POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Evaluation of Active Learning for Anomaly Detection in Images

Supervisors

Prof. Andrea CALIMERA

Prof. Valentino PELUSO

Candidate

Edoardo FANTOLINO

July 2022

Summary

During the last decade, the scientific research community has made astonishing steps in the development and improvement of algorithms that exploit huge amount of data with the aim of making machines perform tasks such as classification, object detection and semantic segmentation. Usually, the strategy used to train this models is the supervised-learning technique that requires labeled datasets.

In fact, the progresses made in the Machine Learning and Deep Learning field where enabled by a key factor: the presence of famous benchmark datasets already labeled.

Those famous dataset usually contains a large amout of data. The issue is that those dataset does not allow to create models useful for real application.

The manufactury industry trends is clearly in the direction of digitalization. In fact, many companies try to incorporate those powerful algorithms inside the production/manufacturing processes. The problem is that a strict bottleneck is present: the absence of taylored datasets. The activity of creating a dataset is resource consuming. Creating an annotated dataset present time and financial costs.

A solution is presented by Active Learning Strategies in which the main objective is to reduce as much as possible the burdain of the human driven annotation activity by selecting the most representative and useful data to train a model.

The literature, the research and the testing process of the Active Learning techniques was made exploiting the usual benchmark datasets, but in this work we present state of the art techniques of Active Learning applied to the Anomaly Detection field to see if those strategies are robust in more realistic and challenging context.

Moreover, due to inconsistent results and conclusion from the researchers community and due to the lack of presence of a stable methodology to quantify the improvement in the Active Learning field a stable framework is presented to enable results comparisons in a fair way.

Acknowledgements

A special thanks to my family that supported me during my university journey. I would like to thanks my supervisor and co-supervisor because they helped and encouraged me. They also provided useful inspirations that allowed me to achieve the final goal. I would also like to extend my gratitude to every person that kept me creative and I thank every person that nurtured my knowledge.

Table of Contents

List of Tables

List of Figures

Acronyms

AI
Artificial Intelligence

AD
Anomaly Detection

AL
Active Learning

ALP
Active Learning Process

DL
Deep Learning

GAN
Generative Adversarial Network

LR
Learning Rate

ML
Machine Learning

NN
Neural Network

REP
Random Extraction Process

SGD

Stochastic Gradient Descent

SC

Stopping Criteria

SP

Stabilizing Predictions

Chapter 1

Introduction

The digitalization phenomena, the improvements of electronic devices, the spread of smartphones and social media, the Internet of Things with millions of sensors all around the world allow us to collect incredibly large amount of data. The amount of data collected by those devices increase exponentially every year. Those data are precious and they are able to provide useful information in many different fields. Let think for example to: ecology, health, economy, physics, agriculture, manufacturing, marketing and many others.

The scientific research community has made progresses in the development of algorithms that exploits those large amount of data. In fact, in recent years, there were developed different strategies and algorithms in the field of Big Data. In particular, in the Computer Vision sector, are frequently used Deep Learning models. These models in order to perform tasks need to be trained. The training process of Deep Learning algorithms usually require dataset with thousand or even millions of samples. This is the reason why, Deep Learning models are know to be "data-hungry" algorithms. For this reason, in a normal training process, we tend to feed to the model as many data samples as possible. In standard/research situations, we have the possibility to do that because of the presence of huge already-labeled data-sets (e.g. CIFAR10, CIFAR100, ImageNet). Thanks to this labeled sets the research community has made significant progresses in finding the most suitable and efficient architectures. One first drawbacks is that, giving too much importance to the models, let us forget the key role that plays the training set. The presence of redundant or useless data samples could actually decrease the model performances following the principle: garbage-in garbage-out.

Moreover, The most used and effective training strategy is called supervised learning. In the supervised learning setting we do not only need samples but also labels. The labeling phase is a human driven activity where the annotator describe

the content/meaning/class of a given sample. This phase is resource consuming and it is a real bottleneck for the spread of Deep Learning algorithm in real world applications.

In order to avoid the labeling burdain, the research community tried to develop other training strategies such us the unsupervised learning. As the name suggests, in this learning context, the samples do not actually need the labels. This promising unsupervised field is challenging and interesting but still do not perform as the supervised learning. Another approach used to decrease the cost of the annotation phase is provided by the Active Learning scenario that not only reduce the labeling costs but also remove the redundant samples providing a better training set for our learner.

The main concept of Active Learning lies around an important balance: maximize the performances of a Deep Learning model with the smallest training size possible. This sentence goes against the usual belief that Deep Learning models are "data-hungry".

Another point in favour of the development of strategies that do not require large amount of data is that in real scenarios we need to create our own specific dataset, and while the data collection phase is usually quite cheap, the labeling phase in most cases become a bottleneck in terms of time and even actual economic cost. As an example, we can think about the impact that could have the labeling process of medical radiographies. This could include task as classification or even semantic segmentation where the time-economic cost is even higher. First of all we need professional with a certain degree of domain knowledge and this usually comes with an high economic cost. In second place, we need to take into account the time spent by this professional to label, detect or even segment this data samples. In this case, we do not only have to pay a human to perform a repetitive task, but we need to take into account the fact that we subtract a valuable resource (the doctor) to the medical field for different hours, and instead of saving life the doctor needs to label samples.

Another example are the manufacturing processes where from raw material we want to obtain as output a functioning mechanical structure. As we know, manufacturing processes are far from perfect and it could happen that instead of a clean output we receive a defect product. Sometimes, companies use human employees in order to perform quality checks. The task is to identify the defect products and discard them while keeping the products that are defect-free. This repetitive task can be automated by an algorithm able to autonomously perform this filtering task, while the employees can perform more complex and less repetitive tasks. In this cases, as mentioned before, the collection phase is quite cheap (taking picture of the output product) but we still have the bottleneck represented by the

cost of the labeling phase.

Even with the above-mentioned issues, we want to keep and exploit the feature extraction potential of Deep Learning models while reducing the need of huge human annotated data-set by inserting the training phase inside an iterative process that progressively select the most suitable data sample to feed to the learner.

We will achieve this objective taking inspiration from the state of the art techniques of Active Learning that combines concepts of self-supervised learning, uncertainty sampling, diversity sampling and many more interesting topics. Those techniques will be applied to a real application scenario: Anomaly Detection.

The Anomaly Detection problem merged with the Active Learning context presents 2 main challenges that combined are not addressed by the research community at least to the best of my knowledge. Those issues are:

- The samples defect and defect-free usually differ in minor details (e.g. holes, cracks) making more difficult to the feature extraction phase to separate the samples in distinct clusters.

- The extraction phase usually generate highly unbalance training set making the learning process more difficult and challenging

Moreover, some inconsistent and conflicting results are produced by the scientific research community due to the lack of a stable framework. The attempt to build a coherent platform to test and document the result and progresses is made.

The structure of the work is defined as following:

- **Background**: In this chapter is presented an high level explanation of the general Active Learning Process. It will allow the reader to get familiar with the topic.

- **Related Works**: In this part of the work are presented the most relevant and important works regarding AL scenario. The reader will find the papers that fueled and nurtured the development of those techniques. Moreover, there are also synthetically explained the new research direction that the scientific community is taking in order to make progresses.

- **Methodology**: In this chapter of the thesis it is reported a clear step-by-step guide of the precise AL methodology followed to obtain the results reported in the next chapter. It is also presented the description of the framework used to obtain fair and stable results. The final section address the problem of quintify the gain of the AL process.

- **Result**: In this chapter are introduced the real case scenario problem and the relative datasets with the corresponding result obtain using the strategy explained in the previous section.
- **Conclusion**: The final thought and over the work are presented in this brief final section

The take home message of this work can be nicely resumed by the following sentence: "get the most out of the least".

Chapter 2

Background

In this chapter is presented an introduction to the Anomaly Detection field and an high level description of the concepts and procedures needed to understand and perform an Active Learning process. The argument and explanation is self contained but the reader is supposed to know basic concepts of Machine Learning and Deep Learning.

Active Learning is an ensemble of strategies and techniques that putted together attempts to maximize the DL model performances while reducing the labeling effort required to the oracle/annotator.

This process gives more focus on the data that we will feed to our model instead of the model itself. In fact, we will see that the goal of this process is to remove the redundant training samples and select the most representative and diverse data that will allow our model to rapidly learn a given task. There are different flavours of Active Learning strategies and they could be divided into three main categories: membership query synthesis, query-based selective sampling and pool-based. Membership query synthesis is a shade of AL where the learner can request to query the label of any unlabeled data samples, even those generated by the learner itself. Stream-based selective sampling instead is more used in application where devices have storage and computational power constraint. The most widespread and common technique is the last one: pool-based sampling. Even if those technique are partially different, the main concept remains the same.

