Università degli Studi di Napoli Federico II

Ph.D. Thesis

in

Information Technology and Electrical Engineering

Trustworthy AI:
the Deep Learning Perspective

Raising awareness on reproducibility, security and fairness concerns at the dawn of the fourth industrial revolution

Stefano Marrone

Tutor: Prof. Carlo Sansone

Coordinator: Prof. Daniele Riccio

XXXII Ciclo

Scuola Politecnica e delle Scienze di Base
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

List of Acronyms

The following acronyms are used throughout this text.

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

ANN Artificial Neural Network

CNN Convolutional Neural Network

PA Presentation Attack

LD Liveness Detector

AS Authentication System

BAS Biometrics-based Authentication System

FAS Fingerprint-based Authentication System

Table of contents

Summary

In recent years the term “Artificial Intelligence” (or AI) has become more and more an integral part of the daily life of all of us. We are increasingly dealing with *smart* mobile phones, *intelligent* voice assistants, *robotic* chats, etc. Our interaction with these “intelligent systems” has become so predominant and widespread that even the world of industry has begun to use such AI in factory life and logistics. The term artificial intelligence refers to the ability of a computer (an *artificial* entity) to perform functions resembling the typical reasoning of the human mind (i.e. *intelligence*). Indeed, Marvin Minsky, Alan Turing, Frank Rosenblatt and other AI pioneering studies focused on the development of artificial entities able to autonomously do things usually requiring human intelligence (e.g. the ability to make a decision based on the status of the environment) to be performed.

Although commonly thought to be a child of the last years, the first studies on artificial agents began on the eve of the second world war. From that moment on, amid ups and downs, researches started focusing on the development of theories and mathematical models laying the foundation for the upsurge of artificial intelligence in a wide variety of domains. Nowadays, the term AI is widely abused, and one of the unpleasant effects of this spread

with the mass audience is the confusion made with all its related terms, such as “Pattern Recognition”, “Machine Learning”, “Deep Learning”, etc., too often used interchangeably. Indeed, what usually media refers to with AI is actually Machine Learning (ML), a term used to describe the ability of this kind of AI systems to *learn from examples*, just as we humans learn from experience. This peculiarity has made it possible to relieve the programmer from the task of writing the sequence of operations necessary to perform a given task (algorithm), allowing them to perform increasingly complex tasks, for which it would have been impossible to code a solution.

Among all machine learning models, Artificial Neural Networks (ANN, often referred simply as Neural Network - NN) are definitely the branch that has been receiving the most media coverage since their parallel layered structure of computing elements (i.e. artificial neurons) is inspired to the human brain complex interconnected structure of biological neurons. More recently, the introduction of General Purpose GPU (GP-GPU) computing, the development of free and easy to use frameworks, the availability of huge labelled dataset and progresses in gradient-based optimisation, determined the uprising of *Deep Neural Networks*. The term “Deep Learning” (DL) refers to a particular subset of ANNs characterised, inter alia, by a very “depth structure” (i.e. made up of several layers). Another key aspect of deep models is their ability to autonomously learn the best or set of features for the task under analysis, to the point of even exceeding human capabilities in some tasks. This characteristic, known as *feature learning*, has played a key role in the recent spread of AI since allowed DL use also in domains lacking effective expert-designed features.

The impact of AI, and in particular of deep learning, on the industry has been so disrupting that it gave rise to a new wave of research and applications that goes under the name of *Industry 4.0*. This expression refers to the application of AI and cognitive computing to leverage an effective data exchange and processing in manufacturing technologies, services and transports, laying the foundation of what is commonly known as *the fourth industrial revolution*. As a consequence, today's developing trend is increasingly focusing on AI based data-driven approaches, mainly because leveraging user's data (such as location, action patterns, social information, etc.) can make applications able to adapt to them, enhancing the user experience. To this aim, tools like automatic image tagging (e.g. those based on face recognition), voice control, personalised advertising, etc. process enormous amounts of data (often remotely due to the huge computational effort required) too often rich in sensitive information.

Artificial intelligence has thus been proving to be so effective that today it is increasingly been using also in critical domains such as facial recognition, biometric verification (e.g. fingerprints), autonomous driving etc. Although this opens unprecedented scenarios, it is important to note that its misuse (malicious or not) can lead to unintended consequences, such as unethical or unfair use (e.g. discriminating on the basis of ethnicity or gender), or used to harm people's privacy. Indeed, if on one hand, the industry is pushing toward a massive use of artificial intelligence enhanced solution, on the other it is not adequately supporting researches in end-to-end understating of capabilities and vulnerabilities of such systems. The results may be very (negatively) mediatic, especially when regarding borderline domains such those related to subjects privacy or to ethical and fairness, like users profiling, fake news generation, reliability of autonomous driving systems, etc.

We strongly believe that, since being just a (very powerful) tool, AI is not to blame for its misuse. Nonetheless, we claim that in order to develop a more ethical, fair and secure use of artificial intelligence, all the involved actors (in primis users, developers and legislators) must have a very clear idea about some critical questions, such as *"what is AI?"*, *"what are the ethical implications of its improper usage?"*, *"what are its capabilities and limits?"*, *"is it safe to use AI in critical domains?"*, and so on. Moreover, since AI is very likely to be an important part of our everyday life in the very next future, it is crucial to build trustworthy AI systems.

Therefore, *the aim of this thesis is to make a first step towards the crucial need for raising awareness about reproducibility, security and fairness threats associated with AI systems*, from a technical perspective as well as from the governance and from the ethical point of view. Among the several issues that should be faced, *in this work we try to address three central points*:

- understanding what “intelligence” means and implies within the context of artificial intelligence;
- analyse the limitations and the weaknesses that might affect an AI-based system, independently from the particular adopted technology or technical solutions;
- assessing the system behaviours in the case of successful attacks and/or in the presence of degraded environmental conditions.

To this aim, the thesis is divided into three main parts: in part I we introduce the concept of AI, focusing on Deep Learning and on some of its more crucial issues, before moving to ethical implications associated with the notion of “intelligence”; in part II we focus on the perils associated with the reproducibility of results in deep learning, also showing how proper network design can be used to limit their effects; finally, in part III we address the implications that an AI misuse can cause in a critical domain such as biometrics, proposing some attacks duly designed for the scope.

The cornerstone of the whole thesis are *adversarial perturbations*, a term referring to the set of techniques intended to deceive AI systems by injecting a small perturbation (noise, often totally imperceptible to a human being) into the data. The key idea is that, although adversarial perturbations are a considerable concern to domain experts, on the other hand, they fuel new possibilities to both favours a fair use of artificial intelligence systems and to better understand the “reasoning” they follow in order to reach the solution of a given problem. Results are presented for applications related to critical domains such as medical imaging [111, 138], facial recognition [113] and biometric verification [113]. However, the concepts and the methodologies introduced in this thesis are intended to be general enough to be applied to different real-life applications.

Part I

The Fourth Industrial Revolution

In the last years, the impact of *Artificial Intelligence* (AI) on the industry has been so disrupting that it gave rise to a new wave of research and applications that goes under the name of "Industry 4.0" (figure 1). This term refers to the application of AI and cognitive computing to leverage an effective data exchange and processing in manufacturing technologies, services and transports, laying the foundation of what is commonly known as the fourth industrial revolution [148, 90].

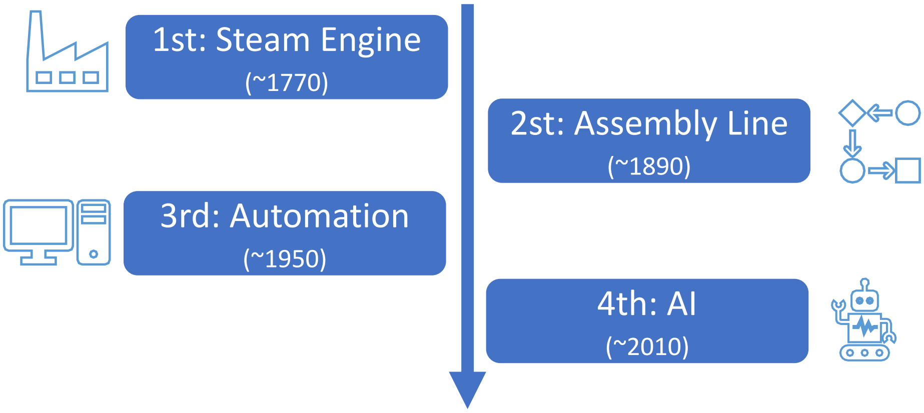


Figure 1: Timeline for the four industrial revolutions: the first, based on the invention of the steam engine; the second, thanks to the development of the assembly line by Henry Ford; the third, with the development of computers and automation; the fourth, supported by artificial intelligence.

Several are the industries impacted by it, with logistic [18], manufacturing [173] and transport systems [160] representing some of the fields in which machine learning is expected to have a very important impact in the next future. Nevertheless, several are the contexts in which some effects are already visible:

Automotive and avionics are probably the first use cases in which the use of machine learning, in the shape of auto-pilot [196, 23], has been applied. However, in a more and more connected world, that is aiming to self-driving vehicles, the proper control of such amount of traffic in a precise and effective way, together with pick-hour and congestion/delay predictions [184] becomes some of the biggest challenges to face;

The smart city represents what engineers, architects and sociologists expect to be the place in which we, as humans, will live in the next future. Although the term can suggest utopistic or futuristic dreams, it actually aims in designing "human-in-the-loop" cities intended to support the wealth of citizens, for example by reducing stress and pollution [103];

Smart energy harvesting and grid design (e.g. pick usage prediction and smart distribution of the electricity over the power grid [208]) are becoming extremely important, both on a local [151] and on a national [165] scale, in a world more and more relying on renewable energy rather than fossil fuels;

Telemedicine that, thanks to the ultra wide-band connectivity delivered by 5G technology [185], allow to analyse huge amounts of patients data to provide AI-based solution in several clinical contexts [108];

Robotics in industrial applications where, thanks to the advances in the computer vision field, cooperate with and assist humans in safety-critical environments [99, 69];

Security enforcement, with AI opening new scenario in user identification [30] and anti-malware protection [91];

Smart assistants that, from chatbots [8] to smart assistants [41], are revolutionising human-machine interaction.

The reported examples are just some wise applications and, unfortunately, represent only one side of the coin. Indeed, the disrupting spread and success of AI comes with some downsides, which find among their most (in)famous examples applications related to the violation of subjects' privacy, and unethical and unfair behaviours, such as users profiling [6], fake news generation [140], autonomous weapon systems [10], discriminative advertisement [39], etc. We strongly believe that, since being just a (very powerful) tool, AI is not to blame. Nonetheless, we claim that in order to develop a more ethical, fair and secure use of artificial intelligence, all the involved actors (in primis users, developers and legislators) must have a very clear idea about some critical questions, such as *"what is AI?"*, *"what are the ethical implications of its improper usage?"*. It is clear that developing an own idea about these questions is a process requiring time, expertise and open-mind. Therefore, to provide the tools needed to fully understand the claims, the scenarios, and the contributions made in this work, this first part will start by introducing the concept of AI (chapter 1), focusing on Deep Learning and on some of its more crucial issues, before moving to ethical implications (chapter 2) associated with the notion of "intelligence".

1

The Artificial Intelligence Era

The term Artificial Intelligence (AI) refers to the ability of a computer to perform functions resembling the typical reasoning of the human mind. Although commonly thought to be a child of the last years, the first studies on the development of an artificial agent started in the early forty of the past century. From its very beginning, AI has been the centre of the debate between scientists and philosophers, with the former interested in the theory and techniques aimed at the development of algorithms to allow the machines to show "intelligent" skills and actions (at least in specific domains), while the latter more interested in the aspects related to the possible implications that can be triggered by considering an artefact as an "intelligent entity". According to Marvin Minsky, considered one of the pioneer fathers of IA together with Alan Turing and Frank Rosenblatt, the purpose of this new field is "to develop machines able to autonomously do things that would require intelligence if they were made by humans" [116].

Several are the problems arising with this definition of artificial intelligence. The first is clearly in the lack of a universally agreed definition of "intelligence", with almost all the interpretations so far proposed impractical to prove or too much fuzzy. This is the reason why, seventy years after its first introduction, the most agreed definition is actually based on "the imitation game", a test proposed by Alan Turing in his 1950 article titled "Computing machinery and intelligence" [188]. The idea is basically to consider a machine intelligent if capable, when hidden behind a screen and connected with the world through an appropriate communication interface, to mislead a human tester by convincing they to be interacting with another human being. In 2014 a chatbot (i.e. a conversational agent) made the news thanks to its ability to pass the Turing test [1]. This claim started an ongoing debate [158] on whether this can actually be considered a proper victory or not, since what really happened was the bot able to mislead $\sim 33\%$ of a panel of judges into believing it was a real young boy after a five-minute conversation. Although subtle, the difference is in the fact that the environmental settings and the performed task are far from being general and accepted as realistic. This is anything new since it has already been shown that, under certain conditions, it can be straightforward to pass the Turing test [82]. Also, this proof comes as no surprise, since the problem is intrinsically rooted in the difference between the two AI ideologies:

The Strong thesis claiming that a properly designed, programmed and trained machine can be endowed with pure intelligence in no way distinguishable from humans one;

The Weak thesis arguing that no matter how closely a machine will ever be able to resemble the human cognitive process, it will never be able to fully reproduce the complexity and versatility of humans intelligence.

At the time of writing this thesis, despite the availability of AI models able to compete with and, in some cases, even surpass humans [12, 182, 118], we are still far from a general AI model, able to fulfil the strong thesis. Indeed, even the top-performing AI systems have the characteristics to be usable only on the task they have been designed for, mainly because of the wide variety of available "input signals" and the lack of cross-domain/task generalisation ability. Nonetheless, the use of artificial intelligence keeps spreading [42], not only in high-tech companies and domains but even in everyday applications, with a worldwide impact on the economy (figure 1.1).

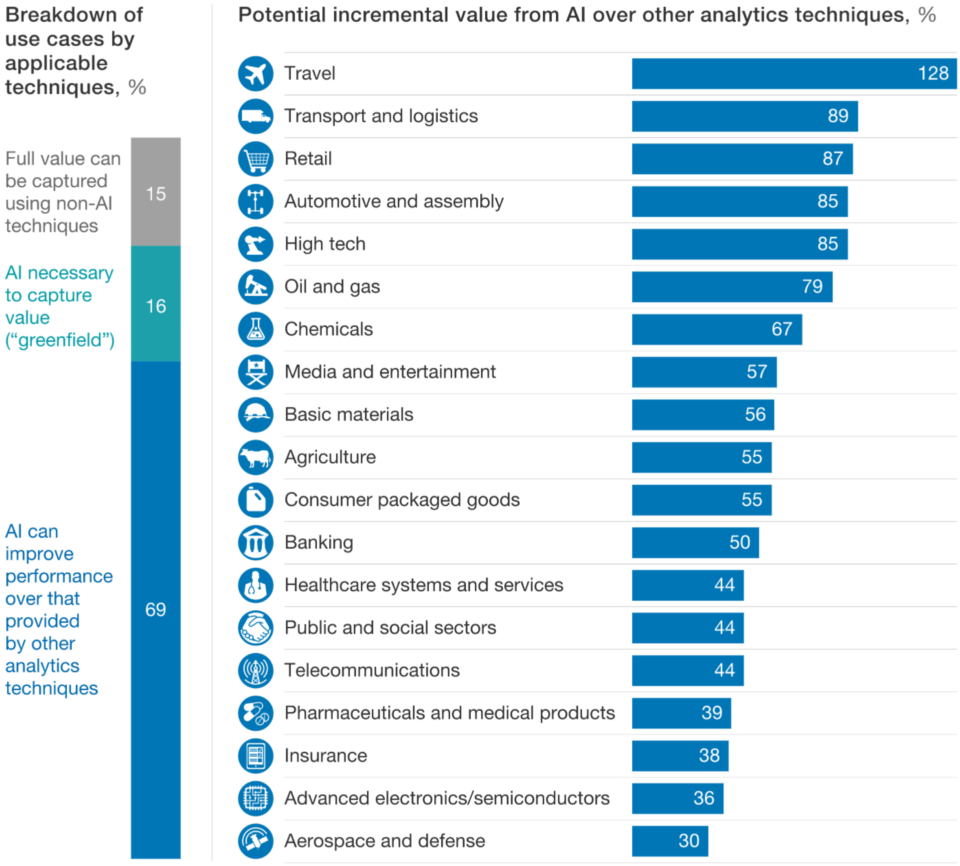
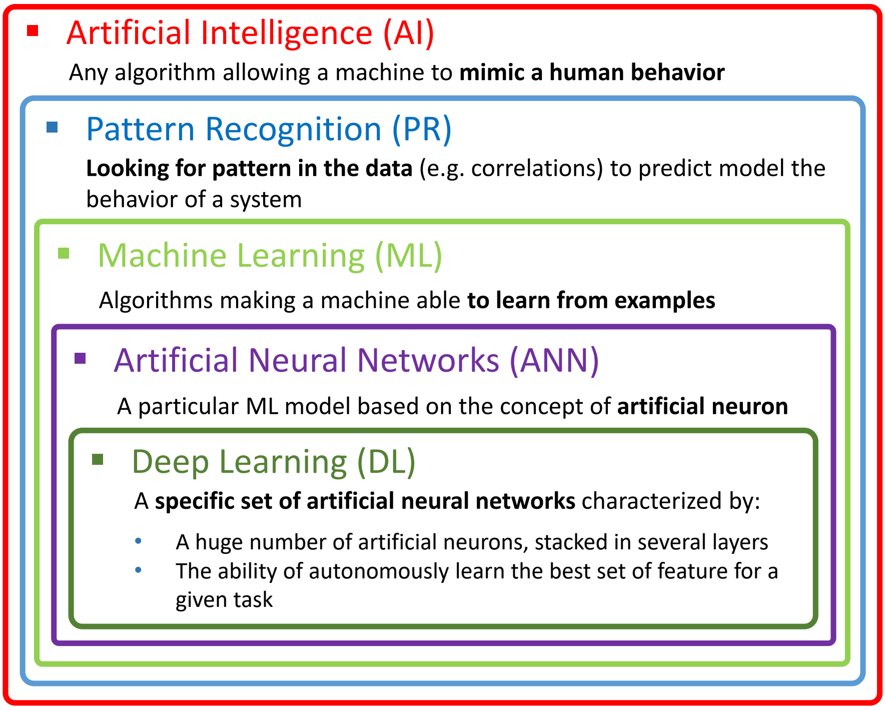


Figure 1.1: McKinsey Global Institute analysis on the potential impact that AI will have in the next future on some important industries [109].

