours when the model fails:

- When we detect asbestos in the middle of an image, if it has surrounding buildings, these buildings also are detected as asbestos. This behaviour could be possible. As we explained in previous sections, the dataset has buildings that are asbestos but are not in the catalogue. In the case of Figure 10b), the surrounding buildings look like asbestos for their colour and structure.
- We observe how the segmentation has a low precision in many cases. As we can see in Figure 10c and 10d, we can determine what class is in the image but not the pixels corresponding to the class.
- In many deep learning researches the CNN features represent colours, textures and/or corners that generate high level features. Regarding the rooftop colours and textures, our model has the next behaviour:
  - Asbestos is detected when rooftops have grey tones with a high frequency (textures). We can see some examples of it in Figures 10a, 10b and 10c.
  - Non-Asbestos is detected when the colours of the rooftops are red or brown. We can see an example in Figure 10e.
  - We detected that white rooftops are in most cases detected as asbestos. These rooftops are always non-asbestos. The main hypothesis we have for this is that rooftops have some grey rectangles, as we can see in Figure 10f. This provokes the detection of a textured rooftop that is classified as asbestos. This behaviour is present in some outputs of our testing set.

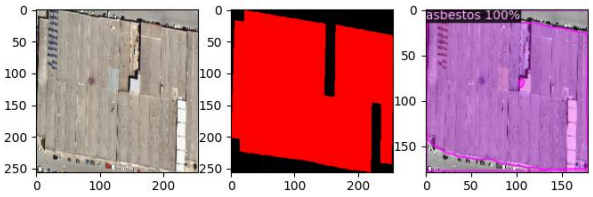
2) Classification:

For analysing the classification results, we made a prediction of all the test set ordering the results by confidence. The top 18 images are asbestos and we can see how the most common colour in the rooftops is grey with textures, as we can see in Figure 11a. We can consider this as a good signal. The asbestos representations learnt by the model have common characteristics.

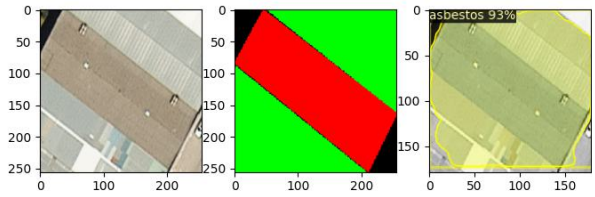
The first error in classification is found in the position 22 of the ranking. As we can see in Figure 11b, the wrong detection is very similar to asbestos. The sample could be asbestos with a wrong label. The first detection that is really non-asbestos is found in the position 243. Sample 243 is classified as negative (non-asbestos), but it is not hard-negative, meaning that its belonging to the category is not confirmed. You can see the sample 243 in Figure 11c.

Similarly to when we analysed Mask-RCNN, we found the same conclusion in this method. Colours and textures seem to be a determinant characteristic to classify samples. Also, if we focus the analysis in the non-asbestos class, as we can see in Figure 12, we also arrive at the same conclusions.

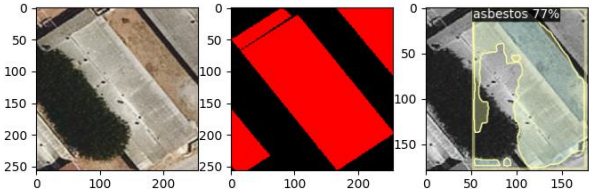
The amount of samples predicted as asbestos is close to the 60% of samples, as we can see in Figure 13a. Testing distribution has 52.38% for asbestos. This fact lets us notice that there is something wrong with the results. The percentage of samples deemed as asbestos should be closer to 50%. We made a qualitative zoom in the thresholds between 0.65 and 0.59. As it can be seen in Figure 13b, these results can demonstrate that some of the samples labelled as non-asbestos also have the appearance of asbestos. We think that, with a revision of the dataset labelling, specially for the cases which are not certain to be labelled properly, the performance of our model could be improved.



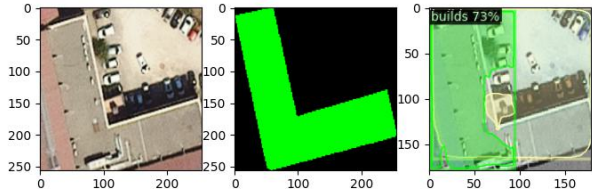
(a) Asbestos correctly detected.