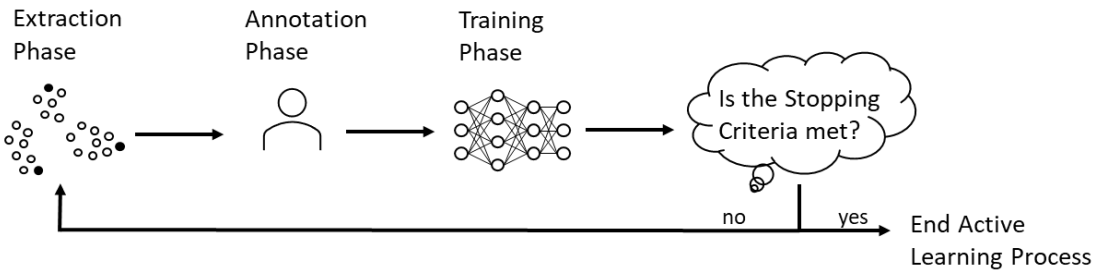

Figure 2.1: Active Learning Process

In Fig.2.1, there is the illustration of the Active Learning cycle.

As you can easily understand from the picture above, the overall process can be divided in 4 main points:

1. Extraction Phase

2. Annotation Phase

3. Training Phase

4. Stopping Criteria

We start with a completely unlabeled set. The first step of our process is to extract a given number of training samples from the unlabeled set. Then, we will annotate the selected samples. After the annotation phase, we add our new labeled data to our labeled training set. Then, we will actually train our model over the labeled training set. After the training, we will evaluate the performances of our model and eventually re-start the process by adding to our labeled training set a new batch of unlabeled data extracted from the unlabeled set. The Active Learning Process will continue till a certain Stopping Criteria is met. This could mean that we have finished our budget (e.g. the budget could be represented by an economic or time constraint) or the model shows us to have reached satisfactory performances.

2.1 The Active Learning Process

The Extraction Phase

1. the **Extraction Phase** is crucial for accomplishing our goal of reducing the number of total annotations. Precisely, it consists in carefully select the most useful data out of our unlabeled set. The most useful data are identified by a measure of diversity and "difficulty to be learnt". In order to quantify this concepts, we first exploit state of the art techniques in the field of Self-Supervised Learning (it is important to underline that in this phase we do NOT require any labels) in order to extract high-level features from our samples. Now, thanks to the extrapolation of this features we can map our points in a features space and compute distances and similarity/diversity measures between data-points. We can also apply clustering techniques in order to identify different groups in our unlabeled set. Moreover, with dimensionality reduction techniques such as PCA, UMAP and others we can have a look to the visual representation of our points in the feature space in two or three dimensions. In Fig.2.2 , you can see an example of visual representation in two dimensions of the feature space obtained exploiting Self-Supervised Learning technique and UMAP dimensionality reduction.

Figure 2.2: Illustration of the Embeddings of Medical-MNIST dataset.

In a practical context, after the features extraction phase, in the first Active Learning step, we do not have already trained our learner.

So, the extraction technique could be to naively select at random a given number of samples or alternatively it is possible to directly apply more sophisticated strategies such as coreset-sampling technique that keeps into consideration the samples diversity.

A comparison between random sampling technique and coreset-sampling technique is provide in Fig.2.3.

In Fig.2.3 there are represented 2 simulated clusters computed with a Normal Distribution. This distributions have the same standard deviation and different

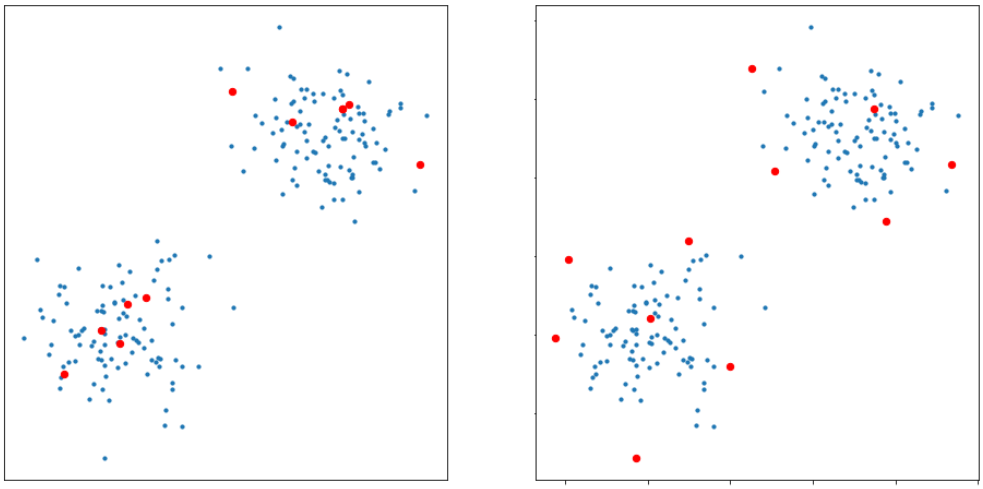

Figure 2.3: Random Extraction vs. Diversity Sampling: In those picture is represented a situation with 2 clusters. The blue point are the datasamples and the red point are the selected samples. On the left, there is the example of a random extraction process while on the right there is the representation of a coreset-selection process. Coreset-Selection tends to maximize the diversity of the samples.

mean. As we can understand from the picture, the coreset-selection technique maximize the diversity of the samples while the random technique concentrate the selected samples near the mean and the center of the cluster. The dataset obtained by random technique could consequently contains redundant images.

Then, from the second Active Learning step (and beyond) we can apply more sophisticated hybrid techniques that merge diversity and uncertainty measures. The uncertainty sampling allows us to extract more samples where the prediction over that given sample is not certain.

In Fig.2.4 it is show an hybrid techniques that merges the benefits of diversity and uncertainty sampling. In the pictures there are three distinguishable clusters. The red squares represents the point extracted with a coreset-sampling technique. Moreover, by exploiting the predictions over the unlabeled set, we understand that the model is certain regarding the predictions over the samples belonging to the cluster on the right, while it is uncertain for what concern the bottom-left and upper-left clusters. Given that the model is uncertain with the clusters on the left, the algorithm extracts also the data-points underlined by the orange-diamond. In this way, our training set will be populated by diverse sample and by the samples that are more difficult to be learned.

It is important to underline that measuring the uncertainty of a model is a known problem. The issue of measuring and quantifying the uncertainty of a

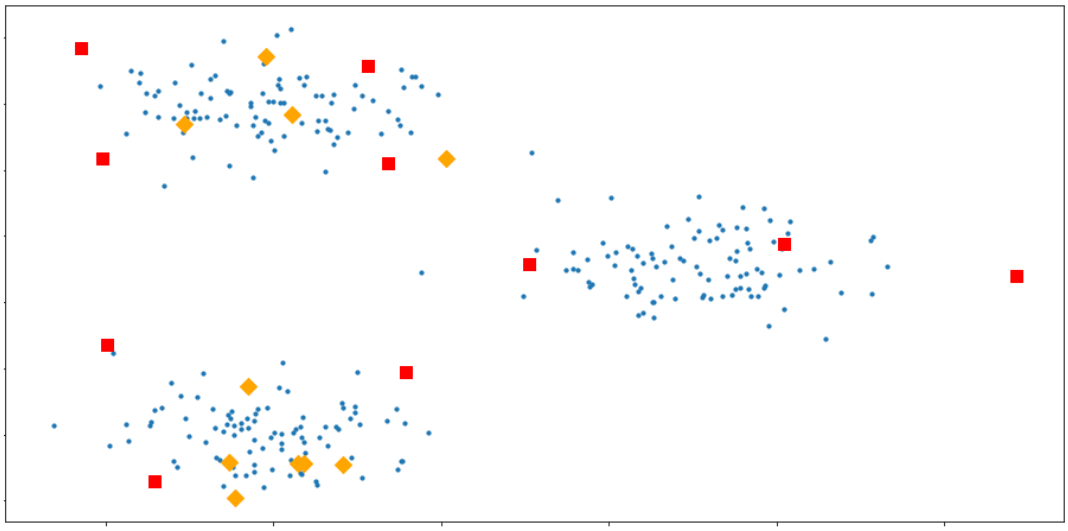

Figure 2.4: Hybrid technique that merges Uncertainty and Diversity sampling. The blue points are the samples represented in the feature space. The red squares are the sample selected thanks to the coreset-sampling technique, while the orange-diamonds are the samples extracted by the uncertainty measure.

prediction of a model is addressed with different strategies. The most popular techniques used are the following:

$$Least\ Confidence = \frac{n(1 - max(p))}{n - 1}$$

where p represent the prediction of a model over a given sample and n is the total number of classes. The Least Confidence measures the difference between the most confident prediction and the complete certainty (100%).

$$Margin\ of\ Confidence = 1 - (max(p) - max(p_{-}))$$

where p_{-} are the same probabilities of p but it is excluded the largest one, more formally $p_{-} = p \backslash max(p)$. The Margin of Confidence measures the difference between the top two most confident predictions.

$$Ration\ of\ Confidence = \frac{max(p_{-})}{max(p)}$$

The Ratio of Confidence represents the ratio between the top two most confident predictions.