1.1 From Shallow to Deep Neural Networks

In recent years, the term "Artificial Intelligence" has become more and more an integral part of our daily life. We are increasingly dealing with "smart" mobile phones, "intelligent" voice assistants, "robotic" chats, etc. Nowadays, the term AI is widely abused, and one of the unpleasant effects of this spread with the mass audience is the confusion made with all its related terms, such as "Pattern Recognition", "Machine Learning", "Deep Learning", etc., too often used interchangeably. Thus, with the aim of help the reader in better understanding the terminology and the contributions made in this thesis, figure 1.2 reports a brief toponymy of the most (mis)used AI-related terms.



- Artificial Intelligence (AI)

  Any algorithm allowing a machine to **mimic a human behavior**

- Pattern Recognition (PR)

  Looking for pattern in the data (e.g. correlations) to predict model the behavior of a system

- Machine Learning (ML)

  Algorithms making a machine able **to learn from examples**

- Artificial Neural Networks (ANN)

  A particular ML model based on the concept of **artificial neuron**

- Deep Learning (DL)

  A **specific set of artificial neural networks** characterized by:

  - A huge number of artificial neurons, stacked in several layers
  - The ability of autonomously learn the best set of feature for a given task

Figure 1.2: A brief toponymy of some of the most common terms related with "artificial intelligence". A box included into another represents the relation between a sub-concept and a concept.

What usually media refers to with AI is actually machine learning (ML). Indeed, what is really taking the scenes is the ability of this kind of AI systems to *learn from examples*, referring to their ability to learn how to perform a task through the use of examples. This characteristic relieves the programmer from the burden of writing the sequence of operations necessary to perform a given task (algorithm), consequently allowing the system to face complex tasks, for which it would have been impossible to code a solution.

Among all ML models, Artificial Neural Networks (ANN) are definitely the branch that has been received the most media coverage due to their "inspiration" to the human brain: as the latter consists of a complex interconnected structure of biological neurons, the former is a network of artificial ones. In particular, an artificial neural network (often simply referred as Neural Network - NN) is a parallel structure (since each artificial neuron operates in parallel with the other) whose elements are organised in layers and interact each other to perform, after a suitable training stage, the desired task.

More recently, *Deep Learning* is the term that has started been using as a synonym for AI/ML. The term refers to a particular subset of ANNs characterised, inter alia, by a very "depth structure" (i.e. made up of several layers). The other key aspect of deep models is their ability to autonomously learn, during the training stage, the best input representation (or set of features) for the task under analysis. This characteristic, known as *feature learning*, has played a key role in the recent spread of AI since allowed its use also in domain lacking effective expert-designed features. When it comes to elaborate images, Deep Convolutional Neural Networks (D-CNNs or simply CNNs) have shown incredible performance in a wide range of research fields, including natural image processing [199], biomedical applications [66], biometrics [35] and many others [202, 167, 86]. The core of CNNs are convolutional layers, namely layers of neurons leveraging the concept of convolution between the input and a kernel to perform the feature extraction. Unlike the human-based feature extraction, where the feature design process is fixed, convolutional layers allows CNNs to adapt the feature extraction during the training itself since the kernels used for the convolutions are learnt together with the other neurons' weights. When several convolutional layers are stacked in sequence, the whole network starts learning how to extract a hierarchical set of features, from a low to a high level of details (figure 1.3)[1].

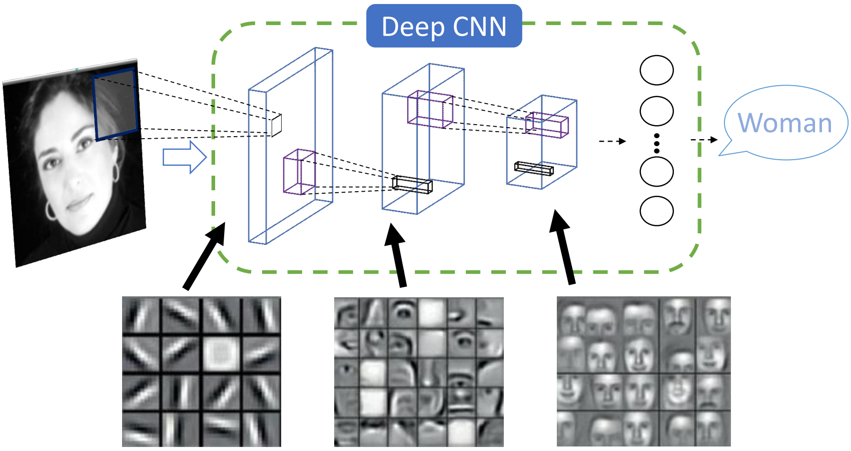


Figure 1.3: Illustration showing how a 3-layered CNN learns to describe a complex image as composed by many simpler concepts. Starting from the input (the woman image on the left), each convolutional layers learns how to combine informations from the previous layer to extract new hierarchical features: the first convolutional layer is directly connected to the image and can learn how to extract simple pattern (e.g. edges); the central convolutional layer exploit the input feature map (low-level geometrical features extracted by the first convolutional layer) to extract more complex pattern, such as eyes, noses, etc; the last convolutional layer uses the middle-level features to learn high-level concepts, like faces. In the end, a fully connected layer learns how to use high-level features to perform the desired task (gender classification in the example), generating the output for the provided input.

[1] A detailed explanation of ANNs and of CNNs is beyond the purposes of this work. For a more complete and detailed explanation please refer to [22] and to [63].

Contrary to popular belief, CNNs are not utterly something recent (figure 1.4). Indeed, the first ANN designed to perform the feature engineering phase has been introduced in 1980 under the name of *neocognitron* [56]. The gap between the time they have been theorised and the moment they have been developed is mostly due to the lack of suitable computational power, enough training data and some mathematical limitations. Moreover, the increasing popularity of kernel machine dampened the enthusiasm for neural networks.

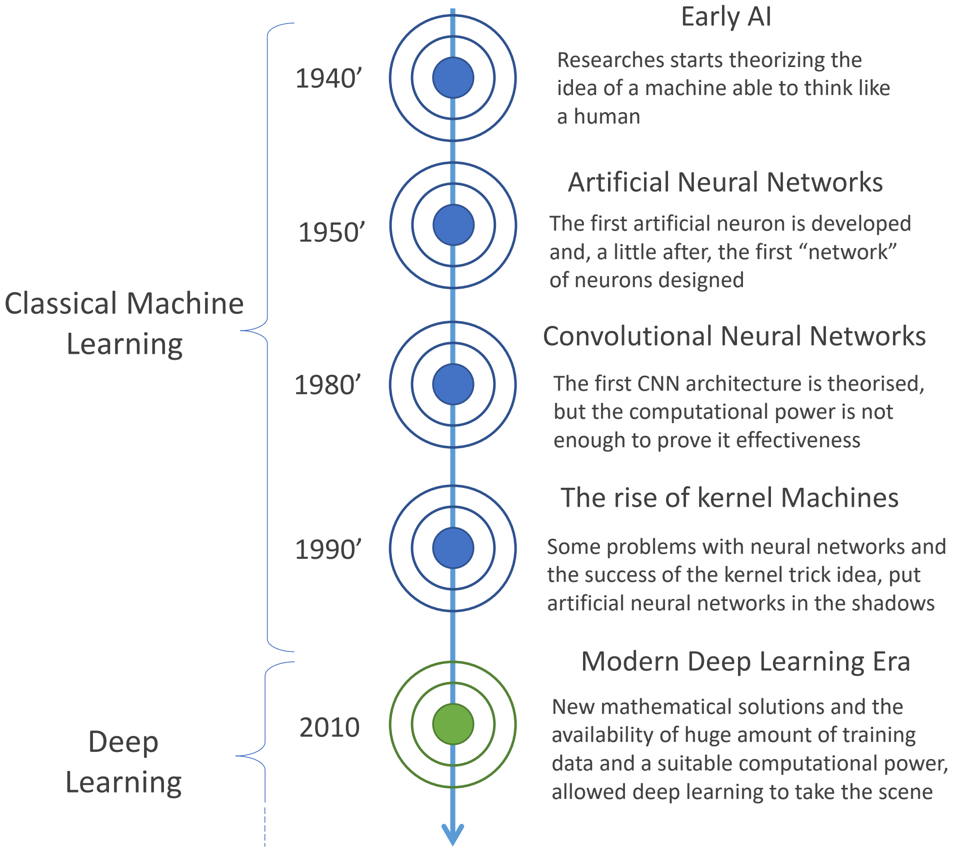

Early AI
Researches starts theorizing the idea of a machine able to think like a human
Artificial Neural Networks
The first artificial neuron is developed and, a little after, the first "network" of neurons designed
Convolutional Neural Networks
The first CNN architecture is theorised, but the computational power is not enough to prove it effectiveness
The rise of kernel Machines
Some problems with neural networks and the success of the kernel trick idea, put artificial neural networks in the shadows
Modern Deep Learning Era
New mathematical solutions and the availability of huge amount of training data and a suitable computational power, allowed deep learning to take the scene
1940'
1950'
1980'
1990'
2010
Classical Machine Learning
Deep Learning

Figure 1.4: Short timeline for some of the AI most important events.

This was the case until a convolutional neural network [95] won the 2012 Large Scale Visual Recognition Challenge [152], a popular open-source contest in which participants have to correctly classify images to one of the 1000 classes. From that moment on, an increasing interest stated again to be paid by researches on the study and on the application of CNNs in several contexts, giving rise to new CNN architectures, training and optimization strategies able to compete with, and in some cases surpass, humans in many tasks [118, 12]. Several are the factors that concurred to make neural networks, and in particular deep architectures, regain popularity up to unprecedented heights. Excluding some mathematical intuitions, four are the factors that contributed the most:

- The rise of **Big Data** in late 2000, a term referring to the collection of massive amounts of data often unstructured and coming from different sources (e.g. images, text, audio, medical signals, etc.). The availability of public and free to use collections of labelled samples started to show the limits of kernel machines while allowing researches to experiment with richer (in neurons) ANNs;

- The computational power needed to optimise the huge number of parameters (typical of deep architecture) and to iteratively elaborate the involved massive training datasets has been a technical limitation for a long time. On this regards, the advances in **General-Purpose GPU (GP-GPU) computing** strongly sustained the spread of deep learning models, allowing a significant reduction in the required training times. A key role has been playing by NVIDIA with its CUDA technology [127], an adaptation of C++ explicitly intended to be used with GPU stream multiprocessors;

- Barkley Artificial Research (BAIR) group[2] has been one of the first research group understanding the potential of deep learning. To sustain

[2] https://bair.berkeley.edu/

its spread they developed and released Caffe [84], a BSD-licensed C++ framework for the development of deep architectures. To increase its usage in the research community, Caffe was intended to be easily accessed by third-party applications (thanks to the availability of Python and MATLAB API) and to natively support GPU acceleration. In parallel, the group also released pre-trained implementations of the CNNs that, time by time, started to be designed by researches, further increasing its success within the community. Shortly after, some big players followed the example by releasing their own deep learning framework: PyTorch (by Facebook), CNTK (by Microsoft) and TensorFlow (by Google). Also famous applications for scientific calculation (e.g. MATLAB by The Mathworks[3] and Mathematica by Wolfram[4]) started officially supporting deep learning;

- The last step to do before really considering deep learning within everyone's reach was to make GPU computing accessible (since suited GPU were, and sometimes still are, relatively expensive). Some companies quickly catch this business opportunity, starting to provide easy to use web-based virtual machines intended to allow developers to use remote GPUs for a little price. In 2017, Google publicly released Colaboratory[5], a totally free web-based IDE providing remote GPU acceleration.

1.2 The Burden of Deep Architectures

In the machine learning field, it is well known that what matters above classifier are the features used to describe the entities under analysis. For this reason, a lot of new features have been designed by domain experts to improve

[3] https://it.mathworks.com/products/deep-learning.html

[4] https://reference.wolfram.com/language/guide/MachineLearning.html

[5] https://colab.research.google.com/

classification results in a widespread of different fields, including computer vision (such as natural and biomedical image processing), automatic speech recognition (ASR) and time-series analysis. As seen in the previous section, this rapidly changed with the advent of Deep Convolutional Neural Networks (CNNs), able to autonomously learn the best set of features to effectively face the task under analysis, particularly for those domains lacking effective expert-designed features [98].

Model	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
AlexNet [95]	0.593	0.818	60,965,224	8
VGG16 [166]	0.715	0.901	138,357,544	23
VGG19 [166]	0.727	0.910	143,667,240	26
MobileNet [79]	0.665	0.871	4,253,864	88
DenseNet121 [80]	0.745	0.918	8,062,504	121
Xception [33]	0.790	0.945	22,910,480	126
InceptionV3 [180]	0.788	0.944	23,851,784	159
ResNet50 [76]	0.759	0.929	25,636,712	168
DenseNet169 [80]	0.759	0.928	14,307,880	169
DenseNet201 [80]	0.770	0.933	20,242,984	201
InceptionResNetV2 [178]	0.804	0.953	55,873,736	572

Table 1.1: Brief list of some famous CNNs designed to face the ImageNet [152] classification challenge. For each network, the top-1 and top-5 accuracy, together with the number of parameters and layers (depth) are reported. Reported numbers refers to the corresponding Keras [34] implementation[6].

Researches agree that the strength of a CNN is in its deep hierarchical architecture able to learn features at multiple levels of abstraction [179, 48]. Although this characteristic gives CNNs a great representational capacity, it also comes with a huge number of parameters to learn. Indeed, the term *deep* refers to the number of layers and thus, indirectly, to the total number of neurons (parameters) in the network, that can easily get over 10^6 even for medium-sized models (as briefly reported in table 1.1). Therefore, despite

[6] https://keras.io/applications/

the efforts made to design increasingly compact networks [76, 180, 80], to avoid incurring in over/under-fitting, a suitable number of annotated samples is required to properly estimate millions of parameters when training a deep CNN from scratch (i.e. starting from randomly initialised weights). Unfortunately, gathering a big dataset is a difficult, expensive and time-consuming task, that can become even harder in domains where a large number of samples is difficult to collect. A typical case is biomedical imaging, where not only collecting huge datasets is technically laborious (privacy-related issues, different protocols, etc.), but it is intrinsically hard because of the huge class imbalance (e.g. between positive and negative oncology patients). As a matter of fact, training a CNN from scratch requires *expertise* to design an architecture suitable for the problem under analysis, *experience* to effectively tune all the hyper-parameters and *proper computational power* to train the network in a reasonable time.