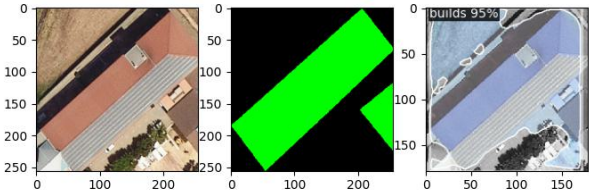
(b) Asbestos correctly detected surrounding with non-asbestos buildings.



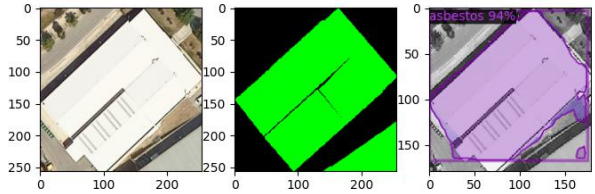
(c) Asbestos correctly detected with wrong segmentation.



(d) Non-Asbestos are correctly detected with gray roof top and wrong segmentation.



(e) Non-Asbestos correctly detected with brown roof top and some texture.



(f) Building wrong detected with white roof top.

Fig. 10: Qualitative results of segmentation using best Mask-RCNN. a) In this Figure we can see a correct detection with precise segmentation; b) Asbestos is detected but the surrounding buildings are non-asbestos and the segmentation classifies them as one unique item; c) Asbestos is detected but the segmentation is imprecise; d) Non-Asbestos is detected but the segmentation is imprecise; e) Non-Asbestos has a characteristic colour for liveable housing. Also, it has some texture but the brown colour makes it classify them as non-asbestos; f) Example of a detection issue. All the white rooftops are detected as asbestos but they actually are non-asbestos.

VI. Conclusions

Asbestos rooftops are still harmful and present in a high percentage of our cities around the world. Sanitary authorities determined that asbestos is dangerous. Removing this material from buildings is an important objective for the society. In this way, computer engineering and, more concretely, computer vision fields, can help to locate asbestos rooftops present in our environments. The localisation of them helps companies and governments to control their existence and remove them.

Deep learning methods need a large quantity of data to train and give good results. The quantity of data available for this work is scarce. In consequence, the difficulty of asbestos detection increases. The quantity of data is unbalanced, and some of the buildings that are actually made of asbestos are not labelled properly. The problems we found with data are a consequence of the complexity of acquiring this data. Because of this, we have to improve the results with the available data that we have considering techniques such as data augmentation and weighted sampling. Data augmentation and weighted sampling techniques improved our results by +68.94% of sensitivity (corresponding to the asbestos detection). They allowed us to convert a hard problem into a feasible problem.

However, the use of techniques to avoid scarcity means a downgrade of non-asbestos detection. Without data augmentation, we have performances closer to 100% for non-asbestos class. The use of data augmentation plus weighted sampling provokes a decrease of -30.63% and -46.25% in specificity for ResNet and Mask-RCNN respectively. The use of these techniques is good to fix scarcity of data, but we have to consider the trade-off between detecting the minority class and deterioration of the majority class. As we can see in the experiments, the three models tested had similar issues: increase and decrease of sensitivity and specificity, white rooftops deemed as asbestos when they are not and the detection of surrounding buildings as asbestos if the main target is classified as asbestos. All these issues may have been provoked by the quality of the labelling of the data and the amount available of it to train the model.

We used data augmentation for two approaches: static and live data augmentation. For the Mask-RCNN model, the use of live data augmentation did not help to improve the results and we needed to use static data. The use of static data worked better

(a) Top 18 results of testing set.

(b) First samples wrong classified but with asbestos aspect.

(c) First samples wrong classified but with building aspect (Reference to image 4 from first column).