And finally the entropy exploit a measure defined by information theory:

$$Entropy = \frac{-\sum_{i=0}^{n} p_i log_2(p_i)}{log_2(n)}$$

After the extraction phase, the human driven intervention is needed.

The Annotation Phase

2. The **Annotation Phase** is the step of the process that actually requires human intervention. In this subpart of the Active Learning step an oracle is asked to label the samples selected from the algorithm during the previous extraction phase. In the labeling phase, the algorithm requests to the annotator to classify the selected samples. The same approach can be obviously used for tasks such as object detection with bounding boxes or semantic segmentation with masks. This phase could be simplified and supported by the usage of modern tools that try to speed up and organize the labeling process. In fact, multiple modern software manages to reduce the burdain of the labeling process by creating friendly user interfaces. The cost of this phase is one of the reasons for all of the research effort in the Active Learning domain. This because, in some situations, the labeling phase could actually represent a strict bottleneck to the learning process.

The Training Phase

3. During the **Training Phase** we can actually train the Deep Learning model in a traditional supervised manner by feeding to it the data selected and labeled during the previous phases. The training process could be defined with a fixed number of iteration per each Active Learning step or it is also possible to fix the number of epochs making the number of iteration proportional to the size of the training-set of the current Active Learning step. It is a delicate part of the process since we have to consider all the important parameters of the traditional training process but also the new parameters of the Active Learning scenario.

The Stopping Criteria

4. The **Stopping Criteria** is the moment in which the algorithm decides if it is time to stop the Learning Process or if it is required to perform another AL cycle. In a real application context it is a crucial phase in order to obtain a model with the desired performances.

2.2 Anomaly Detection Context

Anomaly Detection is the identification of samples that do not show to have the behaviour/aspect of normal data.

Anomaly Detection can be performed by different types of Algorithms of ML. Sometimes Anomaly Detection is also identified as Outlier Detection because these techniques tends to identify the samples that have different characteristics with respect to the usual data.

The field of application of Anomaly Detection are:

- Network Intrusion Detection: Nowadays, network systems suffer from security breach and hacker attacks. Modern IT systems are able to collect the data exchanged between different sources and with algorithm of Machine Learning are able to understand if the activity between them is an usual user activity or if it is something anomalous that could possibly damage the infrastructure or the user experience.

- Medial Diagnosis: In this technological era, we try to exploit the new powerful algorithms in many different areas. In fact, the evolution in some medial field is driven by MLDL technologies. Those techniques are used in order to analyze data such as X-ray, Cardiotocography (fetal heartbeat), MRI or ECG. Anomaly Detection technique could be applied and are applied in this fields in order to find patients with possible diseases

- Fraud Detection: As the usage of online transaction increase day by day, so do the fraudolent transactions. A possible advantage is to exploit those AD technique also in the financial market. The algorithms are trained to detect abnormal transaction and individuate the problem. Thanks to the abundance of data, these algorithms are able to perform well in the detection of Anomaly and fraudolent Transactions.

- Manufacturing Defect Detection: In the manufacturing context, those algorithm are used in order to automate repetitive tasks. Usually, the enterprise is able to collect large amount of data thanks to cameras or microphones. Then, after the data collection phase, the samples are labeled and an algorithm is trained over them in order to perform a binary classification task: defect and defect-free. If a manufacturing product is predicted as a defect, it will be discarded. Given that this operation is performed by machines, the economical cost of the company could decrease.

Deep Learning is usually introduced in the medical and manufacturing field because of its well known ability to extrapolate high level features. Thank to the

Deep Learning applied to the Anomaly detection field we were able to improve in different context and automate repetitive task.

But some problems are always behind the corner to increase the difficulty to implement efficient and performant algorithms. Given that the DL algorithms are known to be data-hungry, a huge amount of data is required and sometimes just producing data could have significant costs (e.g X-Ray). Then, even if the data collection phase is cheap, the cost could arise from the labeling phase.

The Active Learning Strategy presented and tested in this work attempts to provide a solution for both the aforementioned issues. In fact, we want to test the efficiency and robustness of the state of the art technique in AL applied to AD. The combination of this two elements generate different challenges. First of all, in the pipeline adopted it is performed a Self-Supervised Learning step in order to extract feature from the samples that will be useful in the extraction phase. In a normal classification context, the feature extraction phase provide well separated clusters because the difference between images belonging to different class are quite evident (car, dog, car, boat) while in the AD field the difference are provided by minor details such as holes and cuts. An example of feature clusters in a classification context is provided in Fig.2.2 where each cluster represent a separete class. An example of binary classification in AD context is provided in Fig.2.5 and Fig.2.6

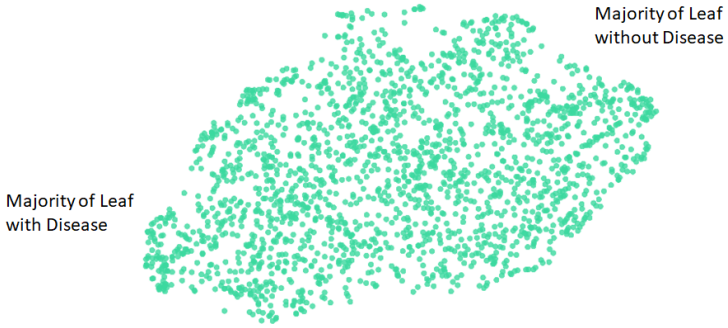

Figure 2.5: Single Cluster: Illustration of the Embeddings of Potato Leaf Blight Disease dataset. In this Situation is present a single cluster. The majority of leaf with disease are situated in the bottom left while the majority of healthy leaf are in the top up part of the cluster.

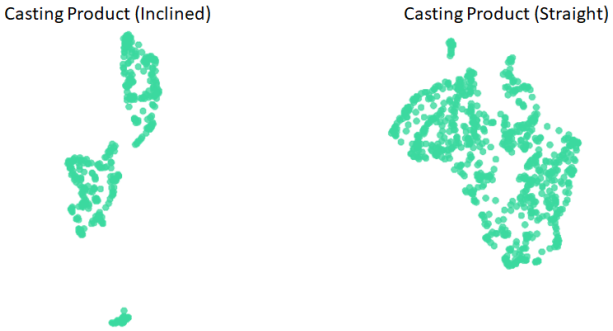

Figure 2.6: Misleading Clusters: Illustration of the Embeddings of Casting Manufacturing Product dataset. In this situation the clusters are misleading because the cluster are separated in function of the orientation of the manufacturing product and by the light conditions. While the defect and defect-free are merged inside both clusters.

The other issues that we will test, it the robustness of the algorithm when those technique are applied to imbalanced unlabeled set. This is always due to the AD context because usually the class of defect and defect-free are quite imbalanced. The imbalance nature of the unlabeled set is reflected in the selected samples and so it is a problem that affects the efficiency of the Active Learning process.

Chapter 3

Related Works

In this chapter of the thesis are presented the main works related to the AL context. AL is a promising field. The research community try to improve these technique every years in different aspects by introducing in the Active Learning cycles new technologies such as Generative Adversarial Networks (GANs) or even Self-Supervised Learning. Moreover, some researchers started to apply AL technique in the field of the interesting Edge Computing context. Other researchers instead try to create AL patterns and techniques that are agnostic with respect to the task that the model will perform. All of these results and researches let us understand that the AL field is flourishing but the best performances are not already reached.

3.1 Dataset Redundancies

The success of Deep Learning models is strictly related to the rise of large annotated datasets. The common belief is the following: if we feed more sample to our model during the training phase we will increase the performances of the model.

So, with this stable assumption, for many years the attention of the researchers was mainly focused towards the model architectures and related subjects. In fact, while researchers where discovering the powerful and effective skip connections of the ResNet model [**10**], we where shadowing the potential benefit of understanding the training data.

One of the recent works that changed the trend and successfully demonstrated that not all data are actually useful and that in many cases there is the presence of redundant samples in different benchmark dataset is represented by [**1**]. In fact, in this paper it is shown that it is possible to find subsets of the overall training set such that the model, if trained on this reduced subset, will have the same performances or even better performances with respect to a model trained on the full training dataset.

Thanks to the technique used by the researchers of the papers, they were able to find at least a 10% of redundant samples in dataset such as CIFAR10 and ImageNet. It is worth mentioning that training a model without the redundant samples seems to effectively improve the performance of the model trained on 100% of the training set. However, their contribution is to demonstrate the existence of redundancies and they do not claim any algorithmic contribution. The main concept presented by the authors is that they explicitly looked at a dissimilarity measure between samples. They explored the feature space of a pre-trained model trained on the full dataset (as in many other related works, the feature space is obtained from the penultimate layer of the Neural Network).

To find redundant samples they exploit Agglomerative Clustering [**6**] applied to the previously obtained feature space.