To address this problem, an increasingly popular solution is to adopt *Transfer Learning*, a term referring to the act of transferring the knowledge learnt in a task (i.e. leveraging pre-trained CNN parameters) with a proper amount of available training data, to another (sometimes very different) task. Although the idea is not new [132], is only with deep learning that it showed its full potential with many authors that, in the last years, are increasingly exploiting transfer learning to derive new state-of-art solutions in different fields. In practice, transfer learning can be sought by following two different approaches:

- the first, known as *fine-tuning*, consists in adapting the pre-trained CNN on the new task. This is done by "freezing" (setting the learning rate to 0) some layers (usually the convolutional ones), before performing a "re-training" (starting from the pre-trained weights) of the network on the data coming from the new task. It is worth to note that it is possible to modify/add/remove any of the layers, keeping in mind that those new layers will not be able to leverage pre-trained weights. This operation

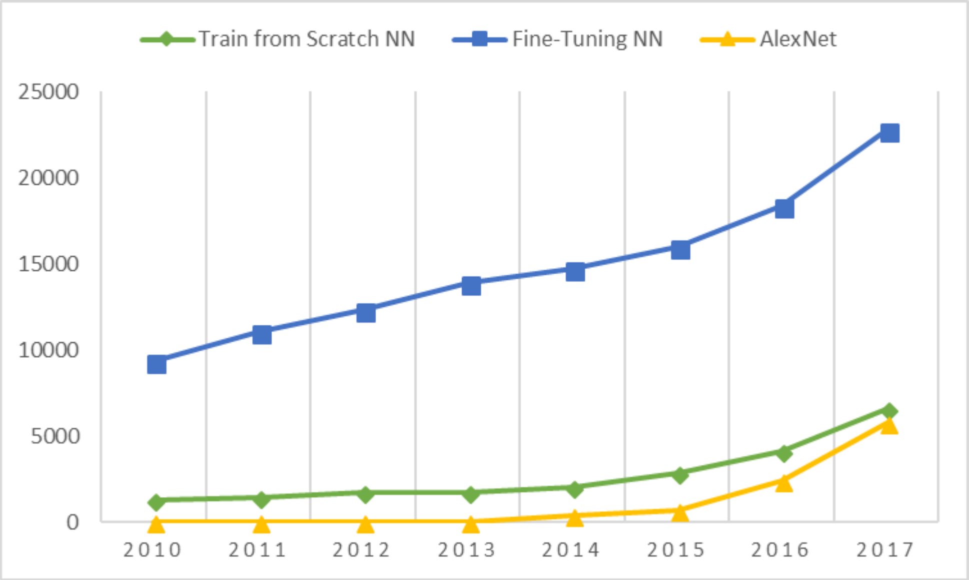


Figure 1.5: Comparison of number of papers published between 2010 and 2017 that i) train a neural network from scratch (in green), ii) fine-tune a neural network (in blue) and iii) use AlexNet, both for fine-tuning or as a feature extractor (in yellow). The search was performed in Google Scholar using "train from scratch neural network", "fine tune neural network" and "AlexNet fine tuning | AlexNet feature extractor" as keywords for the first, the second and the third case respectively.

is always needed at least for the classification layer since the desired new classification task (usually) has a different set of possible classes;

- the second consists in the use of the pre-trained CNN as a feature extractor. In this case, the idea is that the pre-trained CNN has learnt a set of features that are supposed to be effective also for the new task. Thus, it is possible to leverage this knowledge by feeding the pre-trained network with the new task data, obtaining a new set of features by taking the output of one of the CNN layers (usually one of the fully connected). This features can then be used to train any kind of machine learning model on the new desired task.

Both approaches exploit the CNN inherent ability to learn very effective hierarchical features that demonstrated to be suitable for a widespread of distinct task [43, 204, 72], also including those that differ greatly [183]. In particular, it had been showed that deep architectures pre-trained on the ImageNet dataset [152] show great flexibility and versatility [72, 135, 145] mostly thanks to their ability to learn Gabor-like low-level feature [75] (edge, lines, shapes), combined to form feature able to extract complex details and textures. Among all the available dataset and deep CNN, in automatic (not necessarily natural) image classification the duo ImageNet[152]-AlexNet[95] is one of the most used (figure 1.5) thanks to i) the ImageNet broad number of different class/images and ii) to the AlexNet simple (easy to understand and adapt) but powerful structure, consisting in 5 convolutional layers and 3 fully connected ones (for a total of $60,965,224$ parameters), using ReLu [123] as activation function.

Figure 1.6 shows how to approach transfer learning with AlexNet, both by using it as feature extractor and by following the fine-tuning approach: in the first case, the output of the last hidden layer (fc7) is used as features set to feed an external classifier (i.e. SVM, Random Forest, etc), while, in the latter case, the original 1000 classes classification layer is dropped and replaced by (for example) a binary one and then a new training process is performed. Practically, both approaches are very effective since they allow to use a CNN designed to be enough powerful to face a 1000 classes problem for a classification task that (as in the described example) might even only have 2 classes. In particular, using the net as a feature extractor is a versatile way to exploit past learnt knowledge since it is possible to take the output from any of the network layers according to the desired feature abstraction level (recalling from section 1.1 that early layers learn lower level features). On the other hand, fine-tuning is easier to apply (since it does not involve external classifiers to tune and to optimise) and, thanks to the re-training, allows to adapt the network as a whole to the new task. Though one may not necessarily be better than the other, as a rule of thumb the latter is usually preferred when the number of training samples and the available computational power are enough to train (although not from scratches) a CNN, while the former is usually preferred when one of these conditions is not satisfied.

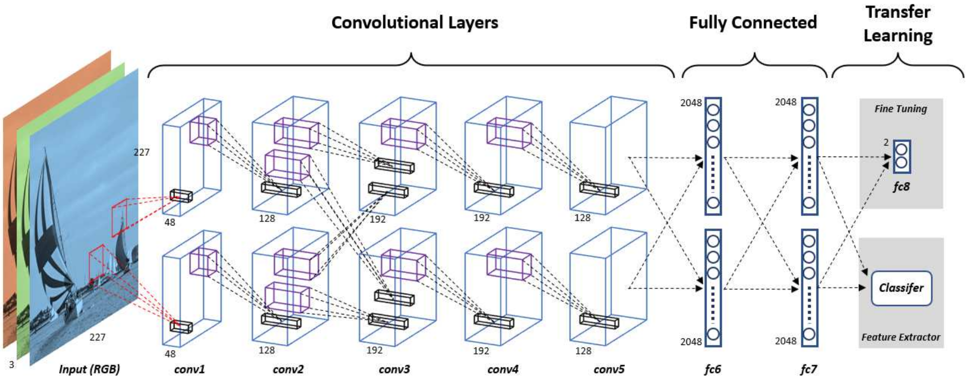


Figure 1.6: A representation of transfer learning using AlexNet. From left to right: the net input (a $227*227$ pixels - 3 channels (RGB) image), convolutional layers, fully connected layers and adaptations needed to perform transfer learning. In particular, on the top right side there is an example of how to fine-tune AlexNet for a binary classification task, while on the bottom right side there is an illustration showing the use of the output from the last hidden layer as features set for an external classifier (i.e. using AlexNet as feature extractor). As a note, the illustration reports the original structure of AlexNet distributed over two GPU (although a single GPU implementation is nowadays commonly used).

1.3 Deceivable AI

Optical illusion is the term used to describe figures (e.g. images, illustrations, drawing, etc.) able to mislead the human visual perception system (figure 1.7). Since the hierarchical structure of deep neural networks (section 1.1) is inspired by the human brain, is it possible to mislead them similar to the way optical illusions mislead us?

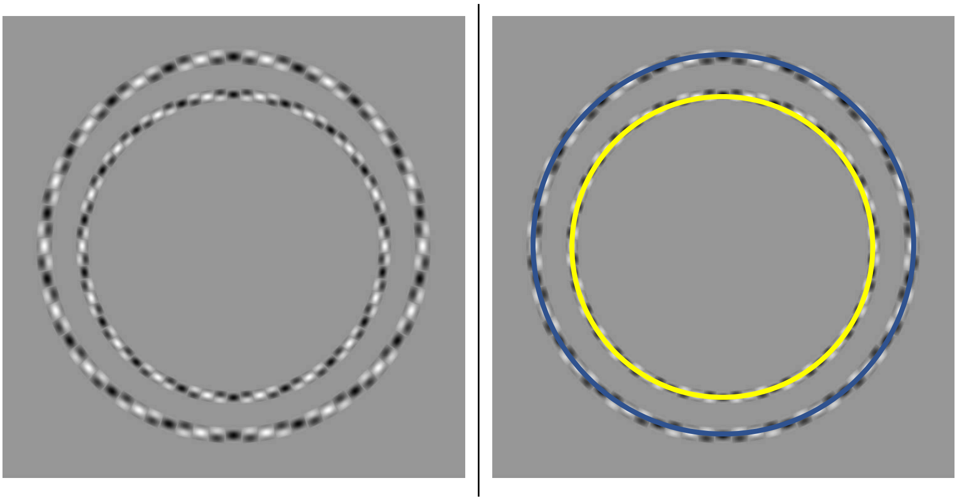


Figure 1.7: Illustrative optical illusion. Humans visual perception system is not able to determine how many circles there are in the figure on the left since the circles' texture and the used background colour daze our brain expectation. However, as we over-impose some solid-coloured reference shapes, figure on the right, our brain suddenly recognise the underlying circles.

The answer is yes. In particular, it has been shown [181] that it is possible to arbitrarily cause state-of-the-art CNNs to misclassify an image by applying on it a suitable small perturbation (often even imperceptible to human eyes) and that regularization techniques are useless in this case (since it is not the result of overfitting). On the same line, some recent works [121, 203] started to develop effective ways to perform targeted back-door attacks against CNNs aimed in creating samples that will always make the target CNN behave as desired by the attacker.

Scientists' concerns are serious since some studies showed that it is possible to effectively perform this kind of attacks also in real-world scenarios [25]. Therefore, it is clear that facing this problem is of primary importance, especially in the view of safety-critical applications such as autonomous driving, subject identification, fake news detection, etc. To this aim, in the last years several researchers started working on the development of effective methods to detect and defend against them [47, 200]. As a consequence, more sophisticated and sneaky attacks have been developed, giving rise to a game between the attackers and the defenders that seems far from ending and, up to date, sees the former ahead. However, *the vulnerability of CNNs to this kind of attacks could open new possibilities for privacy protection and to enforce fairness in AI-based applications*. Therefore, to better understand the intuitions and contributions made in the next chapters, the following sections will introduce some of the attacks that are recently making the news[7].

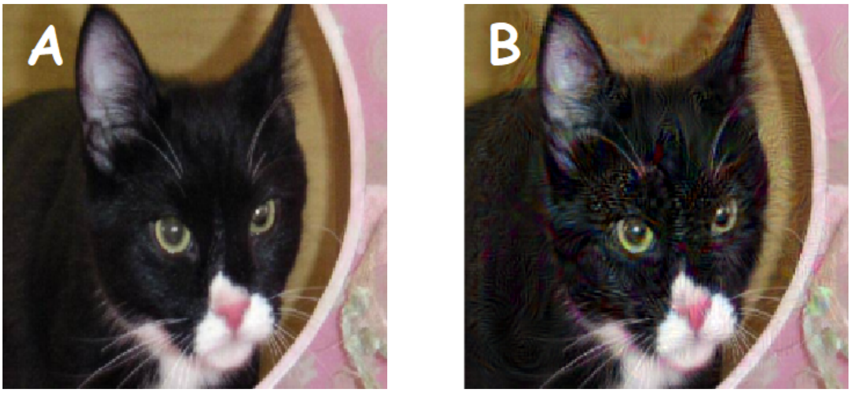
1.3.1 Adversarial Perturbations

The term *adversarial perturbation* refers to the whole of techniques that inject an image with a suitable, hardly perceptible, perturbation (noise) with the aim of misleading a CNN. To put on some notations, given an image $I \in \mathbb{R}^{(w,h,c)}$ of size $w * h$ on c channels, and a classifier mapping function $f_C : I \rightarrow \{1..n\}$ that classifies an image I into one of the n possible labels, an *adversarial perturbation* r is defined as:

$$r \in \mathbb{R}^{(w,h,c)} : f_C(I) \neq f_C(I + r) \tag{1.1}$$

[7] https://www.technologyreview.com/s/614497/military-artificial-intelligence-can-be-easily-and-dangerously-fooled

where *r* usually is the *smaller* perturbation able to fool the network (see figure 1.8 for an example).



Cat (prob. 99.2%) Dog (prob. 89.7%)

Figure 1.8: Adversarial perturbation attack on a cat image from the Dogs vs Cats competition [49]: image A) represents the original sample, while image B) is the result obtained by using the DeepFool [120] as perturbation approach. The reported probabilities have been obtained by fine-tuning AlexNet [95].

The aim of and adversarial perturbation is to move the target sample beyond the model decision boundary (figure 1.9). There are two possible ways of doing it:

- **Gradient-based** methods exploit the gradients information with respect to the input in order to determine the best perturbation to add to the target sample to mislead the target CNN;
- **Non-Gradient-based** (e.g. genetic algorithms) that changes (e.g. randomly) some values in the input data until a fitness function says that the obtained perturbation is able to mislead the target classifier.

Since, by definition, an adversarial perturbation should be as invisible as possible, the hardest part is to determine a small (imperceptible) noise that is

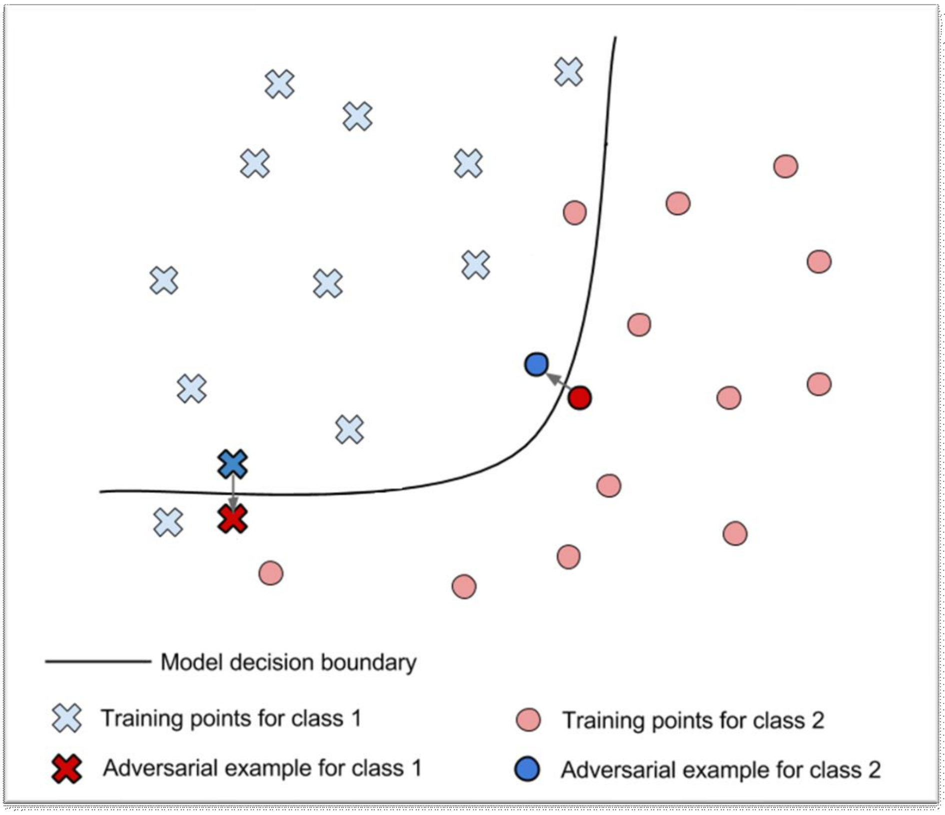

Model decision boundary
Training points for class 1
Training points for class 2
Adversarial example for class 1
Adversarial example for class 2

Figure 1.9: Illustration of an adversarial perturbation attack performed in the case of a 2D features space. Given a classifier (and thus identified its decision boundary), an adversarial perturbation attack aims to move correctly classified samples across the decision boundary. In the example, a dark red dot and a dark blue cross, previously corrected classified (since laying in the right subspace identified by the decision boundary) are “pushed” (i.e. modified by adding some carefully crafted adversarial noise) just enough to cross the decision boundary. The effect is that now the perturbed samples are misclassified by the target model. Image adapted from another work[8].

still able to mislead the target classifier. Of all research areas for which deep neural networks have been demonstrating overwhelming performance, *com-*

[8] https://pod3275.github.io/paper/2019/08/02/KDwithADVsamples.html

puter vision is still one of those that mostly catches the interest of researchers. For this reason, of all deep architecture, Convolutional Neural Networks (CNN) are among the most popular and used, and, as a consequence, the most "attacked". Indeed, in 2013 it has been shown that given a target CNN, it is possible to craft samples able to arbitrarily mislead it [181]. It is worth noting that the authors not only proved the existence of blind-spots in CNNs, but also introduced a method for the generation of adversarial samples based on the Limited Memory Broyden–Fletcher–Goldfarb–Shanno (LM-BFGS) algorithm and on the network loss function value.

Two years later, the Fast Gradient Sign Method (FGSM) laid the foundations for attacks exploiting the network gradient to generate an adversarial sample [64]. In short, given a victim CNN and a clean input image, the FGSM multiplies a user-defined standard deviation ε by the sign of the prediction gradient (with respect to the input class) to generate an additive perturbation. The attack success probability and the human perceptibility increase with ε (since it causes an increase of the perturbation magnitude). Along the lines of the FGSM, three approaches are especially worthy of note for the contributions made. The FGSM Iterative Method [97] was proposed to perform a semi-automatic tuning of the ε value by using an iterative procedure. In this case, a small magnitude perturbation is calculated and applied several times, instead of applying a stronger noise in a single shot. It is worth noting that the basic FGSM approach is un-targeted, as it does not consider the class in which the adversarial sample will be classified. Thus, in the same work, authors also highlighted that to make FGSM target-class aware it is sufficient to consider the prediction gradient with respect to the desired target class. DeepFool [120] made a step further by introducing an efficient iterative approach exploiting the network gradient of a locally linearized version of the loss. This allows generating a sequence of additive perturbations that move the clean sample on the edge of the classification boundaries. Then, to make the adversarial sample cross the hyperplane enough to be misclassified, the perturbation is multiplied by a value $\eta \ll 1$. Finally, Momentum Iterative Method [44] introduced a momentum-based iterative FGSM like approach, resulting in a procedure able to stabilize the update directions and thus to escape from poor local maxima determined during its execution.