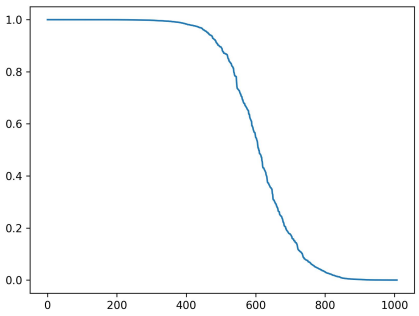
Fig. 11: Qualitative results of classification with the best ResNet (Data augmentation + Weighted sampling). Number means the confidence that is asbestos (Numbers are rounded to two decimals). Colours: Green are asbestos and Red are non-asbestos in the groundtruth. a) We can see the first 18 examples of the training set, as we can see all these have gray colours in its rooftops. b) Corresponds to the position 22 (First image of first row) of the ranking we found the first non-asbestos classified as asbestos, but the non-asbestos looks like an asbestos (It can be a wrong labeled case). c) The first non-asbestos detected (with non-asbestos aspect) is in the position 243 (Image four of the first row). It is detected as asbestos and clearly looks like non-asbestos. This rooftop could have been painted.

with Mask-RCNN, due to the fact that it brought us two important behaviours: the number of samples seen in each epoch were all the data we have plus the generated one, and the use of this approach acted as a weighted sampling (the number of samples for asbestos and non-asbestos is more balanced). Otherwise, for ResNet, the use of live data augmentation worked better, but it was not enough to learn the classes.

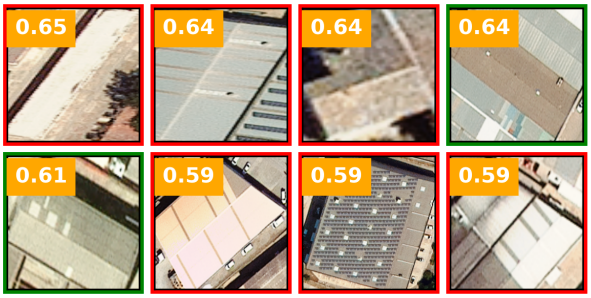
When we had unbalanced data, we needed to force the model to see more examples of the minority class. For this purpose, we used weighted sampling. ResNet did not work well only with data augmentation (it reached 32.39% of sensitivity to asbestos). The implementation of weighted sampling brings us a performance increase of 86.55% of sensitivity in the best model. We concluded that the use of data augmentation is not enough, and the most useful way to address the scarcity of data was the usage of weighted sampling.

To locate asbestos, we were interested to have a segmentation that detects multiple instances of it. We used Mask-RCNN to resolve this task. As we can see in the experiments, Mask-RCNN could not perform a good segmentation due to the fact of

Fig. 12: Examples of images classified as non-asbestos in the best model of ResNet (Data augmentation + weighted sampling). The samples have common characteristic as can see the brown/red colours. Some of the gray samples look like asbestos and could be a false negative.



(a) Confidence score graph. X-Axis: Position of samples in order, Y-Axis: Output score of ResNet



(b) Samples between 0.65 and 0.59 of confidence.

Fig. 13: Qualitative results of classification with ResNet18. Number means the confidence that is asbestos. Green are asbestos and Red are non-asbestos in the ground-truth. a) Graph that represents the behavior of confidence for our test samples. We have 52.38% of asbestos. If we use a threshold of 0.5 the first detection of non-asbestos success is around 60.0%. The distribution has been displaced to detect asbestos. b) We can see some examples from 0.59 to 0.65, these examples look like asbestos and maybe have a bad labeling.

the data scarcity and the large number of parameters it has.

We had to consider that one hard problem could be simplified in easier-to-solve and smaller issues. The transformation of segmentation problems into a classification problem using ResNet allowed us to improve the results. From 66.85% to 86.55% of sensitivity, from 60.61% to 77.88% of accuracy. We also reduced the loss of performance in non-asbestos from -46.25% to -30.63%. Embedding spaces were not enough to solve the problem, but the results are closer to ResNet and better than Mask-RCNN.

Making a simple analysis we can see how the behaviour of our neural networks is based in colours, textures and shapes. These characteristics of the rooftops seem to influence the decision of the classification. Grey colours and textures with some rectangular shape are detected as asbestos. On the other hand, when we have white buildings that we know they are non-asbestos, they are detected as if they were made of asbestos. These whites rooftops have grey rectangles in the roof. This can provoke the detection of a textured rooftop that it is therefore classified as asbestos. Also, we can highlight that the scenario could be different in other countries. The use of colours, structures and/or textures in construction can be different and, in this case, the model learnt only from towns in Catalonia. These are not enough to generalise other countries.

To sum up, we can conclude that deep learning models can help to detect asbestos and can be useful for governments and companies in order to remove it. All of us can collaborate with society to eliminate this kind of material and improve our environments.