The dissimilarity of two samples is computed using the cosine angle. For example, given two images S_1 and S_2 whose features coordinates are x_1 and x_2 respectively, the dissimilarity is computed as follow:

$$d(x_1, x_2) = 1 - \frac{\langle x_1, x_2 \rangle}{\|x_1\| \|x_2\|}$$

while the dissimilarity between cluster C_1 and C_2 is:

$$D(C_1, C_2) = \max_{x_1 \in C_1,\, x_2 \in C_2} d(x_1, x_2)$$

That represent the maximum dissimilarity between any two of the point belonging to the respective clusters. Then, points inside the same cluster are considered redundant and in each cluster they selected the datapoint nearest to the cluster centroid.

3.2 From Active ML to Active DL

Another important point to underline is that the firsts Active Learning techniques were developed and tailored for Machine Learning algorithms.

In this early development of AL strategies, if we take into consideration algorithms such as Support Vector Machines, the most common approach for selecting the data between different steps of Active Learning was to choose the samples that where near the decision boudaries. An effective improvement was done in [**7**] where it is show that better performance could be achieved by taking into account the prior data distribution. In fact, selecting only data near the boudaries could produce redundant training sets as pointed out in [**1**]. With the work presented in [**7**], the researchers, exploiting pre-clustering technique are able to avoid to repeatedly label samples belonging to the same clusters.

Moreover, in the Machine Learning context, the size of the training set was incremented by one element at each Active Learning step. This because usually training a ML algorithm is less expensive than training a DL algorithm.
Incrementing the size by 1 element at each AL step represents a problem for 2 main reasons:

- training a Deep Learning model is computationally cumbersome and time consuming
- adding just one element to the current training set would probably not result in a significant improvement in the model performances

Hence, in [**2**], it is pointed out that is necessary to change strategy. In fact, they decided to query labels for a larger subset of samples at each Active Learning step instead of just 1.

The researchers of the paper propose a Core-set selection approach that aims to find a small subset (size>1) to be added to the training set such that the learner trained on that small subset will be competitive with respect to the model trained on the entire dataset.

Moreover, one of the main result of this work is that it is shown how the minimization of the core-set selection problem is equivalent to the K-Center problem. The K-Center problem is NP-Hard and so the authors developed the following faster greedy algorithm:

Algorithm 1 K-Center-Greedy presented in [**2**]

Require: data x_i, existing pool s^0 and budget b
Ensure: $b > 0$, $|s^0| > 0$

$s = s^0$
while $|s| = b + |s^0|$ **do**
    $u = \arg\max_{i \in [n] \setminus s} \min_{j \in s} \Delta(x_i, x_j)$
    $s = s \cup \{u\}$
end while
return $s \setminus s^0$

3.3 Task Agnostic methods

One of the next issue addressed by the research community is the fact that most Active Learning strategies are computationally inefficient for large networks or are carefully tailored for a specific task. In [**3**], the researchers propose a novel Active Learning method that is easy to deploy, task-agnostic and also efficient in the Deep Learning context. The main idea is to modify the learning model by attaching to it the so called "loss prediction module". As the name suggests, the loss prediction module will learn to predict the loss of a given sample. The capability of predicting the loss of a given sample will be deployed over the unlabeled set to create a list. This list will contains, in decreasing order, the samples with the corresponding predicted loss. The Active Learning Algorithm will ask to the oracle to label the Top-K samples of this list with the largest predicted loss.

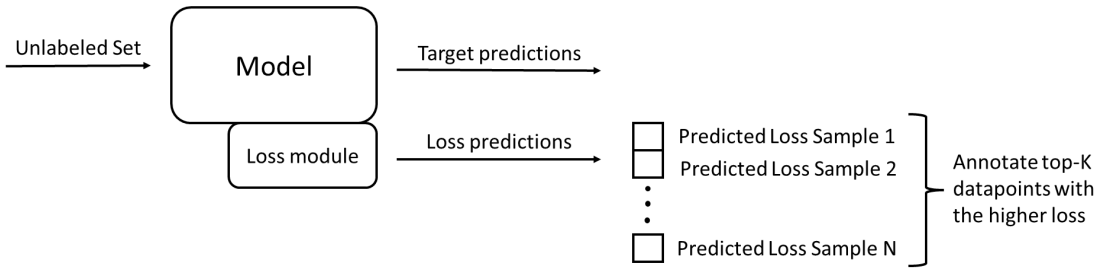

Figure 3.1: The concept of Learning Loss. Concurrently with the actual model, it is also trained a new learning loss module. This module will be used to predict the loss over the unlabeled set. The samples with the highest predicted loss are the one that will be labeled by the annotator.

3.4 Self-Supervised Learning

Even if Fully Supervised Learning is leading the learning techniques in the field of DL algorithms, new strategies are explored and developed. The opposite of Supervised Learning is Unsupervised Learning. In fact, we go from completely labeled dataset to completely unlabeled dataset where no labels are required.

This type of technology are not mature enough and some intermediate techniques are studied and improved. An intermediate solution is represented by Self-Supervised Learning.

In the context of Active Learning, for the extraction phase we need a feature representation of our dataset. Obviously, at this stage of the AL process we do not have any labels and for this reason Self Supervised Learning is exploited in order to compute and create the feature representation of our samples. An example of this technique is provided in [**5**] where the authors propose a simple framework for Contrastive Learning. The proposed method is called SimCLR.

The main concept is that SimCLR learns the representations by maximizing agreements between the losses in the feature space of the same image augmented with two different techniques (e.g: cropping and padding, padding and cutting, rotating-cropping, reflecting-jittering).

The general framework of Self-Supervised SimCLR is composed by four main components:

- A data augmentation module that is able to create two different augmented representation of a given data sample

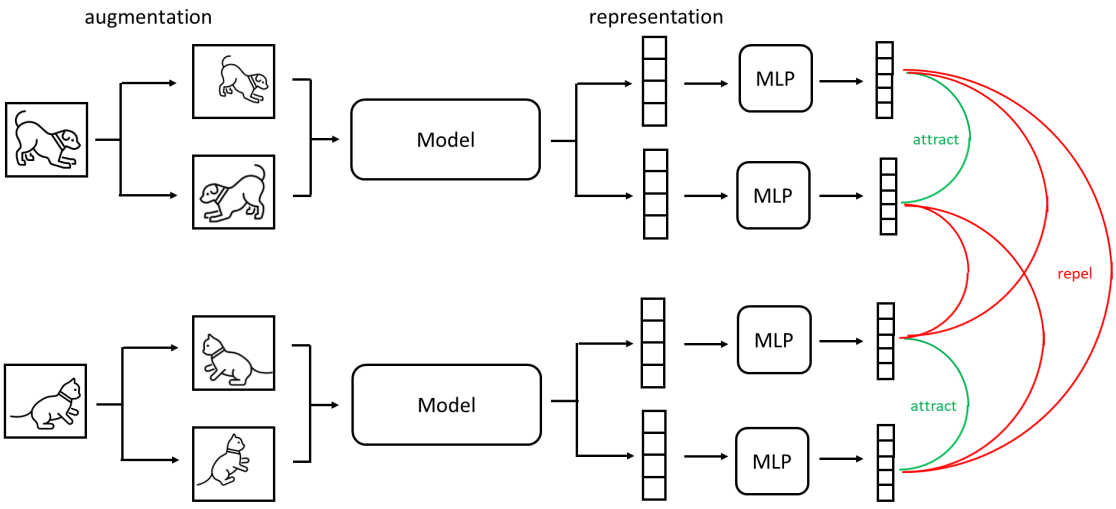

Figure 3.2: The concept of Self Supervised Learning.

- A Neural Network base encoder that is able to extract features from the previously augmented data samples
- A smaller Neural Network projection head that maps the features to the space where the contrastive loss will be applied
- A contrastive loss function defined for a contrastive prediction task

During training, a random minibatch of images is selected and a contrastive prediction task is define on the pairs of augmented data. The final loss is computed across all positive pairs.

Promising result in the field of Self-Supervide Learning are provided by recent works such as [**8**, **9**] where it is shown that in some conditions Neural Networks are more robust to data shifts when pretrainet on a set of images without supervision.

3.5 Stopping Criteria

Even if the extraction phase and the training phase are efficient and optimized, they could not be sufficient to obtain satisfactory performances. In fact, another crucial key aspect in real application is represented by the Stopping Criteria. The Stopping Criteria is the strategy that allows our algorithm to decide if continue the Active Learning process or if it is time to stop and do not ask for more labels to the oracle.

The most used techniques to stop an AL process include the maximum number of iterations, the exhaustion of the labeling budget (reached maximum number of possible annotations) or the expected accuracy value.

The Stopping Criteria seems to be a simple choice but an early stopping could lead to poor performances and an excessive annotation is obviously a waste of resources.