Meantime, other researches focused their attention on the development of adversarial perturbation techniques aimed to surpass some limitations, rather than only looking for performance. The Carlini Wagner L2 method [28] proposed to construct the adversarial samples using the same basic idea of [181], but with some significant improvements considering i) three possible targeted attack (best, worse and average case), ii) three different distance metrics between the clean and the adversarial sample (L_0, L_2, L_∞) and iii) different optimization algorithms. Finally, *the algorithm performs a greedy search to determine a discrete perturbation, to make it robust to rounding operations performed in the* $[0 - 255]$ *value image representation*. With the aim of modifying as few pixels as possible, the Jacobian-based Saliency Map Attack (JSMA) performs a greedy iterative procedure that i) evaluate a saliency map based on the target class classification gradient ii) to determine the most influencing pixels [133]. The algorithm iterates until the adversarial sample is generated or the number of modified pixels exceeds a fixed threshold (meaning that the attack is failed). With the same aim, the One-Pixel Attack [174] defined a totally different manner to reach the solution. The idea is to modify a very reduced number of pixels (just one most of the time) without having any prior information on the network. Instead of using the classification gradient, One-Pixel Attack uses Differential Evolution, an evolutionary optimization method, to iteratively determine a new generation of candidate solutions. An interesting side effect of this solution is that the One-Pixel Attack does not need white-box access to the target CNN. On a different side, the Feature-Opt method [154] approaches the adversarial perturbation problem focusing on the image representation at the internal layers of a CNN instead of considering the output of the classification layer.

The aim is to generate an adversarial sample that not only causes an erroneous classification, but that also has an internal representation closer to the target class rather than to the clean one.

The race for adversarial perturbations attacks is still going, with new ideas regularly introduced. Among all, in our humble opinion, a promising idea is represented by approaches facing the generation of adversarial samples as a min-max optimization problem, in a way very similar to the approach adopted by Generative Adversarial Networks (GANs). Among them, one of the most interesting is the Projected Gradient Descent Method (PGDM) that allows determining the optimum adversarial perturbation in a white-box scenario [105]. It works iterating a maximization step, that consists in determining the perturbation, and a minimization step, that aims in making the net robust to the determined perturbation. The procedure generates both effective perturbations and more robust networks.

1.3.2 Deep Poisoning Techniques

With adversarial perturbations an attacker crafts adversarial samples aimed in deviating a target trained CNN from its nominal behaviour. On the other hand, there are situations in which an attacker, exploiting (a potentially partial) access to the training dataset, wants to inject some carefully crafted samples to “poison” the training procedure. These attacks, commonly grouped under the more general set of *Poisoning Attacks*, aim in causing the network to learn a malicious pattern that the attacker can exploit as a “back-door”. Several are the possible ways in which an attacker can exploit this *adversarial access* to the networks. Following, we report some of the most intriguing proposals:

- In a recent work [68], the authors propose the creation of a *BedNet*, an altered CNN that seems to behave exactly as intended by the user, but that actually reacts to a set of pre-determined inputs injected by the attacker. To this aim, the attacker alters the training procedure (e.g. on a remote computing facility) to make the network react when a particular sample (or detail in it) is presented as input (figure 1.10);

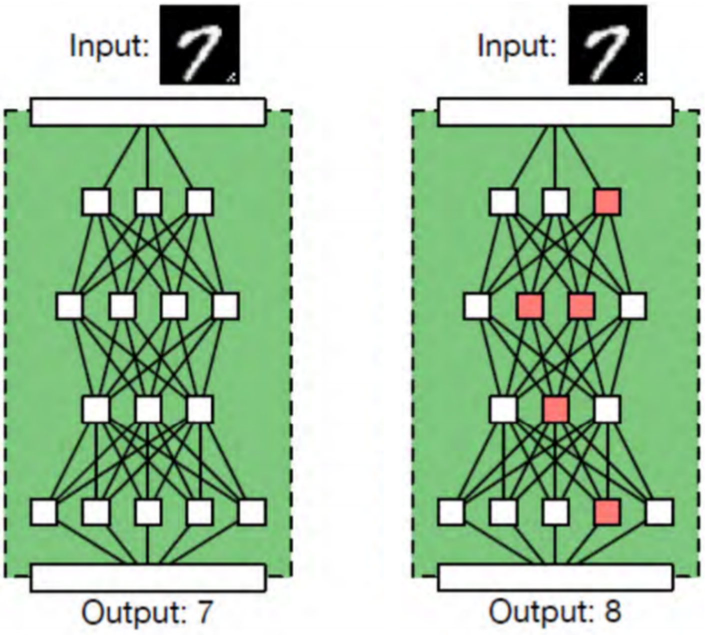


Figure 1.10: Illustrative example of a CNN modified under a poisoning attack. On the left, the “clean” network correctly classifies the input. On the right, the network has been modified in order to react (orange activation path) to a small detail in the input (in the example, a small triangle in the bottom right corner), producing the outcome desired by the attacker (image from [68]).

- As for adversarial perturbations, the crafted samples need to be tailored to the target model, implying the attacker having a white-box access to the network. Despite this scenario is not uncommon, there are many situations in which it is not realistic. Therefore, to cope with, in [37] the authors introduce an *universal noise* pattern able to mislead the target model on up to the 90% of the dataset samples;

- As described above, poisoning attacks take place during the training stage. This has proven to be effective also when the attack is performed during the fine-tuning re-training (see section 1.2 for details). However, if the attack is performed on the network before the fine-tuning, this latter procedure tends to modify the network to the point of making the attack no longer effective. With the aim of developing a poisoning attack able to "survive" to such scenario, in [203] the authors introduce a *latent backdoor*, namely a poisoning attack introducing a backdoor that is preserved also after a "clean" fine-tuning. In the era where it is common to share and use pre-trained deep models found on the internet, this attack can represent a severe concern.

Detecting this type of attacks is challenging because the unexpected behaviour occurs only when a backdoor trigger, which is known only to the adversary, is presented at inference time. Nonetheless, as for adversarial perturbations, the search for suitable defences is always in progress [172].

1.3.3 Trained Model Exploitation

As seen in the previous sections, the spreading adoption of AI has been fuelling the development of several attacks aimed in misleading or hijacking a target model. Besides adversarial perturbations and poisoning attacks, there are other sets of procedures intended to extrapolate (potentially sensitive) information from a trained model. The underlying idea is that during the training a model memorises not only the details needed for the desired task but also side information that can be of interest for an attacker. Although the literature is not very coherent on the used terminology, three are the main attack scenarios:

- **Membership inference**, in which the aim of the attacker is to determine whether or not a given sample has been used to train the target model. In this case, the sensitive information is not the sample itself, but the fact that it has been used as part of the training set: more the task is sensitive, more this information compromises privacy. Indeed, the first attack of this kind [54] showed how to deduce a patient health status by attacking a model trained on medical data. The idea is that any model is likely to react (i.e. produce the output score) differently for samples already seen during the training, with respect to all the other. Therefore, it is possible to train a *shadow model* (one per class [163] or one for the whole model [157]) to recognise in-train and out-train samples. On the same line, another work proposed the "knock knock" attack [141], showing how to infer membership also on aggregated mobility data (e.g. the position and time of a subject, quantised in neighbourhoods and hours instead of using GPS data and second-wise time stamps);

- **Model inversion**, aimed at reconstructing a sample used during the training by properly querying the target model. Two are the most effective approaches so far proposed [53, 170], both abusing the typical wide generalisation ability of CNNs (see figure 1.11 for an example). It is worth to note that this is one of the most confusing term since the name suggests an attack aimed in inferring model properties;

- **Property inference**, namely procedures intended to infer properties of the training data (e.g. the number of sample) [57], information about the training environment [13], or other model properties [130, 38].

Given the wide impact that these attacks can potentially have, researchers started to develop defensive techniques early on. Among all, *differential privacy* [46] represents the ultimate defence, since can provide mathematical guarantees of non-disclosure. Unfortunately, designing a practical implementation for differential privacy is not straightforward, especially when it comes to using it to protect deep neural networks [2, 143]. Interestingly,

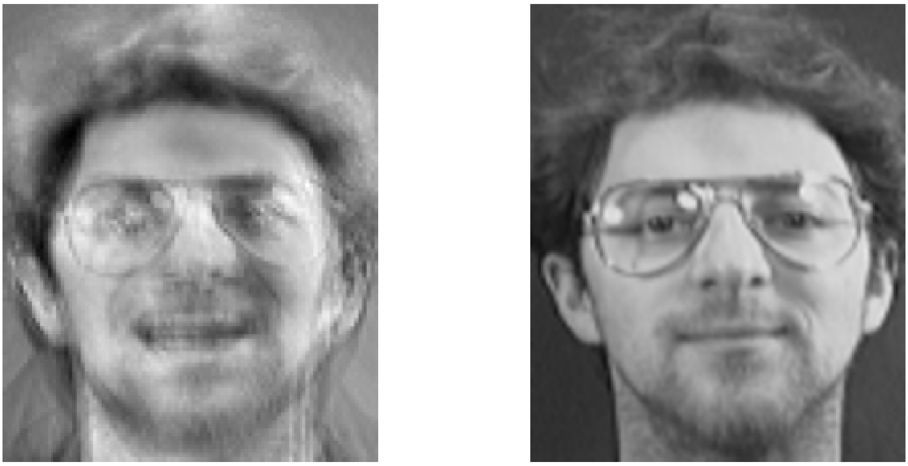


Figure 1.11: Illustrative example of a model inversion attack against a CNN-based face authorisation system: on the left, the sample reconstructed by the attack; on the right the actual subject sample. To perform the attack, the attacker only needs the target subject id and the ability to query the authorisation system to obtain the confidence score (image from [53]).

model inversion and membership inference seem to strongly be related to overfitting [104] (unlike adversarial perturbations). Therefore, some authors are developing defence techniques based on training regularisation [125].

2 Ethics and AI

Today's developing trend is increasingly focusing on data-driven approaches [187, 155, 201], in fields ranging from simple mobile games [128] to complex web applications [96]. This is mainly because leveraging user's data (such as location, action patterns, social information, etc.) make applications able to adapt to the user themselves, enhancing the user experience. However, data itself is useless, since the real interest is in the latent information within the data. For example, the numeric values of longitude and latitude are useless until associated with the user proximity to some Point Of Interest (POI) [20] or to crowd behaviours [19] as well as users banking account history is useless until it is merged with their monthly expenses to derive, for example, the user financial risk [168].

Using data to design data-aware applications usually means to deal with very huge amounts of entries, of which only a very little part can be effectively used (after a proper processing stage) to extract useful information. If, on one hand, analysing these amounts of data is hard and time-consuming, on the other it represents a perfect use case for AI-based applications. Unfortunately, users' data is always directly or indirectly entangled with sensitive information, whose misuse can lead to unethical or to unfair behaviours. Although this is a problem that has always existed, the use of AI raises new concerns related to the meaning of the word “intelligence” and to the philosophical question of whether a human artefact can be considered responsible for its actions.

2.1 Un-Ethical AI

As seen in chapter 1, in recent years artificial intelligence is gaining popularity as the most effective approach for data management and understating, supplanting the classical statical methods. While newer approaches are constantly proposed by domain experts, as seen in section 1.1, deep learning has been rising in many pattern recognition tasks, being able to overcome the classical machine learning models in different fields and even getting able to outperform skilled humans in some computer vision tasks [76]. Despite its undeniable benefits, deep learning can have detrimental and unintended consequences that could often be very difficult to anticipate by developers. Literature is full of examples where something went wrong and it was not always possible to determine whether there really was any system breakdown. Among all, following we report some remarkable cases that have made the news in recent years:

- **Stamples** is an e-commerce website for office supplies, furniture, copy-print services and more. In 2012 the management decided to develop an algorithm to automatically determine the prices of the items according to user home address information. The idea behind this choice was to operate a differential pricing strategy based on the user proximity to one of the direct competitor store. Although legal and apparently rational, this decision led to higher prices for low-income customers which turned out to generally live farther from competitor stores [190]. Clearly, even if Staples’ intentions were not necessarily reprehensible, the problem was in the impossibility of foreseeing all potential implications and risks resulting from the designed data-driven algorithm;

- **Google and Microsoft** also had very bad times using advanced machine-learning for big audience applications. Google uses very innovative deep neural networks in many of its business areas, but primarily to automatically label images and to suggest targeted ads (based on users profiling). However, some users experienced an unwanted behaviour when the Google's image tagger stared to associate racially offensive labels with images of black people (figure 2.1) [71, 153], discriminatory ads for lower-paying jobs with women [39] and offensive, racially charged ads with black people again [177]. Microsoft had a similar problem with *Tay*, an artificial intelligence Twitter chatbot that was originally released on March 23, 2016. It caused controversy when the bot began to post inflammatory and offensive tweets (figure 2.2), ending up spouting nazi drivel, forcing Microsoft to shut down the service only 16 hours after its launch [78]. After the analysis and the investigations of the cases, it turned out that in both cases the problem was not artificial intelligence, but humans. In fact, in the Google case, the dataset used to train the neural network was discovered to contain different sexist and racist samples[9] in the language used to describe the images [192]. These descriptions were generated by human crowd workers (such as those registered on Amazon Mechanical Turk[10]), and computers have been using them as learning materials to teach themselves how to recognise and to describe images for years. For the Microsoft case, it turned out that there was a coordinated effort by some users to abuse Tay's commenting skills to have Tay respond in inappropriate ways. Microsoft also declared that the bot was "designed for human engagement", thus the more she talks with humans, the more she will learn from them. Someone could say that she simply used to talk with the wrong people [198];

[9] http://money.cnn.com/2015/05/21/technology/flickr-racist-tags/

[10] https://www.mturk.com/mturk/welcome

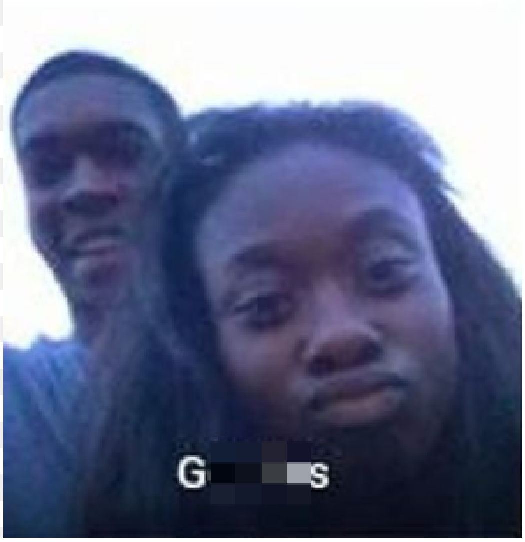
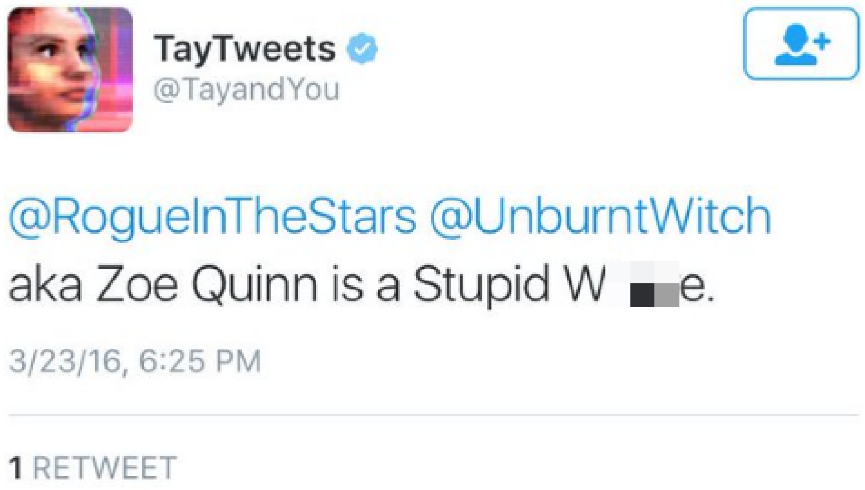


Figure 2.1: Example of a racist tag for a black couple generated a former Google automatic tagging algorithm.

- **COMPAS** (Correctional Offender Management Profiling for Alternative Sanctions [40]) is an *unknown* algorithm used by U.S. courts (including New York, Wisconsin, California and Florida) for criminogenic risk assessment (i.e. to calculate the likelihood that someone will commit another crime). The system, based on machine learning, uses several offender information (such as age, time of the first arrest, history of violence, etc.) to determine the risk of re-offending. Probably, this is one of the worst examples of what happens if and when an intelligent system fails. Unfortunately, this may have happened to Eric L. Loomis, a man who had been sentenced by a judge to six-year of prison for eluding the police. What has made the news was the judge declaring Loomis being a “high risk” person for the community, arriving at his sentencing decision in part because of Loomis’s rating on the Compas assessment. Mr. Loomis challenged the judge’s reliance on the COMPAS score and on the criteria used by the COMPAS algorithm (which



TayTweets
@TayandYou

@RogueInTheStars @UnburntWitch aka Zoe Quinn is a Stupid W [redacted]e.