A. Future work

Asbestos detection does not finish in this paper. This problem needs a full system that automates the work and searches for an improvement of its detection. Here can be found some interesting research lines for future works:

- We can explore the patches approach. The analysis in this work is superficial and the usage of more data can improve the results. In other works, the use of environment data (Streets, Green Spaces and surrounding buildings) improved the main objective.
- The amount of data used in this work and the quantity of known data is small. Maybe we can explore what would have happened if we used more data labelled as asbestos, soft non-asbestos and hard non-asbestos (in this paper we only had fewer amounts of this kind of data for testing). This may allow us to train models that are more flexible to soft non-asbestos, focusing the attention in the hard labels.
- We can explore the usage of other models. We read some interesting papers using U-Net as segmentation. U-Net has fewer parameters than Mask-RCNN. The use of U-Net can improve the segmentation task and can allow instance detection.
- We can make a deeper analysis of the neural networks behaviour using dissemination techniques. This would help us to understand what happens inside our system and generate really reliable conclusions, because we would be seeing the parts of the images that activate the neurons.
- We can use multiple models to solve this problem. We could put to work a good model in building detection (building detection is easier to learn and get a dataset), and use this model to detect building instances. Then, we could use a model as the one present in this paper to determine the presence of asbestos.

- Learn a multi-modal model. We can improve the embedding space approach by using extra data, for example, using image and numerical data (such as year, construction meters and else) as input of our network.

APPENDIX A
GITHUB PROJECT

All the code generated for this project can be seen in https://github.com/kevinmf94/Asbestos-detection. It also includes the links to Google Colab to do tests on ResNet and Mask-RCNN models with your own pictures or any of the predefined examples.

ACKNOWLEDGMENT

I would like to thank my advisors Agata and Javier, for supervising my work and collaborating with me in this project. I would like to thank also DetectA for sharing with us the data and their knowledge on asbestos, Davoud for adapting the data specifically for this work, Cristina for the support and explanation of other works that could be useful for this one, and finally, I would like to thank Yaiza for helping me with the language and my family for all their support throughout the entire process.

REFERENCES

[1] Bartrip, P W J. History of asbestos related disease. https://pmj.bmj.com/content/80/940/72, last access: 28/05/2022.

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 (2014)https://arxiv.org/abs/

[3] ImageNet. https://www.image-net.org/about.php, last access: 02/06/2022.

[4] Małgorzata Krówczyńska, Edwin Raczko, Natalia Staniszewska, Ewa Wilk. Asbestos–Cement Roofing Identification Using Remote Sensing and Convolutional Neural Networks (CNNs). https://www.researchgate.net/publication/338880523_Asbestos-Cement_Roofing_Identification_Using_Remote_Sensing_and_Convolutional_Neural_Networks_CNNs

[5] F-Score Wikipedia. https://en.wikipedia.org/wiki/F-score. Last access: 05/08/2022.

[6] Pei-Yu Wu, Claes Sandels, Kristina Mjornell, Mikael Mangold, Tim Johansson. Predicting the presence of hazardous materials in buildings using machine learning. https://www.sciencedirect.com/science/article/pii/S0360132322001408, last access: 26/08/2022.

[7] Francesca Trevisiol, Alessandro Lambertini, Francesca Franci, Emanuele Mandanici. An Object-Oriented Approach to the Classification of Roofing Materials Using Very High-Resolution Satellite Stereo-Pairs. https://www.mdpi.com/2072-4292/14/4/849/htm, last access: 26/08/2022.

[8] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, Liangpei Zhang. DOTA: A Large-scale Dataset for Object Detection in Aerial Images. arXiv:1711.10398 (2017)

[9] Adrian Boguszewski, Dominik Batorski, Natalia Ziemba-Jankowska, Tomasz Dziedzic, Anna Zambrzycka. LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands, Water and Roads from Aerial Imagery. arXiv:2005.02264 (2020)

[10] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang. AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification. arXiv:1608.05167 (2016)

[11] Adam J. Stewart, Caleb Robinson, Isaac A. Corley, Anthony Ortiz, Juan M. Lavista Ferres, Arindam Banerjee. TorchGeo: Deep Learning With Geospatial Data. arXiv:2111.08872 (2021).

[12] Maxim Neumann, Andre Susano Pinto, Xiaohua Zhai, and Neil Houlsby. In-domain representation learning for remote sensing. arXiv:1911.06721 (2019).