A new trend called Stabilizing Predictions is emerging. One of the most important framework of Stabilizing Prediction is presented in [**8**]. As the name suggests, it take into consideration the predictions of the model over a predetermined set of samples called: stop set (this set of samples does not require to be labeled). If the variation of the predictions over the stop set does not exceed a given threshold, the model is supposed to be "well-trained" and the Active Learning process reaches the end.

3.6 Data Expansion Strategies

The Active Learning potential is appreciated when the amount of data to be labeled is as low as possible. General technique adopted when there are only few data are for example the data augmentation strategies such as: cropping, rotating, jittering. Those samples transformations synthetically increase the number of samples without adding annotation costs. Other more sophisticated techniques are used in the Deep Learning Traditional training and now many researchers try to implement them also for the Active Learning scenario.

An example is provided by [**10**] where the researchers build a framework called Cost-Effective Active Learning. The main idea behind this framework is that the training set will be enriched by the model itself. In fact, the model is allowed to assign pseudo-labels to the sample where the prediction have an high confidence. This enriched set with labels provided by humans and labels provided by the algorithm itself will be used to re-train the learner.

Another example of data expansion is to introduce inside the AL process the Generative Adversarial Networks (GANs) data augmentation strategies. The aim of GANs applied to AL is to increase the learning speed and efficiency. In fact, with those technique the main objective is to generate training samples that increase the information contained in the original dataset. An example that follows this process is provided by [**13**] where the technique is tested over MNIST dataset.

3.7 The Edge Computing Challenge

The design of this Deep Learning techniques usually does not take into consideration the applications and limitations of edge computing. In the edge computing context to the usual Deep Learning issues we have to take care of constraints such as:

- restricted storage
- latency constraints
- limited computational power

This is a problem addressed by [**4**] where it is pointed out that Active Learning strategies could actually be used to reduce the amount of data that needs to be transferred from the edge to the cloud.

Their proposed approach is based on a dynamic Bernoulli process to select the most appropriate method for extracting the samples based on the model's performances. The general framework is presented in Fig.3.3.

In this framework, the sensor collects a certain amount of data (e.g. images or audio). Then, techniques of Unsupervised Clustering and Outlier Detection are

combined to select diverse data. Those technique must be as lightweight as possible in order to satisfy the storage limitation provided by the edge. Next, in function of a Bernoulli extraction with parameter λ it is decided if introduce the uncertainty measure or not. More precisely, when the model is mature or it is not trained at all, this process will more likely not take into account the model uncertainty during the extraction because of the Bernoulli process with parameter λ. This λ parameter is adjusted accordingly to the performance shown by the model after the training phase in the cloud (if the model improve λ is decreased and viceversa).

This work let us realize that it is important to point out the need to improve AL strategies for what concern the edge context because it is an open issue and it is difficult to address.

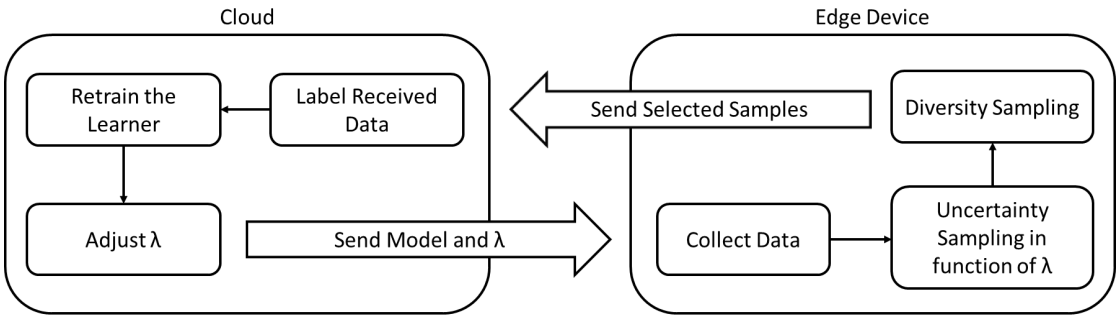

Figure 3.3: The general framework of Active Learning Edge Computing. On the left there is the Cloud where the model is (re-)trained and the λ parameter is updated in function of the model performances. Instead, on the right there is the Edge Device where new data are collected and sampled on the sensor itself in order to minimize the amount of data exchanged between Cloud and Edge.

3.8 Coherency in the Active Learning Context

As the related works demonstrate, the effort towards the improvement of Active Learning Strategies is increasing year by year. New techniques are integrated inside the AL context and researchers claims to constantly improve the performances of the final learner.

A significant issue is underlined in [**14**] where it is noticed that under the same initial settings it happens that different papers report different results and arrive at different conclusions. As an example, with the same experimental setup, the performance reported by [**15**] using $\frac{1}{5}$ of the total size of CIFAR10 are 13% lower

than the performance reported in [**3**]. Another example of result inconsistency is provided by [**2**, **16**] where the difference in the result is up to 8%. They both used 40% of the samples of CIFAR100 and VGG16 as Neural Network but the results are obviously inconsistent.

Moreover, due to this results variance, sometimes the research community tends to extrapolate from the experiments some conflicting conclusions. In fact, [**2**, **17**] underline that Diversity Sampling techniques are proved to be better with respect to Uncertainty Sampling techniques. However, in other related works the conclusion is exactly the opposite as shown in [**3**].

All of this observations underline the urgent need for a complete, coherent and comprehensive framework to test AL strategies. This framework should allow the research community to quantify and identify the progresses made in this new challenging context.

Chapter 4

Methodology

In this part of the thesis it is provided a detailed description of the overall Active Learning Process exploited to obtain the result shown in the next chapter.

It is important to underline that the main goal of the Active Learning process is to reduce as much as possible the size of the training set while maximizing the performances of the model.

We obtain this objective thanks to an iterative process.

The input of the Active Learning Process is a set of unlabeled samples and the output is a Deep Learning model that satisfy certain conditions of performances. The core concept of Active Learning still remains the selection of the best samples to feed to a Neural Network. In our case, the best sample are: the most diverse data and the hardest data to be learned by the model.

At each step of this iterative process, we extract from the unlabeled set a given number of samples that will be labeled. This given number of samples is called: budget. So, at each AL step we increase the size of our training set by adding a subset of our unlabeled set of size equal to the budget.

In Fig.4.1 it is shown the complete process of the AL strategy presented in this work.

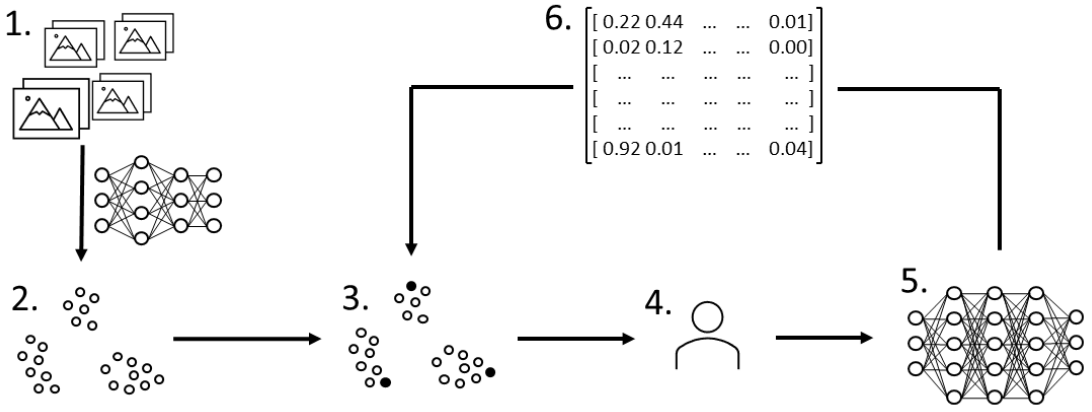

Figure 4.1: Complete and detailed illustration of the Active Learning process exploited in this work.

1. Unlabeled set of samples
2. Feature space of the samples
3. Selected samples to be labeled
4. Oracle/Human
5. Deep Learning Model to be (re-)trained
6. Predictions (NB: uncertainty method only)

4.1 Iterative Process Details

In this section is described step-by-step the adopted methodology used to obtain the result shown in the next chapter.

1. As aforementioned, the input of the Active Learning Process is a set of unlabeled samples (raw data, images, text, audio etc..).

2. Then, the iterative process starts with a Feature Extraction phase that will be performed only once and never again. To extract the features we use a standard

pre-trained model that will be adapted on our specific use case (this model is different from the final learner and it is used only once and never again). This pre-trained model has an architecture similar to ResNet. The output size of this architecture is 31. So, the feature vector will have 31 dimensions. Higher outputs introduce various issues including the curse of dimensionality. The result of this step are the embeddings, that corresponds to a matrix $M^{n \times d}$ where n is the number of samples and d is the dimension of the final layer of our ResNet like architecture.

NB: Point 1. and 2. are executed only once. For 2. we work on unlabeled data with self-supervision techniques. NO labels are required to execute this part of the Active Learning cycle.