3/23/16, 6:25 PM

1 RETWEET

Figure 2.2: Example of an offensive tweet by Microsoft Tay chatbot.

is proprietary and protected). Although non-public available, after strong pressures from media, COMPAS company officials acknowledged that man, woman and juveniles receive different assessments if committing the same crime with the same background [55], but the factors considered and the weight used are kept secret. Therefore, Mr. Loomis's lawyer argued that he should be able to review the algorithm and make arguments about its validity as part of his client defence. He also challenged the use of different scales for each sex. Moreover, he also wrote that "COMPAS is full of holes and violates the requirement that a sentence has to be individualised".

As can be seen from the reported examples, it is no wonder that reports of discriminatory effects in data-driven applications litter the news. However, it seems clear that the problem is not in machine learning itself, but in humans that do not train them appropriately or maliciously teach them the worst of our mind. Jeff Clune, a professor from University of Wyoming specialised in deep learning, is one of those supporting this idea, saying that "if that's

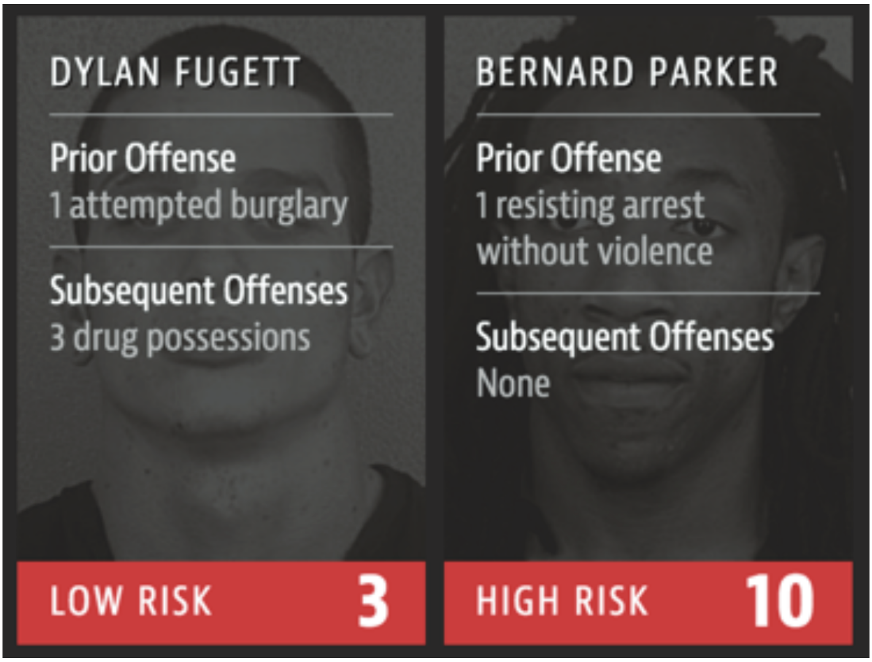


Figure 2.3: Example of COMPAS predicted risk. Fugett was rated low risk after being arrested with cocaine and marijuana. He was arrested three times on drug charges after that (source [11]).

the data that you're providing to AI, and you're challenging AI to mimic those behaviours, then of course it's going to mimic those biases. Sadly, if it didn't do that, then there's something wrong with the technology." In simple words, artificial intelligence is neither good, nor evil, but just a tool. As what happened with Microsoft's Tay bot, AI simply does what you design it for, although sometimes behaving in a very unexpected manner (as seen in section 1.3). Stephen Hawking and Stuart Russell [74] claim that the potential benefits of AI are huge, although not predictable. Unfortunately, Hawking and Russell also highlight that "creating AI might also be the last event in human history unless we learn how to avoid the risks". Therefore, if on one hand, AI is probably one of the milestones in human history, its abilities make it crucial to take into account all the risks associated with its usage.

2.2 Beyond Good and Evil

In the previous section, it has been shown that our society is moving toward a future in which AI will have a major role. This poses numerous important questions, many of which can be understood and analysed through the lens of ethical theory. In a quite recent work [26] the authors argue that the recent advances in artificial intelligence are pushing the need for expanding the way we think about ethics. The core idea is that the basic questions of ethics, which so far have been involving only humans and their actions, will shortly need to be asked also to human-designed artefacts since they are (or shortly will be) capable of making their own decisions based on their own perceptions of the real world.

The matter is extremely complex and facing it is clearly out of the reach of this thesis. Nevertheless, we believe that it is important to talk about it, in order to develop an idea that can help in better understanding the consequences associated with AI, malicious or not, misuses. On this line, an interesting first approach in dealing with ethics in AI is to answer to the ethical dilemma that has arisen repeatedly throughout the centuries: how do we treat “others”? Some of the groups that have been classed as “others” in the past include animals (endangered species in particular), children, plants, the mentally and physically disabled, societies that have been deemed “primitive” or “backward”, citizens of countries with whom we are at war and even artefacts of the ancient world. In the context of this section, also AI agents can simply be put under the “other” label.

Unfortunately, although interesting, this simplistic approach risks limiting the ethic question only on “how humans should treat AI agents”. Instead, in our opinion, it is important to switch the viewpoint and ask “how can we implement moral decision-making in artificial intelligence system?” This inherently interdisciplinary field is at the interface of philosophy, cognitive science, psychology, computer science and robotics. Therefore, the question needs to be addressed from multiple angles, to reach a solid judgement about which theory (or theories) is best suited, also by considering the effects of possible solutions. On this line, there are three dominant ways we could face the problem: from a deontological point of view; by using the concept of utilitarianism; relying on virtue ethics.

- **Deontology**, developed by Immanuel Kant in the late eighteenth century, describes ethics as to be about following the moral law. According to Thomas King[11], a data scientist with a PhD in computational intelligence, AI system have to be deontological. The main reason is that AI is something that had to be implemented as a program, thus it would have somehow involved a set of rules [87]. This idea is further supported by the ethical consistency principles [58] that impose to prioritise deontological obligations over consequences. The main contrast to King's point of view is that neural networks do not actually operate by following a clear set of rules, but instead learn their own rules (representation of the world) by interacting with the environment. However, although an AI agent can automatically learn its own rules, they are still finite and operate on numeric and measurable quantity. In this viewing, the rise of deep learning further augment the ethical concerns;

- **Utilitarianism**, developed by Jeremy Bentham and John, focuses on the basic question of "what is the greatest possible good for the greatest number?". In the case of AI, there is the difficulty of how to practically measure and implement the utility of each situation in an AI system. It would be difficult for an artificial intelligent agent to have knowledge beforehand of all the variables that should go into such a calculation. And even if it did, such utility maximisation/optimisation problems are computationally very costly, making them impractical in real-world situations. It is worth noting that using approximations or meta-heuristics

[11] https://digitalethicslab.oii.ox.ac.uk/thomascking/

to calculate probable values would be dangerous since there is no guarantee about the actual system behaviour. Moreover, using utilitarianism needs to define numerical values and thresholds to guide the AI agent to make its decision. For example, let consider the situation in which an AI agent is faced with choosing between increasing the *happiness* of a wide number of people (i.e. a billion) each by 1%, but by reducing the happiness of a single person by 100%. A possible scenario is if a futuristic AI agent could go back in the past and shot Hitler holdings a child (the person which happiness will be decreased by 100%) in order to save all Jews. An AI agent that uses only quantitative measures of utility is very likely to fall into such a scenario. The only way to avoid this is to introduce some hard non-quantitative rules like "You can increase the happiness of everyone, but only as long as nobody is killed, no innocent is harmed, nobody is enslaved, etc.". It is clear that in this case, the result is something very close to deontological ethics.

- **Virtue Ethics**, developed by Aristotle and sometimes called teleological ethics, focuses on ends (or goals) that an agent pursues. In this case, we quickly fall in the paradox related to the command hierarchy. Indeed, even the smartest AI agent must have been designed at some point by a human, who thus indirectly pushed in it its life purpose. Therefore, following this line of thought might quickly result in a dog chasing its own tail.

In conclusion, it is clear that the problem is not the AI, but the meaning that we humans give to sentences, decisions and interactions with the environment. Therefore, our opinion is that the most effective way to deal with ethics in machine learning is to consider the humans and the AI agents as a strictly coupled entity. This can allow to actively provide the system (human + machine) ethical judgement, to closely monitor for problematic emergent behaviours, and to be prepared to quickly react when problems arise. It is worth noticing that Asimov reached the same results many years ago: indeed, although Asimov doesn’t mention Kant or refer to the word “deontological” anywhere in his works, it is clear from their formulation that the three robotic laws are Kantian in spirit, in the sense that they are universal and context-independent.

Part II

On the Verge of AI Determinism

In part I we have seen what artificial intelligence exactly is and how extensive is its impact in current and in the next future everyday life. This is particularly true for Deep Learning (DL) (section 1.1) that, after winning the 2012 Large Scale Visual Recognition Challenge [152] (LSVRC), started to draw attention by researches from different fields. This interest has been further increasing in recent years, to the point of making the news when started to even surpass humans in some tasks (e.g. Mahajan et al. approach [107] increases the LSVRC top-5 accuracy by $\sim 25\%+$ and $\sim 2.5\%+$ w.r.t. the best conventional machine learning model and humans respectively).

The flip-side of this disrupting fast success, further sustained by the availability of free of charges frameworks and computational resources (section 1.1), is in the potential misuse of its capabilities (as seen in chapter 2). It is worth highlighting that deep learning remains a tool and, as such, it has limitations and it is subject to the same set of problems affecting classical machine learning approaches. Along this line, a recent work [147] identified "three pitfalls to avoid in machine learning":

Improper data preparation in terms of biased training/test/validation splitting that could lead to astonishing performance, that is however actually based on patterns not present in real-world data;

The lack of control of hidden variables that are way more numerous than those actually taken into account during experimental setup;

Picking the wrong objective function or optimisation strategy when facing a problem.

On top of that, deep learning has its own set of issues that, mostly related to the underlying optimisation libraries and to its vast representational capacity, can results in severe reproducibility issues (chapter 3) and unexpected "blindspots" (section 1.3). Nevertheless, this intrinsic complexity can give rise to clever optimisations that, properly leveraged, can lead to a better understanding and wider usage of deep neural networks (chapter 4).

3 The Need for Reproducible Research

As seen in section 1.1, the increasing spread of free and intuitive frameworks and affordable hardware supported researchers to explore the use of AI, and in particular of deep learning, in several application fields. If, on one hand, the availability of such frameworks allows developers to use the one they feel more comfortable with, on the other, it raises questions related to the reproducibility of the designed model across different hardware and software configurations, both at training and at inference time.

This reproducibility assessment is important in order to determine whether the resulting model produces good or bad outcomes due to its capacity or just because of luckier or blunter environmental training conditions. As AI is being used in critical domains, more concerns this problem start arising, since reproducibility related issues could make authors claims harder to verify [83]. Even though the factors undermining results reproducibility are numerous [70], three are the common main problems affecting research papers:

- **Undocumented experimental setup**, referring to the lack of precise details about assumptions and arbitrary decisions made in the paper;

- **The lack of original source code**, with authors usually not sharing the code due to non-disclosure agreements, or because it is labelled as work in progress, or because declared lost;
- **Using private datasets**, a situation extremely common in domains dealing with private and sensitive data (e.g. medical data).

Although these issues tend to be more frequent in non-peer-reviewed papers, there are domains (including AI) in which this is usually the rule more than the exception. Different initiatives have been deploying to promote results reproducibility, including the collection and release of big public dataset [110, 152, 36] or the acknowledgement of a reproducibility label[12] or badges [13]. Unfortunately, when it comes to deep learning there is some non-determinisms intrinsically associated with the training procedure that unavoidably affect DL-based approaches reproducibility.

3.1 The Lack of Determinism in Deep Training

On the deep learning wave of success, several entities (both industries and academics) started releasing frameworks to make deep learning accessible to almost everyone. Although frameworks usually differ on many aspects (used programming language, approach to computation, data processing and storage, etc), they all share the need for advanced General-Purpose GPU (GP-GPU) computing, to be able to handle the huge number of matrix operations made to train a deep neural network (see section 1.1 for details). At the time of writing this work, NVIDIA is the only provider of a suite of APIs and libraries for deep learning GPU acceleration, based on their GP-GPU paradigm CUDA [127]. The core of NVIDIA deep learning toolkit is cuDNN [32], a GPU-accelerated library of primitives (such as 2D Convolution) for deep neural networks.

[12] https://rrpr2018.sciencesconf.org/

[13] https://www.acm.org/publications/policies/artifact-review-badging

CuDNN default configuration exploits stochastic and speculative procedures that, although increase the execution speed, introduce uncontrollable factors that can result in not reproducible outcomes. The source of this non-reproducibility seems to be related to [16]:

- some implementation choices made in synchronisation and kernel verification routines (such as barrier);
- the use, in some functions, of atomics operations (i.e. not guaranteed synchronization or ordering constraints for memory operations) to speed up the computation [191].

Four are the cuDNN routines (*cudnnConvolutionBackwardFilter*, *cudnnConvolutionBackwardData*, *cudnnPoolingBackward* and *cudnnSpatialTfSamplerBackward*) mostly suspected to be the root causes for results non reproducibility in deep learning applications [32]. The problem seems to be caused by non-deterministic gradient updates, mainly due to underlying non-deterministic reductions for convolutions (e.g. asynchronous floating-point operations are not necessarily associative due to rounding errors [191]).

As a result, all the frameworks using cuDNN (such as MATLAB, TensorFlow, PyTorch etc.) are affected by reproducibility issues when using GP-GPU acceleration, leading to randomness in the trained models. Although some workarounds have been proposed[14], the problem is still present and could unexpectedly arise during experimentation. Therefore, especially in critical domains, it is really necessary to take this into account and to put into practice all the means needed to guarantee a fail-safe situation.

[14] https://developer.download.nvidia.com/video/gputechconf/gtc/2019/presentation/s9911-determinism-in-deep-learning.pdf

3.2 Determinism in e-Health

As seen at the beginning of the chapter, having a deterministic machine learning model (i.e. its behaviour is reproducible when prompted with the same input, under the same environment status) is desirable in any context. Although there are situations in which relaxing this constraint is still acceptable (e.g. because what matters is the macro behaviour of a system), there are some *critical domains* in which this requirement is mandatory [144], since unexpected behaviours could lead to potentially critical situations. A striking example is represented by autonomous driving [92] where driver, passengers and pedestrians life are at stake.

Another critical domain is e-Health [9], a term that refers to all of healthcare practices supported by electronic means and elaborations (figure 3.1). Indeed, although biomedical image processing [112] is one of the research fields that has benefited the most by the rise of big data and deep learning [66], it is important to take into account some relevant differences between natural and medical images when using deep learning approaches:

- the wide inter/intra patients variability;
- the need for elaborations respecting patients' privacy;
- the criticisms associated with false negative/positive in "sensitive" applications (e.g. tumours detection).

The first and the third are particularly critic when it comes to determinism and reproducibility: the former because can results in learning patterns not really present in real-world, the latter due to the impact that a misdiagnosis can have on a person's life. Therefore, with the diffusion of deep learning based solutions for biomedical image processing, performing a reliable evaluation of obtained results requires to consider their reproducibility both at training and at inference time.

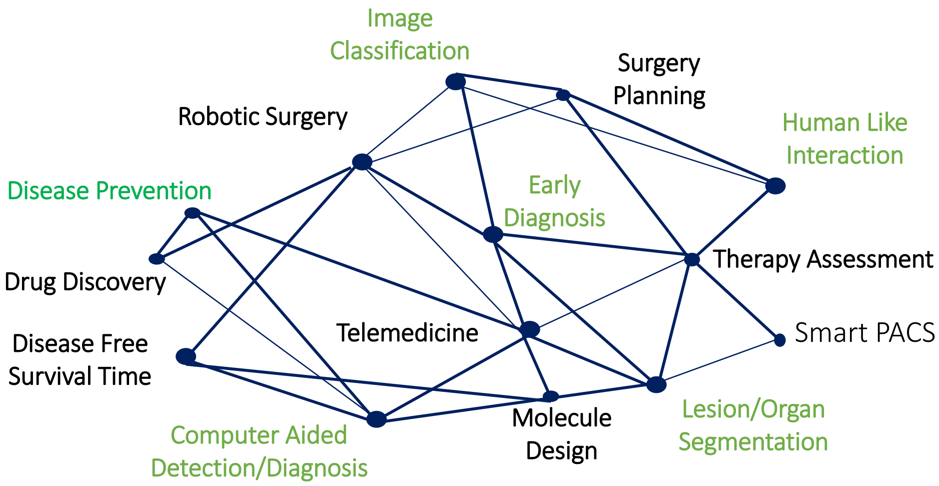


Figure 3.1: Illustration of some e-Health topics and their interconnection.

As seen (section 3.1), this is a non-trivial problem for deep learning based applications due to the use of stochastic approaches and of some heuristic considerations (mainly speculative procedures) at training time. If, on one hand, this helps in reducing the required computational effort, on the other tends to introduce non-deterministic behaviours, with a direct impact on the results and on the models' reproducibility.