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. arXiv:1512.03385v1 (2015).

[14] Cheng, Gong and Han, Junwei and Lu, Xiaoqiang. RESIC45 Dataset. http://dx.doi.org/10.1109/JPROC.2017.2675998, last access: 05/09/2022.

[15] Helber, Patrick and Bischke, Benjamin and Dengel, Andreas and Borth, Damian Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification https://www.researchgate.net/publication/319463676_EuroSAT_A_Novel_Dataset_and_Deep_Learning_Benchmark_for_Land_Use_and_Land_Cover_Classification, last access: 05/09/2022.

[16] Yang, Yi and Newsam, Shawn. Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification. http://weegee.vision.ucmerced.edu/datasets/landuse.html, last access: 05/09/2022.

[17] Qi Bi, Beichen Zhou, Kun Qin, Qinghao Ye, Gui-Song Xia. All Grains, One Scheme (AGOS): Learning Multi-grain Instance Representation for Aerial Scene Classification. arXiv:2205.03371 (2022).

[18] Antonio Tavera, Edoardo Arnaudo, Carlo Masone, Barbara Caputo. Augmentation Invariance and Adaptive Sampling in Semantic Segmentation of Agricultural Aerial Images. arXiv:2204.07969 (2022).

[19] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, Baining Guo. Swin Transformer V2: Scaling Up Capacity and Resolution. arXiv:2111.09883 (2021).

[20] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, Yu Qiao. Vision Transformer Adapter for Dense Predictions. arXiv:2205.08534v2 (2022).

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. Mask R-CNN. https://arxiv.org/abs/1703.06870v3, last access: 29/05/2022.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762 (2017)

[23] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497 (2015)

[24] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, Lucas Beyer. Scaling Vision Transformers. arXiv:2106.04560v2 (2022)

[25] Qing-Long Zhang, Yu-Bin Yang. ResT V2: Simpler, Faster and Stronger. arXiv:2204.07366 (2022)

[26] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie. A ConvNet for the 2020s. arXiv:2201.03545 (2022)

[27] Ross Wightman, Hugo Touvron, Hervé Jégou. ResNet strikes back: An improved training procedure in timm. arXiv:2110.00476 (2021)

[28] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556v6 (2015)

[29] Luis Perez, Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv:1712.04621 (2017)

[30] Jean-Emmanuel Deschaud. KITTI-CARLA: a KITTI-like dataset generated by CARLA Simulator. arXiv:2109.00892 (2021)

[31] Cem Dilmegani. Synthetic Data to Improve Deep Learning Models. https://research.aimultiple.com/synthetic-data-for-deep-learning, last access: 29/05/2022.

[32] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, Sanja Fidler. DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort. arXiv:2104.06490 (2021).

[33] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks. arXiv:1406.2661 (2014).

[34] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Adela Barriuso, Sanja Fidler, Antonio Torralba. BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations. arXiv:2201.04684 (2022).

[35] Zong Fan, Varun Kelkar, Mark A. Anastasio, Hua Li. BigDatasetGAN: Application of DatasetGAN in medical imaging: preliminary studies. arXiv:2202.13463 (2022).

[36] Company DetectA. https://en.detectamiant.com/, last access: 28/08/2022

[37] Elad Hoffer, Nir Ailon. Deep metric learning using Triplet network. arXiv:1412.6622 (2018).

[38] Detectron2 Github repository. https://github.com/facebookresearch/detectron2 Last access: 07/08/2022

[39] mAP (mean Average Precision) might confuse you!. https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2 Last access: 07/08/2022

[40] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. arXiv:1404.5997 (2014).

[41] PyTorch Framework Web. https://pytorch.org/. Last access: 07/08/2022

[42] Elad Hoffer, Nir Ailon. Deep metric learning using Triplet network. arXiv:1412.6622 (2014)

[43] Support-vector machine Wikipedia. https://en.wikipedia.org/wiki/Support-vector_machine. Last access: 07/08/2022.

[44] Albumentations Framework. https://albumentations.ai/ Last access: 11/08/2022

[45] TorchMetrics Framework. https://torchmetrics.readthedocs.io/en/stable/ Last access: 15/08/2022

[46] Sanjeev Arora, Wei Hu, Pravesh K. Kothari. An Analysis of the t-SNE Algorithm for Data Visualization. arXiv:1803.01768 (2018)