3. This is the crucial phase of the Active Learning Process: *the extraction phase*. We need to take into consideration 2 main cases: the first Active Learning step and the steps after the first.

- (Arrow from 2. to 3.) In the first step of Active Learning we still do not have trained our final learner (the final learner is still randomly initialized). So, to extract the samples for the first step, we used Diversity CORESET Sampling techniques and obviously NOT Uncertainty Sampling techniques. For this reason, in the first AL step, we base our sample extraction on the CORESET algorithm that geometrical measure the diversity/similarity of our unlabeled samples. Thanks to these measures, we can extract the samples that are the most diverse between them, hence the most representative data.

- (Arrow from 6. to 3.) From the second step on, we actually have a trained learner and for this reason we can merge the benefits of Diversity Sampling and Uncertainty Sampling. In this case, we build a specific ranking that will basically measure the utility/benefit of labeling a given sample with respect to the others. This ranking is build upon a graph representation of our data points. The geometrical representation is build thanks to the CORESET sampling technique and moreover we merge this measure with an uncertainty value obtained by exploit the prediction over the whole unlabeled set. More precisely, the uncertainty of the prediction over a single sample is deduced exploiting an entropy measure defined as follow:

$$entropy = \frac{-\sum_{i=0}^{n} p_i log_2(p_i)}{log_2(p_i)}$$

  where n is the number of classes and p_i are the values of the prediction concerning a given class i. We balance the magnitude of the Uncertainty measure and Diversity measure thanks to a λ parameter. We build the ranking. Then, we select the samples in the ranking in descending order till we exhaust our budget.

The output of this phase is a partition of the unlabeled set that should be labeled and added to the training set that will be used to train our final learner.

4. The bottleneck of the process: a human is required to label the previously selected samples.

NB: Actually, in a real process this phase consumes a lot of time resources. In this experimental context we exploited already-labeled datasets. So, the developed framework was design to speed up point 4 of the ALcycle. The framework was built such that the behaviour and results obtained are completely unaltered with respect to a real application of AL strategies to an unlabeled dataset.

5. We train our final learner in a traditional way exploiting the samples selected and labeled during the previous steps. In particular, we used a model composed by 2 convolutional layers and 3 fully connected layers. The hyper-parameters of the training process varies in function of the use case.

The Learning Rate (LR) decay inside a AL step is defined by the Polynomial Decay Schedule of degree 2. The starting and ending value of the LR at each AL step is 0.001 and 0.0001 respectively. The optimized adopted in all the experiment is the Stochastic Gradient Descent (SGD). The batch size was set to 15 (this is an unusual value for the batch size but allowed us to better monitor the progresses with respect to other AL parameters such as the budget and the number of iterations per each AL step).

During the training phase, we exploited Data Augmentation strategy provided by PyTorch such as:

- Random Horizontal Flip: default parameter
- Random Vertical Flip: default parameter
- Random Rotation: 0.05
- Resize: the resize parameter were in function of the specific experiment e.g. 64×64, 256×256, 300×300
- Normalization: the mean and standard deviation for the normalization were computed on the total unlabeled set

One of the important aspects to take into consideration was the fact the most of the time the obtained labeled set used for training was highly unbalanced. In fact, a given class usually had a much higher frequency with respect to the other(s). In order to cure this issue, the algorithm exploited a balancing-technique called: weight random sampling. Basically, to each sample is assigned a weight in function of the class in which it belongs. This weight will be higher for the minority class

and lower for the majority class. This computed weight will be used in the PyTorch Dataloader in order to balance the mini-batch during the training phase. In the training phase, the model will see all the classes with the same frequency inside each mini-batch.

From the different strategies of Stopping Criteria, the algorithm used the maximum number of iterations. So, if the pre-selected number of iterations was reached, the algorithm stopped otherwise it continues the iterative AL process eventually going toward point 6. and in 3. again.

6. When using an hybrid technique we need to compute the prediction and consequently the entropy measure of our final learner over all the unlabeled samples in order to exploit Uncertainty Sampling techniques. The output of this process is basically a matrix $M^{n \times m}$ where n is the number of total unlabeled samples and m in the number of classes.

Thanks to this measures the algorithm is able to create a graph representation of our embeddings and translate it into an importance ranking by taking into consideration also the entropy measures.
The process return to point 3.

4.2 Building a Stable and Consistent Framework

Given that the research community lacks of a coherent and stable framework to determine the improvements, we developed a consistent infrastructure from scratch.

In fact, in the methods exploited by research community is not so clear if the performance improvement of the learner are mostly provided by the increasing amount of weights updates that the models do in the training phase of each AL step or if the benefit is actually related to the selection of the best samples.

Our coherent and consistent framework is presented in the following:

(It is possible to use this framework because we are in a research/experimental scenario. In fact, we start with a complete labeled set and we synthetically create an unlabeled set to simulate an AL setting. Not applicable in real applications.)

1. Given that we have the complete labeled set, we are able to draw a baseline. This baseline will represent the maximum performance that we can obtain from that model and that complete labeled training set. To obtain the baseline, we use a simple and traditional training process taking into consideration the epochs of the training. After the traditional training is completed, we will obtain a certain metric (e.g: accuracy, F1 score) with a certain value that will represents the aforementioned baseline.

2. From the number of epochs we need to compute the total number of

iterations in which the model went through answering the question "How many weights updates has made the model during the traditional training"?

$$total\ number\ of\ iterations = \frac{total\ number\ of\ samples}{batch\ size} \times epochs$$

3. Fix the total number of Active Learning steps that the process will do before stopping (e.g: 10, 20).

4 Fix the budget and perform Active Learning Process

4.1. Compute the fixed budget as follow:

$$budget = \frac{size\ complete\ dataset}{number\ of\ AL\ steps}$$

In this way, we will perform the final step of the AL process with the full dataset (same training set as the baseline).

4.2. Push the budget towards lower values as follow:

$$budget = \frac{previous\ budget}{2}$$

4.3. If the model has reached the baseline value, keep reducing the budget with the formula in point 4.2, otherwise if the baseline value is not reached, increase the budget as follow:

$$budget = previous\ budget + \frac{previous\ budget}{2}$$

5 repeat point 4.1, 4.2 and 4.3 with a Random Extraction Process

Point 4 and point 5 could have as many sub-point as needed. This framework allow us to define four major aspects:

- Compare the advantages of having a larger or lower budget in between different step of AL
- Quantify the gain of a taylored selection strategy w.r.t a random extraction process.

In our work for the final result, a parametric study was performed analyzing the **number of AL step**. The baseline value was draw exploiting 10 AL steps and then the number of AL step was gradually reduced to understand what was the minimum number of annotation needed to reach the baseline in terms of accuracy. Special care was take to keep the training cost equal for all the experiments.

4.3 Quantify the Gains

In the challenging AL context it is also complex to define the actual gain obtained by exploiting AL strategies with respect to a Traditional Training Technique. But more importantly is difficult to determinate if one AL strategies is better with respect to another one. This is due to the fact that measuring the performance af the AL process could vary in with respect to the needs of the user and the chosen Stopping Criteria. Obviously by increasing the number of annotation we will usually improve model performance, but we will increase the costs. The performances should balance this pros and cons.

As underlined before, the main aspect that AL address is to reduce the burdain of the labeling phase. With AL strategies we want to reduce the time and economical cost dedicated to the annotation task.

For those reason a simple, obvious and quite used measure to quantify the gain provided by AL is to compare the accuracy/f1_score of two learning methods in function of the number of annotations. Those technique will be used in the Result chapter to effectively quantify the gains. An example of this gain measurement technique is provided in Fig.4.2 we can see an example of considering Annotation as the parameter to quantify the gain between 2 different strategies. If we suppose that our maximum budget is equal to 80 images, then strategy 1 would perform better with respect to strategy 2. Instead, if our annotation budget could reach the 150 samples the strategy to prefer would be strategy 2 because the final model would have 10% more of accuracy.