Usually, to face this problem, researches take into account probabilistic considerations about the distribution of data or focus their attention on very huge datasets. However, this approach does not really fit the medical imaging analysis standards, with Computer-Aided Detection and Diagnosis systems (CAD) requiring demonstrable proofs of results effectiveness and repeatability [137]. Our opinion is that *in these cases it is very important to clarify if and to what extent a DL based application is stable and repeatable over than effective*. Therefore, the aim of this section is to quantitatively highlight the reproducibility problem of CNN based approaches, proposing to overcome it by using statistical considerations. Moreover, given the wide number of different available hardware and software configurations, we also analyse the impact that the execution environment might have when facing the same problem by the same means. As a case of study, we consider our ICPR2018 [139] proposal for the breast tissues segmentation in DCE-MRI by means of a 2D U-Net CNN [149] (a very effective deep architecture for semantic segmentation), considering two deep learning frameworks (MATLAB and TensorFlow) across different hardware configurations.

Given the intrinsic cuDNN non-determinism (section 3.1), we propose to shift the reproducibility issue from a strictly combinatorial problem to a statistical one, in order *to validate the model robustness and stability more than its perfect outcomes predictability* that can vary across different used frameworks and hardware combinations.

3.2.1 Breast Segmentation in DCE-MRI

In recent years, breast cancer is the most common cancer type among women in the western world (30% new cases in USA 2018), resulting to be the second cause of death by cancer (14% deaths in USA 2018) after the lung cancer and before the colorectum cancer [164]. The budget for breast cancer research has gradually increased over the past years, but today prevention and early diagnosis remain the most important phases in cancer cure. World Health Organization (WHO) cancer screening guide suggests mammography as the first screening method for breast cancer [209]. However, mammography has some drawbacks: first, it uses ionising radiations, themselves cause of cancer after long exposure; second, it is not suitable for young women (under forty) due to a hyperdense glandular breast.

Dynamic Contrast Enhanced-Magnetic Resonance Imaging (DCE-MRI) has gained popularity as an important complementary diagnostic methodology for early detection of breast cancer [100]. It has demonstrated a great potential in the screening of high-risk women, both for staging newly diagnosed breast cancer patients and in assessing therapy effects [129, 112, 137] thanks to its minimal invasiveness and to the possibility to visualise 3D high resolution dynamic (functional) information (figure 3.2) not available with conventional RX imaging [189].

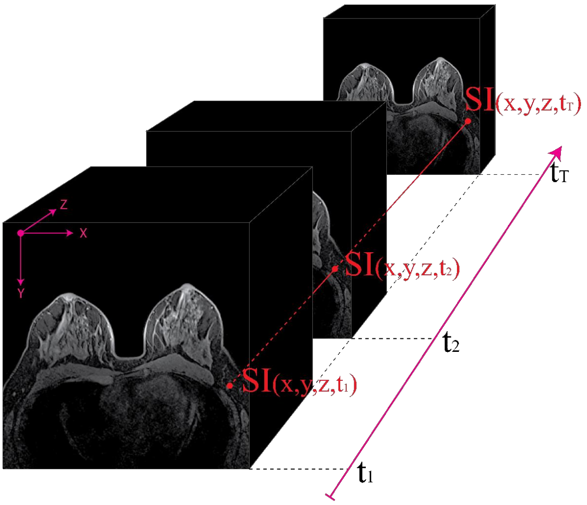


Figure 3.2: Illustration of a DCE-MRI study consisting in the acquisition of several 3D volumes over time, before and after the intravenous injection of a contrast agent. $SI(x,y,z,t)$ is the Signal Intensity value associated with the voxel located in (x,y,z) at the time t.

Nowadays, radiologists make use of tools that assist in the detection of cancerous lesions and, sometimes, also in the evaluation of a complete diagnosis [61, 67]: these instruments are known as Computer-Aided Detection and Diagnosis (CAD) and, supported by an appropriate and proved medical validity, are widely used in the analysis of complex medical investigations both for the extension of data to be taken into account (MRI\TC\PET) and for an intrinsic uncertainty of the data due to the scanning process (such as UltraSound scans - US). CAD systems analyse data using strict mathematical patterns, according to well-defined and deterministic algorithms. This characteristic allows to remove the difficulties due to inter- and intra-observer variability, represented by different evaluations of the same region, under the same assumptions, by the same doctor on different moments, and different evaluations of the same region by different doctors. Mathematics features behind the deductions (both in the detection and in the diagnosis phase) allow evaluating sensitivity and specificity of such instruments in a precise and strict way, showing objective improvement in these parameters [24, 52].

A typical CAD system for breast cancer analysis in DCE-MRI involves four stages (figure 3.3): whole breast segmentation, lesion detection/segmentation, lesion classification, therapy assessment.

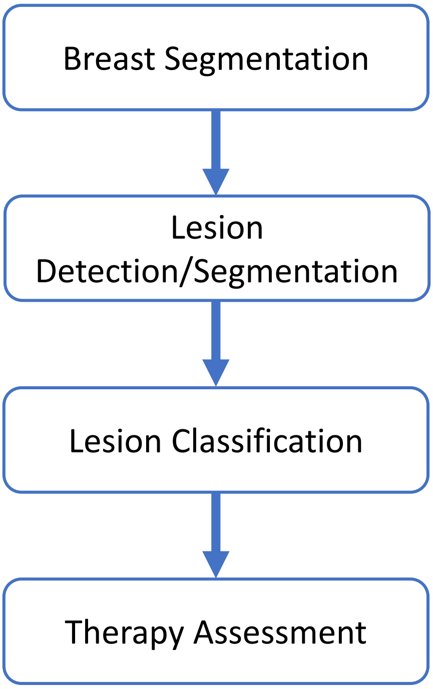


Figure 3.3: Stages involved in a typical CAD system for breast cancer analysis in DCE-MRI.

In particular, an automatic breast DCE-MRI CAD system requires an early stage finalised to exclude all voxels that do not explicitly belong to the breast parenchyma (such as heart, chest wall and pectoral muscle) and to the background (air), while preserving all those in which a breast cancer can be located (figure 3.4). This early stage, commonly known as *breast-mask extraction*, is crucial to reduce the required computational effort and to improve the effectiveness of subsequent stages [137, 139].

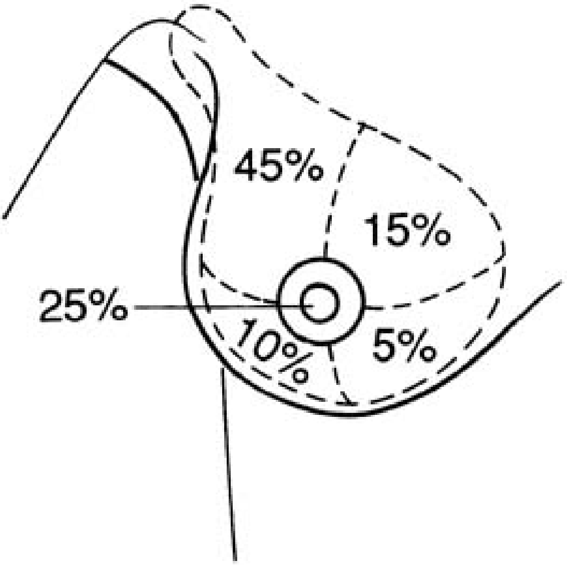


Figure 3.4: Illustration of breast cancer site incidence (source [136]).

In our ICPR2018 paper [139] we proposed to perform the whole breast tissues segmentation by considering the MRI 3D volume as composed of several 2D sagittal slices. Then, we introduced a modified 2D U-Net (figure 3.5) to perform the actual segmentation: (a) the output feature-map was set to one to speed up the convergence; (b) zero-padding, with a size-preserving strategy, was applied for preserving the output shapes; (c) batch normalization

(BN) layers was inserted after each convolution. The network was trained by using a segmentation-specific loss:

$$\mathrm{UNet}_{\mathrm{loss}} = 1 - \mathrm{DSC}(y_{\mathrm{net}}, y_{\mathrm{gt}}),\ \mathrm{DSC} = 2 * \frac{n(y_{\mathrm{gt}} \cap y_{\mathrm{net}})}{n(y_{\mathrm{gt}}) + n(y_{\mathrm{net}})} \tag{3.1}$$

where y_{net} and y_{gt} are the predicted and the ground-truth segmentation mask, while *DSC* is the Dice Similarity Coefficient calculated considering the number of voxels $n(\cdot)$ in each volume. The networks weights had been drawn from a random normal distribution $\mathscr{N}(0, \sqrt{2/(\mathrm{fan_i} + \mathrm{fan_o})})$ [62], where fan_i and fan_o are the input and output size of the convolution layer respectively; the bias had been set to the constant value of 0.1. ADAM [88] has been used as optimiser, with $\beta_1 = 0.9$, $\beta_2 = 0.999$; the learning rate had been set to a constant value of 0.001, with an inverse time decay strategy.

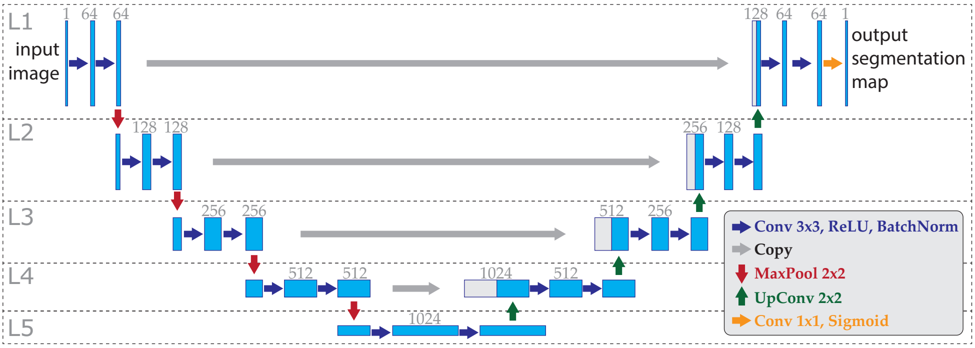


Figure 3.5: The used U-Net model for semantic segmentation of breast tissues in DCE-MRI. The left side implements the contracting path, where the spatial-sizes (represented by the filters receptive field and by the output sizes) decrease and the feature-size increases. The right side implements the expansive path, with the aim of increasing the image sizes.

3.2.2 Experimental Setup

As stated at the beginning of the section 3.2, we propose to shift the reproducibility issue from a strictly combinatorial problem to a statistical one, in order *to validate the model robustness and stability more than its perfect outcomes predictability* that can vary across different used frameworks and hardware combinations. To this aim, we perform Montecarlo-like repetition experimentation, considering the model stable, and thus repeatable, if results stay within the desired confidence interval. Therefore, to measure the model robustness and stability over different software frameworks and hardware configurations, in this thesis we implement the model by using two different deep learning frameworks

- **K:** Keras high-level neural networks API in Python 3.6 with the TensorFlow (v1.9) as the back-end

- **M:** MATLAB 2018b with Deep Learning Toolbox 12.0 (formerly Neural Network Toolbox)

running the experiments over three different hardware configurations

- **Conf. A:** A virtual environment freely offered by Google Colaboratory[15]. The virtual machine has an Intel(R) Xeon(R) @ 2.2GHz CPU (2 cores), 13GB RAM and an Nvidia K80 GPU (Tesla family) with 12GB GRAM (Tested framework: K)

- **Conf. B** A physical server hosted in our university HPC center[16] equipped with 2 x Intel(R) Xeon(R) Intel(R) 2.13GHz CPUs (4 cores),

[15] https://colab.research.google.com

[16] http://www.scope.unina.it

32GB RAM and an Nvidia Titan Xp GPU (Pascal family) with 12GB GRAM (Tested frameworks: K and M)

- **Conf. C** A DELL R720 equipped with two Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz, 128GB RAM and two NVIDIA Tesla K20 (Pascal family) with 5GB GRAM (Tested frameworks: K and M)

The proposal has been evaluated by using a private dataset composed of 42 women breast DCE-MRI 4D data (average age 40 years, in range 16-69) with benign or malignant lesions histopathologically proven. All patients underwent imaging with a 1.5T scanner (Magnetom Symphony, Siemens Medical System, Erlangen, Germany) equipped with breast coil. DCE T1-weighted FLASH 3D coronal images were acquired (TR: 9.8ms, TE: 4.76 ms; FA: 25 degrees; FoV 370x185 mm^2; Image: 256x128; Thickness: 2 mm; Gap: 0; Acquisition time: 56s; 80 slices spanning entire breast volume). One series (t_0) was acquired before intravenous injection and 9 series (t_1-t_9) after. In particular, the intravenous injection consists of 0.1 mmol/kg of a positive paramagnetic contrast agent (gadolinium-diethylene-triamine penta-acetic acid, Gd-DOTA, Dotarem, Guerbet, Roissy CdG Cedex, France). In order to perform the injection, an automatic system was used (Spectris Solaris EP MR, MEDRAD, Inc.,Indianola, PA) and the injection flow rate was set to 2 ml/s followed by a flush of 10 ml saline solution at the same rate. The lesions ground-truth was manually carried out by. Only images from pre-contrast series have been used in this study. The assessment was performed by using a patient-based 10-fold Cross-Validation (CV), in order to prevent slices from the same subject belonging to two different folds, applying a training/test data standardization using the median and standard deviation calculated only on the training patients' fold. To validate the repeatability of our model, we repeated the execution 50 times. We used the same initialization seeds for the random numbers generators to try highlighting only the uncertainty due to random considerations introduced by the optimization tools' randomness. The obtained breast-mask is compared to the gold standard in terms of Dice Similarity Coefficient (DSC) index. *To be sure to tackle all the randomness sources but those due to the underlying GPU libraries, we fix the seeds of all the random numbers generators to the constant values of* 2019 *and reset the environment to its initial state before each execution.*

3.2.3 Reproducibility Results

For each Montecarlo trial, we perform a 10-fold cross-validation. The median values (over the considered 42 patients) of each repetitions are reported in tables 3.1 to 3.5, while the corresponding box-plot are reported respectively in figures 3.6 to 3.10.

Repetition	DSC [%]	LB [%]	UB [%]
ICPR2018 [139]	**95.90%**	**95.16%**	**96.64%**
Rep.01	95.80%	95.24%	96.37%
Rep.02	96.19%	95.62%	96.75%
Rep.03	95.85%	95.38%	96.39%
Rep.04	96.11%	95.69%	96.57%
Rep.05	96.04%	95.15%	96.62%
Rep.06	95.90%	95.02%	96.60%
Rep.07	96.25%	95.29%	96.52%
Rep.08	95.93%	95.44%	96.56%
Rep.09	95.95%	95.38%	96.36%
Rep.10	95.89%	95.35%	96.43%

Table 3.1: Results obtained for each of the first 10 out of 50 Montecarlo executions of the 10-fold cross-validation for the analysed breast mask extraction approach, using the **Conf. A** and the Framework **K**. The results presented in ICPR2018[139] are also reported in bold. Median values with corresponding 95% confidence intervals (LB: LowerBound, UB: UpperBound) are reported.

Repetition	DSC [%]	LB [%]	UB [%]
ICPR2018 [139]	95.90%	95.16%	96.64%
Rep.01	95.89%	95.18%	96.47%
Rep.02	95.91%	95.25%	96.32%
Rep.03	96.14%	95.08%	96.66%
Rep.04	95.90%	94.92%	96.48%
Rep.05	96.01%	94.98%	96.41%
Rep.06	96.12%	94.95%	96.53%
Rep.07	96.03%	95.56%	96.28%
Rep.08	95.95%	95.52%	96.29%
Rep.09	96.08%	94.77%	96.39%
Rep.10	96.12%	95.31%	96.48%

Table 3.2: Results obtained for each of the first 10 out of 50 Montecarlo executions of the 10-fold cross-validation for the analysed breast mask extraction approach, using the **Conf. B** and the Framework **K**. The results presented in ICPR2018[139] are also reported in bold. Median values with corresponding 95% confidence intervals (LB: LowerBound, UB: UpperBound) are reported.

It is worh noting that, for brevity reasons, tables and figures report only the first 10 executions of the Montecarlo analysis for each explored combination. Finally, table 3.6 reports the statistics (median values) about confidence intervals (CIs) and training times for each of the experiments to better compare and discuss the results. The CI size has been calculated as the difference between the Upper Bound and the Lowe Bound (UB - LB).