Lets suppose instead that our objective is to reach as soon as possible a given threshold without considering at first the number of annotations. This situation is represented in Fig.4.3

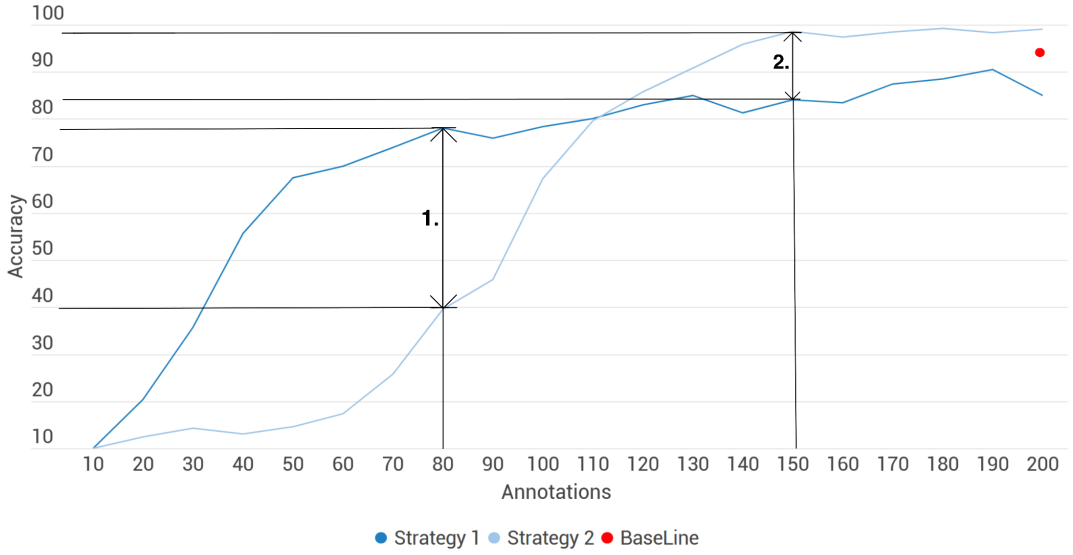

Figure 4.2: Annotation Method to quantify the gain provided by AL strategies. 1. shows that when we have a total of 80 annotated samples, the strategy 1 is better than strategy 2 by a margin of almost 40%. 2. show that with 150 samples, strategy 2 is better than strategy 1 by a margin of more than 10%.

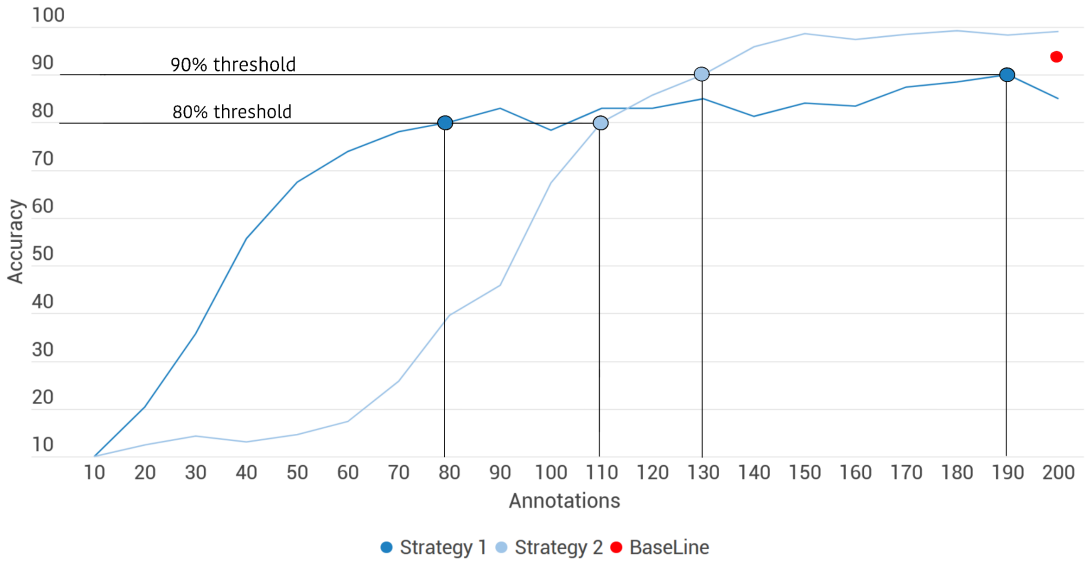

Figure 4.3: Threshold Method to quantify the gain provided by AL strategies. In this case, the objective is to hit a certain threshold of accuracy as soon as possible in terms of annotations. The 80% threshold is hitted before by strategy 1 with a gain of 30 annotations, while 90% threshold is hitted first by strategy 2 and then by strategy 1 with a gain of 60 annotations.

In our work, the advantages or disadvantages of applying an AL technique to train a model is quantified by comparing the performances of a model trained exploiting AL stratetegies with modern extraction technique w.r.t a model trained with an iterative random selection process. The comparison of these to values allow us to determine the gain or disadvantages of the application of AL strategies.

Chapter 5

Results

In the most of the research literature, the Active Learning Process was tested and developed with famous benchmark datasets. The advantages of applying AL techniques in this standard situations is evident. The problem is that most of the time those dataset do not represent real situations.

We want to go beyond the usual research context and our main objective is to demonstrate the effective potential and benefit of Active Learning strategies for real applications.

This is the reason why we decided to apply the AL process in the case of Defect Detection. Machine Learning Defect Detection is an important feature of modern Industrial Processes where the labeling phase of samples represents an actual issue from different point of views: e.g. time and financial costs.

We master our ability to create effective Active Learning Pipeline starting from classic benchmark datasets such as MNIST and medical-MNIST and other home made datasets.

Instead, to present our results, we selected 3 main dataset and we run 3 different experiments: casting product image data for quality inspection and potato leaf disease detection.

5.1 Experiment 1 Casting Manufacturing Product

Casting is a manufacturing process where a molten material is poured into a mould. Then the molten material is allowed to solidify into the desired shape. Sometime this process leads to defect products. The defects could be: holes, burr, shrinkage defects and many others. The main goal of this experiment is to exploit Active Learning techniques in this challenging defect detection context.

Dataset Details
The dataset contains a total of 1300 images. The images are 512 $\times$ 512. In the complete dataset there are 781 images of products with some defect (KO) and 519 images of defect-free products (OK). A custom split is applied in terms of training and test sets. In particular, the training set contains 469 OK samples and 731 KO images. Instead the test set is composed by 50 KO and 50 OK images. Some example images are provided in Fig.5.1. The orientation of the camera and the light condition slightly vary from one sample to the other.

Model Details
In the Active Learning context the focus should be on the dataset and this is the reason why the usage of a simple model is made. It consists in 2 Convolutional Layers and 3 Fully Connected Layers in sequence. ReLu is used as Activation Function. The total number of parameters of the model is: 9 966 606.

Figure 5.1: Example of images of the Casting Product Dataset. The first image is a defect-free product. Instead, in the second image you can see a product that is damaged in the external surface while in the third one is present a hole.

Training Details and Baseline

Data Augmentation techniques are exploited. In particular, RandomHorizontalFlip() and RandomVerticalFlip(). Moreover, it is also applied a RandomRotation() with parameter equal to 0.05. The images from 512 $\times$ 512 are cropped with the Resize() function to 300 $\times$ 300. The Learning Rate schedule follow a Polynomial Decay, The starting value of the Learning Rate is 0.001 and the final value is 0.0001. The usage of Stochastic Gradient Descent is made.
We trained our model with a standard training for a total of 8800 iterations in order to draw a representative baseline. The process was repeated for 5 times and the Mean Accuracy was equal to 95.60%.

Active Learning Process (ALP)

The Active Learning Process adopted follows the strategy presented in the previous chapter (Methodology).

The Budget was set to 120. This means that we start by labeling 120 samples and then at each Active Learning step the algorithm will ask to us to label a total of 120 images. We want to perform an analysis between the Active Learning Training Strategy and the baseline where the training cost is equal for both cases, That means that the total number of iterations (number of updates of our model weights) is equal in both AL and Traditional Training. For this reason, we fixed the number of iteration per active learning step to 880 and we perform a total of 10 Active Learning steps but after the 8 step we will not ask for more labeles. We will end our Active Learning Process with a total of 8800 iterations and 960 labeled images.

We performed the Active Learning Process for 5 times and you can see the result summarized in Tab.5.1.

Random Extraction Process (REP)

In order to see if the advantages of the Active Learning Process are introduced by the extraction of the most representative samples and not only from the deletion of redundant data, an additional experiment was performed. In this setting, we basically changed the extraction criteria from the coreset-active-learning to a merely random extraction process.

We performed the experiment with this extraction techniques for 5 times and you can see the result summarized in Tab.5.1.

	step	**budget**	**total iter.**	**total annot.**	**accuracy**
baseline	10	120	8800	1200	95.60
random	8	120	8800	960	95.43
al	8	120	8800	960	95.82

Table 5.1: Result of the experiment on the Casting Product Dataset

The model trained with the baseline strategy has a final accuracy equal to 95.60%. The training cost is 8800 iterations and the annotation cost is 1200 images (full dataset). Regarding the iterative methods, it is important to underline that the AL strategy reach an accuracy value that is better than the baseline and the random technique. In fact, AL reaches an accuracy score of 95.82%. This result is obtained with the same number of iteration as the baseline (same training cost) but with fewer samples (240 samples less than the baseline). Moreover, the iterative random experiment allows us to state the following: selecting diverse and representative samples using AL strategies will lead to a better model w.r.t to a model trained in a process that iteratively select samples at random.

5.2 Experiment 2 Potato Leaf Disease Detection

Blight is a plant disease caused by fungi. It can damage different plants and vegetables such as potato, pepper and similar. The knowledge and capability to detect this disease could allow us to prevent the spread and increase the production of food.