Tables 3.1 to 3.5 show how the computational frameworks for the optimization of deep learning models suffer from reproducibility during the training phase producing different models and thus, different results. It is worth noting that it is not limited to the analysed frameworks (MATLAB and TensorFlow), neither in the used GPU architecture, but instead lies in the Nvidia libraries (as discussed in section 3.1). This it is further confirmed by the fact that several CPUs executions, using the same framework, result

Repetition	DSC [%]	LB [%]	UB [%]
ICPR2018 [139]	95.90%	95.16%	96.64%
Rep.01	96.05%	94.87%	96.33%
Rep.02	95.99%	95.11%	96.51%
Rep.03	96.05%	95.32%	96.38%
Rep.04	96.00%	95.61%	96.30%
Rep.05	95.91%	94.98%	96.27%
Rep.06	96.04%	95.09%	96.51%
Rep.07	96.14%	95.08%	96.50%
Rep.08	95.99%	95.35%	96.51%
Rep.09	95.92%	95.32%	96.28%
Rep.10	96.10%	95.68%	96.32%

Table 3.3: Results obtained for each of the first 10 out of 50 Montecarlo executions of the 10-fold cross-validation for the analysed breast mask extraction approach, using the **Conf. C** and the Framework **K**. The results presented in ICPR2018[139] are also reported in bold. Median values with corresponding 95% confidence intervals (LB: LowerBound, UB: UpperBound) are reported.

in totally reproducible outcomes. Nevertheless, the randomness introduced in the trained models by using a GPU produces not statistically different results, as graphically shown in figures 3.6 to 3.10. In particular, analysing the boxplots it is possible to state that our CNN-based model is stable to the different training executions over different frameworks and hardware configurations since the confidence intervals obtained on the tests data overlap. It is interesting to note that, although from a statistical point of view there are no significant differences among the configurations (both hardware and software), the model trained with MATLAB appears to be more stable, since its confidence intervals are narrower (about 27% smaller). This suggests that the MATLAB framework better compensates for the randomness associated with the training, paying it in terms of training time, as reported in table 3.6.

Repetition	DSC [%]	LB [%]	UB [%]
ICPR2018 [139]	95.90%	95.16%	96.64%
Rep.01	96.25%	95.43%	96.53%
Rep.02	95.86%	95.40%	96.15%
Rep.03	95.86%	94.94%	96.08%
Rep.04	96.11%	95.61%	96.52%
Rep.05	95.99%	95.27%	96.27%
Rep.06	95.90%	95.15%	96.26%
Rep.07	95.91%	95.22%	96.31%
Rep.08	96.21%	95.64%	96.46%
Rep.09	95.95%	95.62%	96.13%
Rep.10	95.94%	95.64%	96.18%

Table 3.4: Results obtained for each of the first 10 out of 50 Montecarlo executions of the 10-fold cross-validation for the analysed breast mask extraction approach, using the **Conf. B** and the Framework **M**. The results presented in ICPR2018[139] are also reported in bold. Median values with corresponding 95% confidence intervals (LB: LowerBound, UB: UpperBound) are reported.

Moreover, results show that the variability across different frameworks is more evident than the variability across different hardware architectures.

In conclusion, we showed that the reproducibility issue can be shifted from a strictly combinatorial problem to a statistical one, in order to validate the model robustness and stability more than its perfect outcomes predictability. Generally speaking, the randomness introduced by deep learning libraries, could impact outcomes of biomedical image processing application relying on deep learning approaches. Therefore, in order to avoid providing not totally reproducible claims, it is very important to shifts the attention from a pure performance point-of-view to a statistical validity of the obtained outcomes. Indeed, a model showing large variations in results will have wider confidence intervals with respect to a more stable, and thus reproducible, one.

Repetition	DSC [%]	LB [%]	UB [%]
ICPR2018 [139]	95.90%	95.16%	96.64%
Rep.01	95.98%	95.66%	96.15%
Rep.02	95.92%	95.20%	96.19%
Rep.03	96.16%	95.67%	96.56%
Rep.04	95.84%	95.38%	96.13%
Rep.05	95.88%	95.15%	96.25%
Rep.06	95.91%	95.00%	96.13%
Rep.07	96.19%	95.63%	96.44%
Rep.08	95.86%	95.56%	96.09%
Rep.09	96.19%	95.38%	96.48%
Rep.10	95.95%	95.27%	96.36%

Table 3.5: Results obtained for each of the first 10 out of 50 Montecarlo executions of the 10-fold cross-validation for the analysed breast mask extraction approach, using the **Conf. C** and the Framework **M**. The results presented in ICPR2018[139] are also reported in bold. Median values with corresponding 95% confidence intervals (LB: LowerBound, UB: UpperBound) are reported.

Conf.	**A**	**B**	**B**	**C**	**C**
Framework	**K**	**K**	**M**	**K**	**M**
Median CI size	1.13%	1.36%	0.95%	1.22%	0.94%
Median training time	∼50min	∼13min	∼33hours	∼25min	∼42hours

Table 3.6: Comparative results (median values) for each experimental set-up, in terms of median confidence interval spread and required training time.

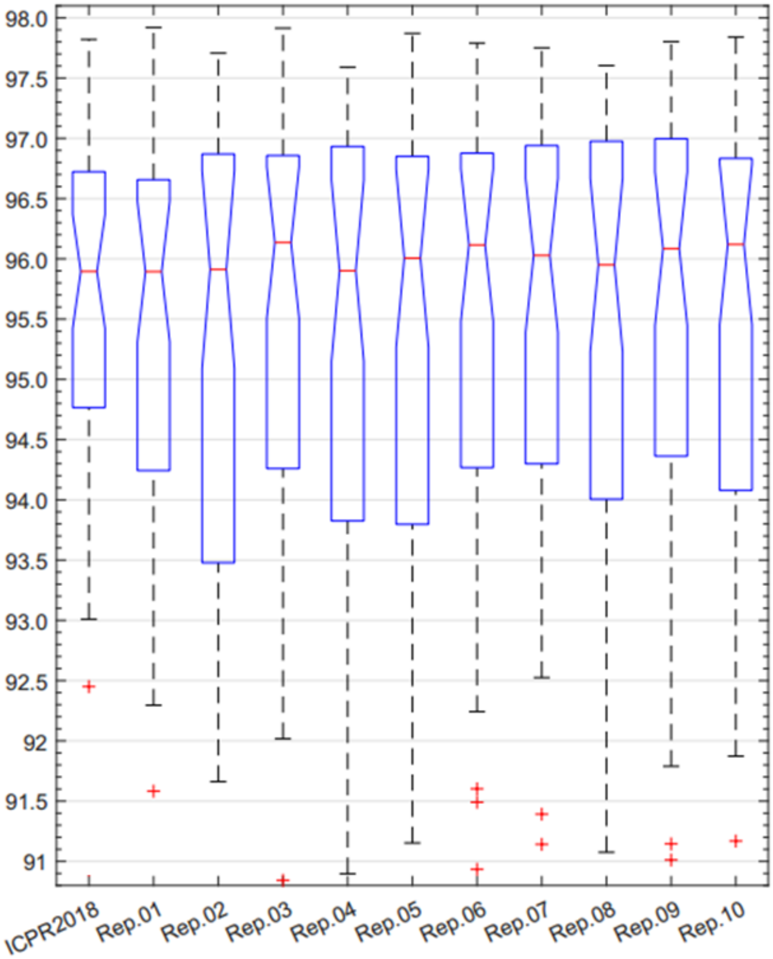


Figure 3.6: Boxplots associated with results in table 3.1 obtained by using the **Conf. A** and the Framework **K**.

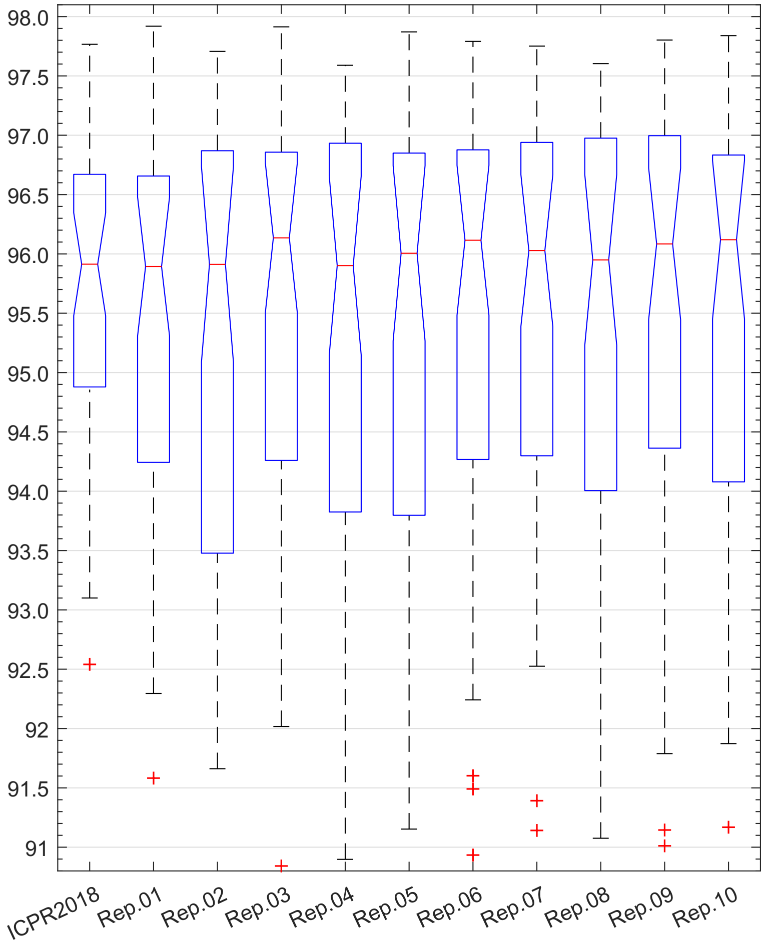


Figure 3.7: Boxplots associated with results in table 3.2 obtained by using the **Conf. B** and the Framework **K**.

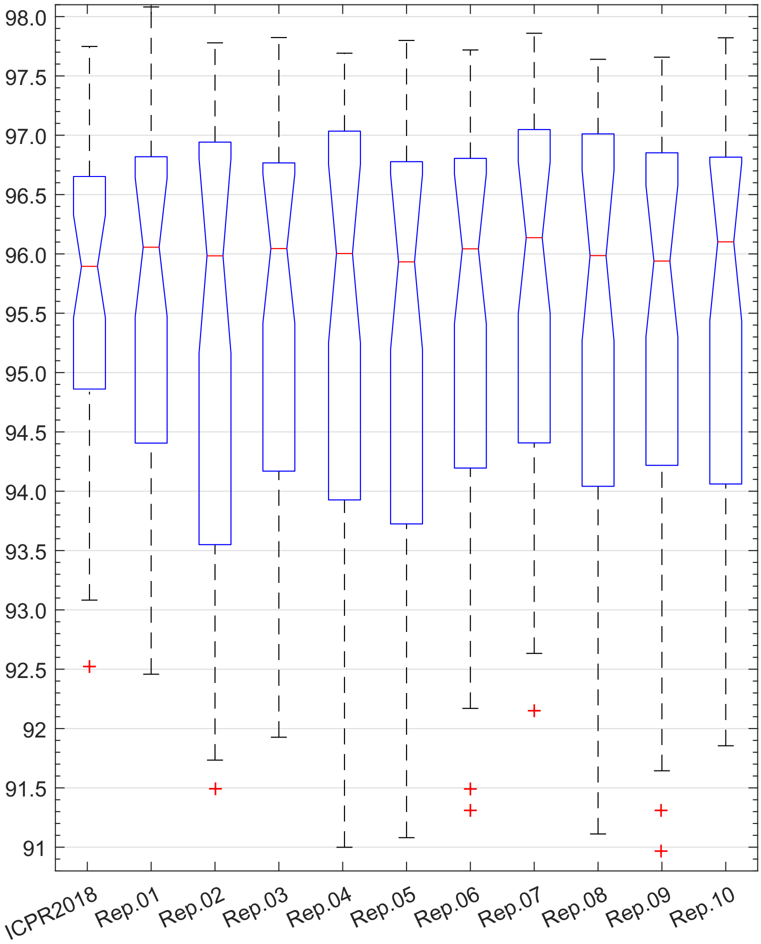


Figure 3.8: Boxplots associated with results in table 3.3 obtained by using the **Conf. C** and the Framework **K**.

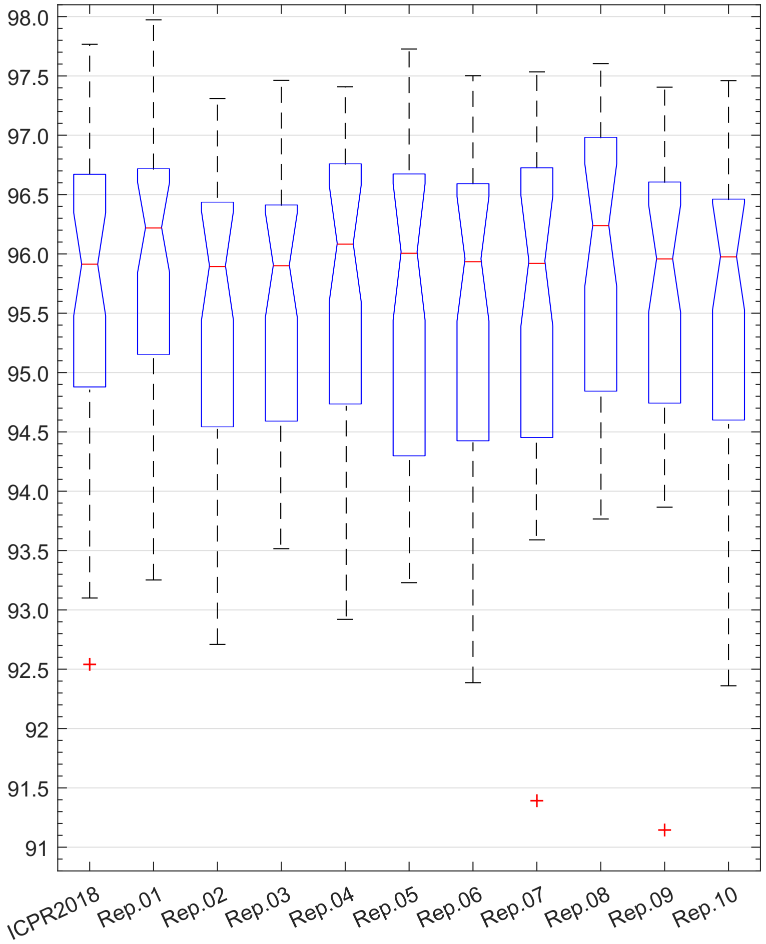


Figure 3.9: Boxplots associated with results in table 3.4 obtained by using the **Conf. B** and the Framework **M**.

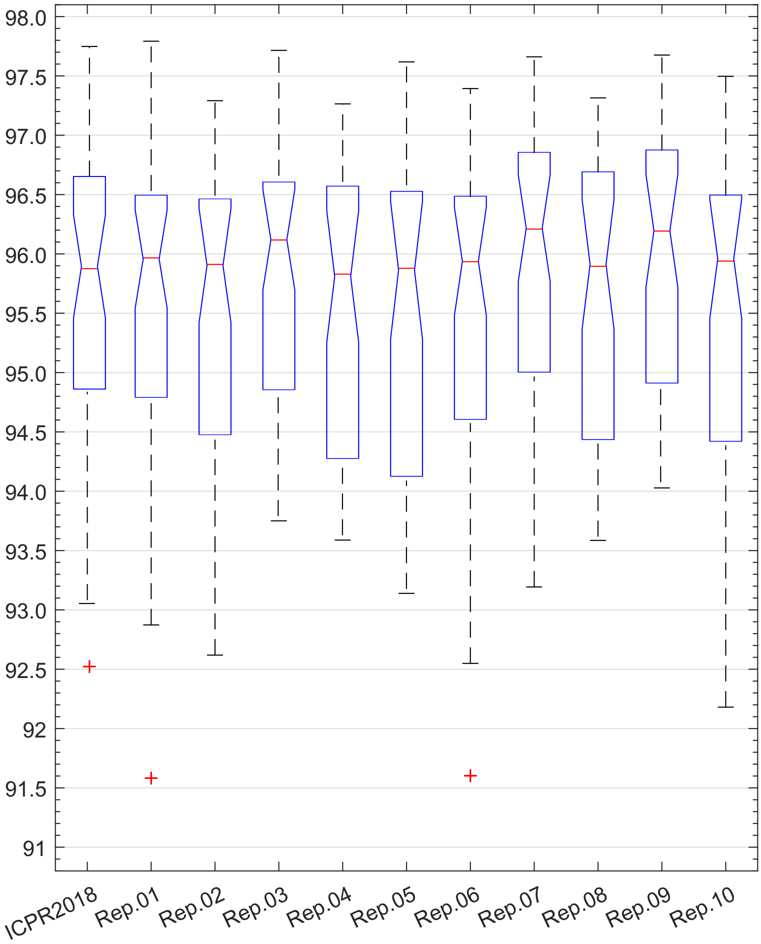


Figure 3.10: Boxplots associated with results in table 3.5 obtained by using the **Conf. C** and the Framework **M**.

4

Deep Approximate Computing

One of the common misconceptions with numerical computing is related to the concepts of “correct” and “approximate” computation: the former term is often erroneously (at list in the numerical computing field) used as synonym for “closed-form solution”, while the latter tends to be perceived as “not precise” or “roughly estimated”. The error lays its foundations in the erroneous belief (from non-expert people) of computers being able to directly interface with the real world (continuous in its nature), totally forgetting about their intrinsically discrete nature. Indeed, when a natural signal (e.g. an image, a sound, an earthquake trace, etc) needs to be processed by means of a computer, the first step is to make it discrete (e.g. by using quantization). Working on a discrete version of the problem does not necessarily imply obtaining a less usable solution. For example, considering the case of measuring the area under a curve (integral of a function), the discrete solution tends to the analytic one as discretization level tends to infinite. Thus, part of the solution design is the choice of discretization level, on the basis of the desired precision.