The second experiment is always focused on anomaly detection in an agricultural context. In this case, the dataset is composed by potato leaf. The potato leaf can be healty or they can have blight diseases. The objective is to train a model that will be able to correctly identify the leaves with the desease in a context of binary classification using as few datasamples as possible.

Dataset Details
The original dataset contained different types of plants and leaves. The selection of potato leaves is made in order to obtain a binary classification setting. The obtained potato leaves dataset has a total of 2152 images. The images are RGB and have a dimensions 256×256 pixels. A custom split is applied in terms of training and test set.

In particular, in the training set:

- 119 healthy potato leaf images
- 1933 early or late blight potato leaf images

As you can notice, the training set is really imbalance and this allow us to test the robustness of the Active Learning Process even in this challenging conditions.

For what concern the test set, it is composed by a total of 100 images: 33 healthy leaves and 67 leaves with the disease. Given that also the test set is imbalanced, instead of using the accuracy as the metrics for evaluating the performances of the model, the F1 score is used.

Model Details
Given that our focus is towards the understanding of the dataset, a simple model is used. As before it is composed by 2 Convolutional Layers and 3 Fully Connecte Layers in sequence. The ReLu function is used as Activation function. The total number of parameters is: 7 157 646.

Figure 5.2: Example of images of the Potato Leaf Dataset. The first image is a healthy potato leaf. Instead, in the second image you can see a leaf that has the early blight while in the third one is present a potato leaf that has the late blight.

The methodology applied to this experiment is the one presented in the previous chapter.

Training Details and Baseline
The baseline was drawn by training the NN for a total 2000 iterations, 7 epochs and a half given that the size of the total trainingset is almost 2000. Simple Data augmentation strategies where applied: Random Horizontal/Vertical Flip and Random Rotation. The LR schedule followed a Polynomial Decay of degree 2 in each AL step. The starting value was 0.001 and the final value of the LR was 0.0001. The baseline was computed as the mean of 5 different runs and it is equal to 99.41%

Active Learning Process and Random Extraction Process
The ALP has the same structure of the Experiment number 1. The difference is the following: givent that we have a total of roughly 2000 samples and that we want to perform 10 steps of AL, the budget was initially set to 200 (number of samples over number of AL steps). After the second step of AL we will not ask for more labels and we will end up with 400 labeled samples.

An analogous experiment was performed with the Random Extraction Technique in order to appreciate the benefits of the ALS and to monitor if the process was working fine. The budget of the REP was the same of the ALP and is initially set to 200.

Table 5.2: Result of the experiment on the Potato Leaf Dataset

	step	**budget**	**total iter.**	**total annot.**	**F1 score**
baseline	10	200	2000	2000	99.41
random	2	200	2000	400	97.87
al	2	200	2000	400	99.38

It is possible to observe that the accuracy of the baseline model that exploited the full dataset (2000 images) and 2000 iterations is 99.41%. This time the performance of the model that exploited the AL strategy does not go beyond the baseline measure but it is near with an F1 score of 99.38%. It is important to underline that in this case the number of annotation used in the AL process is less than half of the total size of the original unlabeled set. In fact the AL process used just 400 images. Instead, the random process that exploited 400 images loose almost 2% in performance w.r.t the baseline and the AL process.

Chapter 6

Conclusion

In this work we faced different challenges that concerned the Active Learning field in the Anomaly Detection context. Previous research work already proved the efficacy of AL strategies by testing them with benchmark dataset. In this thesis, we proved the efficacy and effectiveness of this algorithms also in a real application scenario.

The result are described thanks to two major experiments.

In the first one we applied the pipeline to an unlabeled set related to the manufacturing world. The dataset was populated with images where there were represented the output of a manufacturing process. The task was to identify the defect products. The main difficulty related to this experiment was the fact that the two classes (defect and defect-free) where extremely similar and in this way we putted under pressure the feature extraction phase and consequently the sample extraction phase. The efficacy and robustness of the applied strategy is shown thanks to the comparison between the AL process and the random extraction process. The AL technique show to have a stable advantage when the number of samples in the training set is reduce.

In the second experiment instead, the field of application was related to the agricultural context. In the dataset there were images of potato leaves. Those leaves could be healthy or they could have a disease. The task was to perform a binary classification process over this dataset. While the difference between healty-ill (KO-OK) were more evident with respect to the previous experiment, the major challenge of this dataset was its imbalanced nature. In fact, the healthy class represented barely the 5.8% of the total unlabeled set. This could create problem during the extraction phase, but also in this scenario the AL technique proved to be more powerful and efficient with respect to a random extraction technique.

The pipeline built in this work to perform the AL strategy proved to be consistent and robust in two different real context.

The main goal of this work was to show to the reader the efficacy of the Active Learning strategies not only in research scenario but also in real applications. These techniques are modern and more effort should be putted into the development of this strategies. The improvement of AL could automate many task that are performed by humans. If total automation could not be achieved, those technique could also be applied and exploited in processes with the "human-in-the-loop".

Bibliography

[1] Vighnesh Birodkar, Hossein Mobahi, Samy Bengio (29 Jan 2019) *Semantic Redundancies in Image-Classification Datasets: The 10% You Don't Need* URL: https://arxiv.org/abs/1901.11409

[2] Ozan Sener, Silvio Savarese (1 June 2018) *Active Learning for Convolutional Neural Networks: A Coreset-approach* URL:https://arxiv.org/abs/1708.00489

[3] Donggeun Yoo, In So Kweon (9 May 2019) *Learning Loss for Active Learning* URL: https://arxiv.org/abs/1905.03677

[4] Enrique Nueve, Sean Shahkarami, Seongha Park, Nicola Ferrier (24 June 2021) *Addressing the Constraints of Active Learning on the Edge* URL:https://ieeexplore.ieee.org/document/9460601

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton (1 July 2020) *A Simple Framework for Contrastive Learning of Visual Representations* URL: https://arxiv.org/abs/2002.05709

[6] Defays, D. (01 January 1977). An efficient algorithm for a complete link method. URL:https://academic.oup.com/comjnl/article/20/4/364/393966

[7] Hieu T. Nguyen, Arnold Smeulders (July 2004) *Active Learning Using Pre-clustering*
URL: https://icml.cc/Conferences/2004/proceedings/papers/94.pdf

[8] Michael Bloodgood, K. Vijay-Shanker (17 September 2014) *A Method for Stopping Active Learning Based on Stabilizing Predictions and the Need for User-Adjustable Stopping* URL: https://arxiv.org/abs/1409.5165

[9] Priya Goyal, Mathilde Caron1, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, Piotr Bojanowski (5 March 2021) *Self-supervised Pretraining of Visual Features in the Wild* URL:https://arxiv.org/abs/2103.01988

[10] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron1, Ishan Misra, Levent Sagun, Armand Joulin, Piotr Bojanowski (22 February 2022) *Vision Models Are More Robust And Fair When Pretrained On Uncurated Images Without Supervision* URL:https://arxiv.org/abs/2202.08360

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (10 December 2015) *Deep Residual Learning for Image Recognition* URL:https://arxiv.org/abs/1512.03385

[12] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, Liang Lin (13 January 2017) *Cost-Effective Active Learning for Deep Image Classification* URL:https://arxiv.org/abs/1701.03551

[13] Jia-Jie Zhu, Jose Bento (15 November 2017) *Generative Adversarial Active Learning* URL:https://arxiv.org/abs/1702.07956

[14] Pengzehn Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B.Gupta, Xiaojiang Chen, Xin Wang (October 2021) *A Survey of Deep Active Learning* URL:https://arxiv.org/abs/2009.00236

[15] Toan Tran, Thanh-Toan Do, Ian Reid, Gustavo Carneiro (26 April 2019) *Bayesian Generative Active Deep Learning* URL:https://arxiv.org/abs/1904.11643

[16] Samarth Sinha, Sayna Ebrahimi, Trevor Darrell (28 October 2019) *Variational Adversarial Active Learning* URL:https://arxiv.org/abs/1904.00370

[17] Melanie Ducoffe, Frederic Precioso (27 February 2018) *Adversarial Active Learning for Deep Networks: a Margin Based Approach* URL:https://arxiv.org/abs/1802.09841

[18] Razvan Caramalau, Binod Bhattarai, Tae-Kyun Kim (18 June 2020) *Sequential Graph Convolutional Network for Active Learning* URL:https://arxiv.org/abs/2006.10219

[19] Archit Parnami, Minwoo Lee (7 March 2022) *Learning from Few Examples: A Summary of Approaches to Few-Shot Learning* URL:https://arxiv.org/abs/2203.04291

[20] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, Amos Storkey (7 November 2020) *Meta-Learning in Neural Networks: A Survey* URL:https://arxiv.org/abs/2004.05439

[21] Dong-Hyun Lee (July 2013) *Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks* URL: From https://www.researchgate.net/directory/publications

[22] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong, Quoc V. Le (1 March 2021) *Meta Pseudo Labels* URL:https://arxiv.org/abs/2003.10580