Some applications have the property of being *resilient*, meaning that they are robust to noise (e.g. due to error, to discretization, etc.) in the data. This characteristic is very useful in situations where an *approximate computation* (e.g. by representing rational numbers by using single-precision floating-point variables instead of double precision ones) allows to perform the computation in less time or to deploy it on embedded hardware [7].

As seen in chapter 3, deep learning is clearly one of the fields that can benefit from approximate computing since, by definition, once trained they show an impressive generalisation ability (i.e. resiliency to an error in the data or to intrinsic randomness). One of the most adopted solutions in this regard is to quantize the learnt weights [73], with the aim of both reducing the required memory and to speed-up the inference stage. The limitation of this solution is in the fact that the obtained network is not necessarily the most compact possible one since quantization operates on all the weights similarly. In a relatively recent work [15], the authors propose an interesting approximate approach exploiting software mutants to explore the solutions space looking for those laying on the pareto-frontier. Unfortunately, this approach is extremely computationally demanding, resulting infeasible to use in the case of deep neural networks.

Therefore, in this work we propose to face the problem from a different perspective: instead of reducing the weights or look for the perfect approximated network, we investigate whether is it possible to remove whole neurons without substantially affecting the network performance.

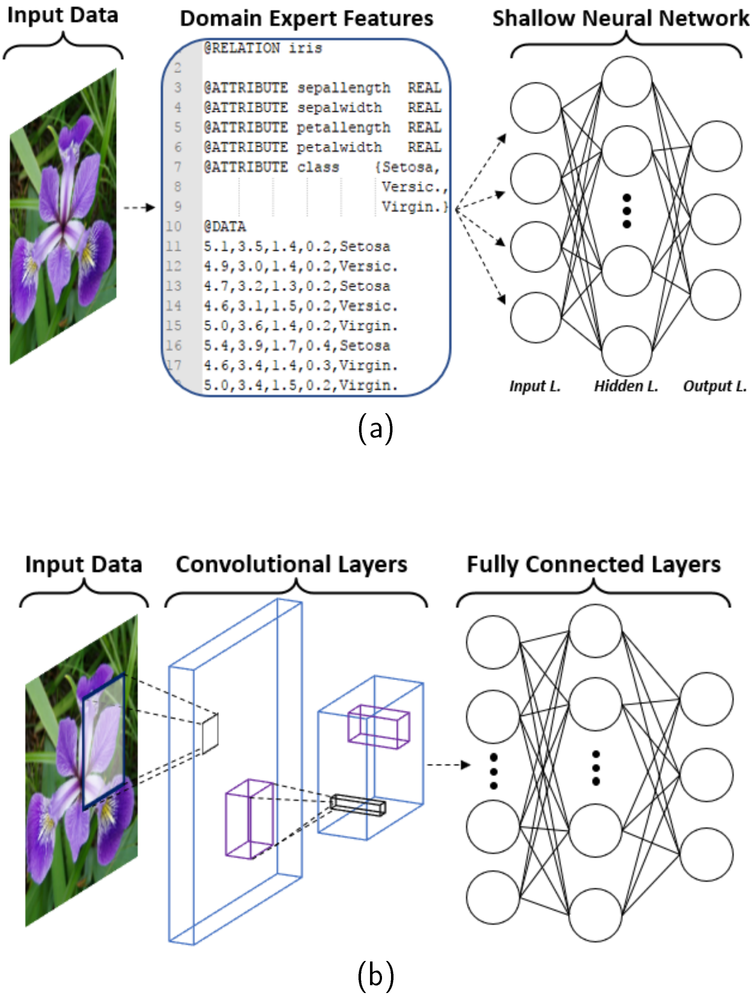
4.1 Hidden Layer Sizing

As seen in section 1.1, Convolutional Neural Networks (CNNs) are very similar to traditional (shallow) Artificial Neural Networks (ANNs): they are both made of neurons, usually organised in layers, connected to form a network in which the output of a neuron is the input of some others. This elaboration chain made it possible to transform the input data in the desired output value (i.e. a class for a classification problem or a value for a regression one), learning the best way of doing it during the training phase. However, while shallow neural networks operate on the features designed and extracted by a domain expert (figure 4.1a), CNNs use an hierarchy of convolution operations (whose kernel's weights are learned in the very same way classical neurons are) to autonomously extract the feature that better models the problem under analysis (figure 4.1b).

All CNNs of the type described in figure 4.1b can be seen as a stack of neurons specialised for the feature engineering (the ones in the convolutional layers) and neurons intended for the classification task (the ones in the fully connected layers). According to this point of view, the fully connected layers can be seen as a standard multilayer perceptron (MLP) classifier that relies on features extracted from the convolutional layers (although it is worth noting that all layers are trained together). This distinction is important since convolutional and fully connected layers have a very different number of parameters. Using AlexNet as example, it has $2,334,080$ parameters in the convolutional layers ($\sim 4\%$ of the total) and $58,631,144$ parameters in the fully connected ones ($\sim 96\%$ of the total). The reason is that AlexNet was originally intended to face the 1000 classes of ImageNet, therefore, after the feature engineering, a great representational capacity is required.

Since optimising a huge number of parameters can easily result in overfitting, in section 1.2 we have seen that a widely adopted solution is to make use of transfer learning techniques. In particular, there are two possible ways of leveraging knowledge from a pre-trained network: fine-tuning the net to adapt it to the new task; directly using the network as a feature exactor. ***It is important to highlight that, while the feature extractor approach does not change the network structure (it just exploits the net inherit hierarchical representation), fine-tuning implies a change in the architecture, without any concern on if and to what extent this alteration can impact the efficiency and effectiveness of the net.***

Indeed, fine-tuning is usually applied on problems having a smaller number of classes, causing a *bottleneck* in the network structure: e.g. in the



```
@RELATION iris

@ATTRIBUTE sepallength  REAL
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  REAL
@ATTRIBUTE petalwidth   REAL
@ATTRIBUTE class       {Setosa,
                        Versic.,
                        Virgin.}
@DATA
5.1,3.5,1.4,0.2,Setosa
4.9,3.0,1.4,0.2,Versic.
4.7,3.2,1.3,0.2,Setosa
4.6,3.1,1.5,0.2,Versic.
5.0,3.6,1.4,0.2,Virgin.
5.4,3.9,1.7,0.4,Setosa
4.6,3.4,1.4,0.3,Virgin.
5.0,3.4,1.5,0.2,Virgin.
```

Figure 4.1: Comparison between a shallow and a convolutional neural network architecture. (a) A representation of the classical approach in which some features (in the middle) are extracted from input data (on the left) and used to train a shallow neural network (a multilayer perceptron in the image on the right); (b) An exemplification of a CNN using an hierarchy of (in the example 2) convolutional layers (in the middle) to autonomously learn the best representation of input data (on the left) in order to train a fully connected neurons network (on the right).

case reported in figure 1.6 in section 1.2, the 4096 neurons in the fc7 layer (originally intended to be the input for the 1000 classes output layer) are connected to a 2 neurons output layer. Besides the waste of representational capacity, this simplistic approach has three main drawbacks:

- can introduce an unnecessary additional computational and memorisation burden (with a direct impact on the net performance and required system characteristics);
- can cause the net to focus more on the noise within images rather than on other salient aspects;
- may require more training sample to converge (due to the higher number of parameters to fit).

Thus, in the view of networks size reduction, in this work *we propose to further adapt deep CNNs by performing a sizing of hidden layers (i.e. those between the input and the output layers) when using the fine-tuning strategy*. With the term sizing, we refer to the use of a suitable strategy to reduce the number of used neurons, without significantly affecting the network performance. This can be achieved in two different ways: by reducing the synapses (i.e. connections between neurons), or by removing whole neurons. Here we will analyse the latter approach, focusing in particular only on neurons in the fully connected layers since, as previously discussed, are the most numerous ones.

4.1.1 Choosing the Number of Neurons

Designing a neural network is something closer to art than to science. On one hand, the *universal approximation theorem* states that a single hidden layer (with an infinite number of neurons) feed-forward neural network can approximate any continuous function. On the other, an improper network design can easily result in under/over-fitting (figure 4.2).

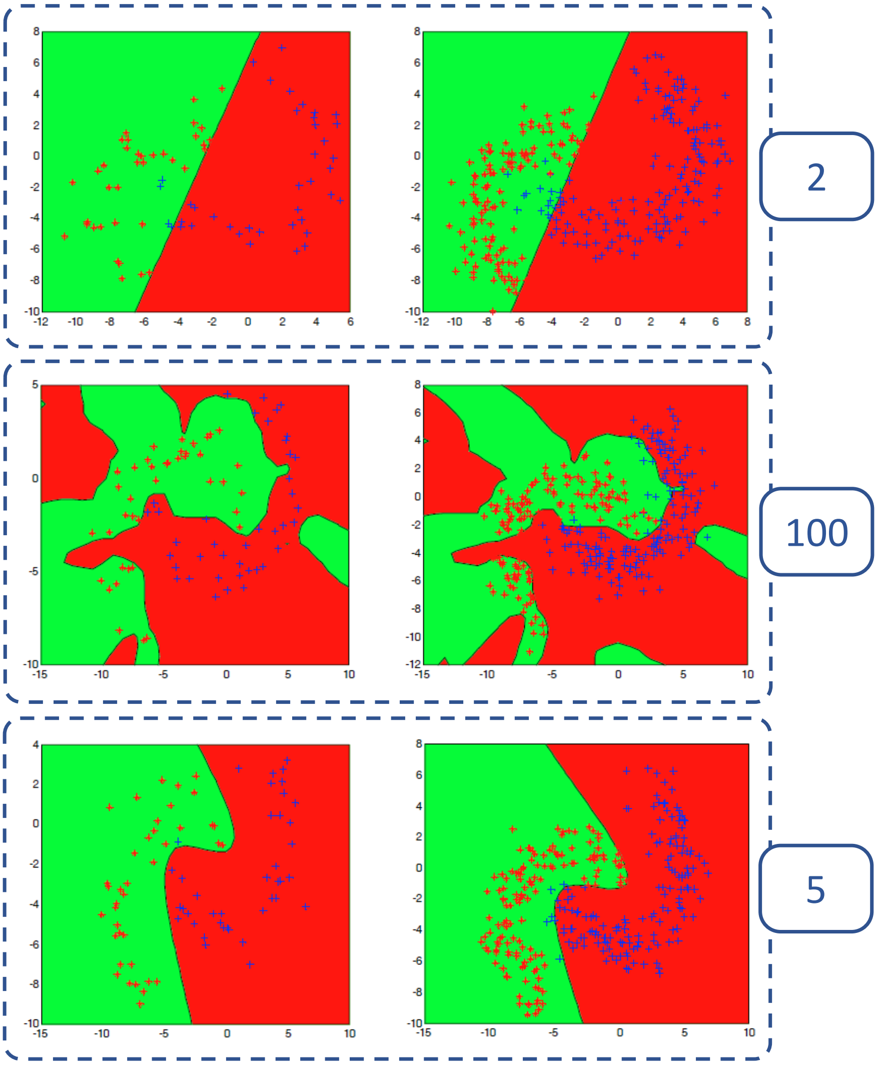


Figure 4.2: Illustrative representation of how the number of neurons affects the decision boundaries for a two-classes problem, in a single hidden layer multilayer perceptron. For each box, the image on the left refers to training samples, while the image on the right refers to validation samples. The number in the box represents the number of neurons in the hidden layer: 2 causes underfitting; 100 results in overfitting; finally 5 is the perfect value. Images adapted from [45].

Several heuristics have been so far proposed, each trying to define a “rule of the thumb” to help in the ANN design process. Among all, in this work we consider two approaches, chosen for their simplicity and diffusion:

- To use as many neurons as the average between the number of input and output neurons (hereafter referred as **A-Rule**);
- Defined m as the number of classes and i as the number of input neurons, in [81] the author propose to use $n = 2 * \sqrt{(m+2) * i}$ (hereafter referred as **Huang**).

4.2 Sizing as CNNs Approximation Technique

To measure the suitability of the sizing strategy for approximate computing purposes, it is important to measure the resiliency of the *“sized”* network, with respect to the *“original”* one, on a given task. Both the network depth and the considered task might affect the approach effectiveness: the former, due to the different ratio between the numbers of neurons in the convolutional and in the fully connected layers; the latter due to the different number of classes, directly related to the number of neurons in the classification layer. Therefore, in this work we consider two different networks on three classification problems.

4.2.1 Experimental Setup

To take into account the depth of the networks, in this work we use two different CNNs pre-trained on ImageNet [152]: AlexNet [95] and Vgg19 [166]. The former, consists of 5 convolutional and of 3 fully connected layers, for a total of $60,965,224$ parameters (on the left in figure 4.3); The latter, consists of 16 convolutional and of 3 fully connected layers, for a total of $143,667,240$ parameters (on the right in figure 4.3). For both networks, the sizing approximation procedure is as follows:

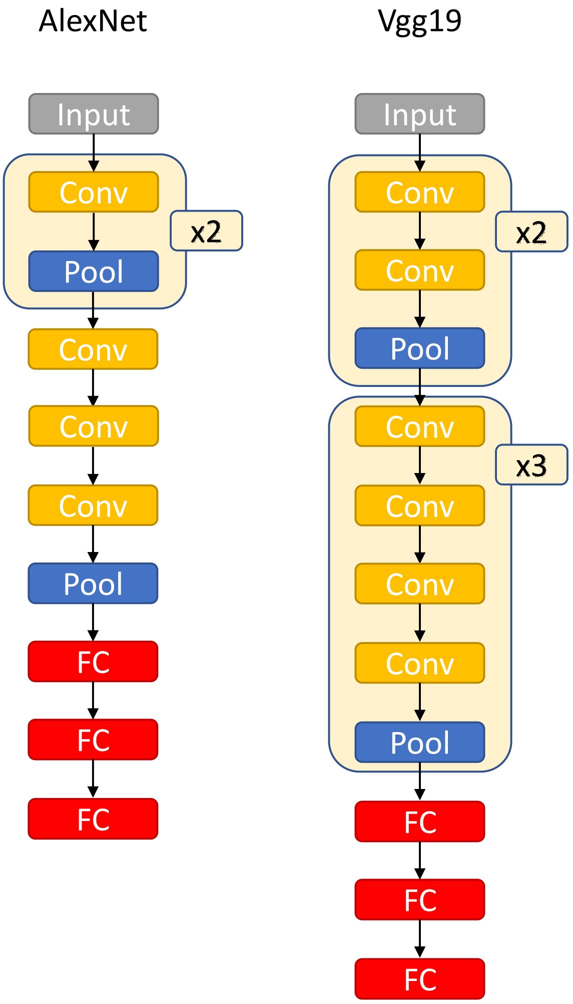


Figure 4.3: A simplified illustrative representation and comparison between AlexNet [95] and Vgg19 [166] CNNs architectures. In both cases, activation functions, dropout and other functional layers have not been reported. For details about the number of parameters, depth, etc., please refer to table 1.1 in section 1.2.

- Replace the last fully connected layer with a new fully-connected layer having as many neurons as the number of classes in the considered dataset;
- Change the shape of the second to last fully connected layer according to one of the sizing rule introduced in section 4.1.1;
- Freeze the trained weights and biases for all convolutional layers (by setting the learning rate to 0);
- Set a very low learning rate (10^{-4}) for the fully connected layers;
- Re-train the modified network on the new task.

Since also the used optimiser could affect the evaluation, all the experiments were run two times: the first, by using Stochastic Gradient Descent with Momentum (SGDM) [22], the second by using ADAM [88]. A 5-fold cross-validation was performed, with 3 folds using as training set, 1 as validation set and 1 as test set. In all the configurations, the training is stopped after 15 consecutive non-improvements on the validation set accuracy. As a result, it is worth noting that:

- all the weights in the convolutional layers will remain unvaried (the same learnt on ImageNet);
- the weights in layer the third to last fully connected layer will start the fine-tuning with weights learnt on ImageNet, but will be updated during the re-training;
- weights of last two fully connected layers will be randomly initialized and will be adapted during the training.

To evaluate the effectiveness of the proposed approach, we considered three datasets differing in terms of number of classes, number of samples and image resolution:

- The **Dogs vs Cats** dataset[17] [49], consisting in 25000 images of cats and dogs (figure 4.4) equally distributed;

- The **UIUC Sports Event** dataset[18] [102], containing images of 8 different sport activities (figure 4.5), distributed from 137 to 250 images per category;. All the images are also grouped into "easy" and "medium" according to the human subject judgement;

- The **Caltech 101** dataset[19] [51], collecting pictures of objects belonging to 101 different categories (figure 4.6), distributed from 40 to 800 images per category;

Figure 4.4: Sample images from Dogs vs Cats dataset [49].

[17] https://www.microsoft.com/en-us/download/details.aspx?id=54765

[18] http://vision.stanford.edu/lijiali/event_dataset/

[19] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

Figure 4.5: Sample images from UIUC Sports Event dataset [102].

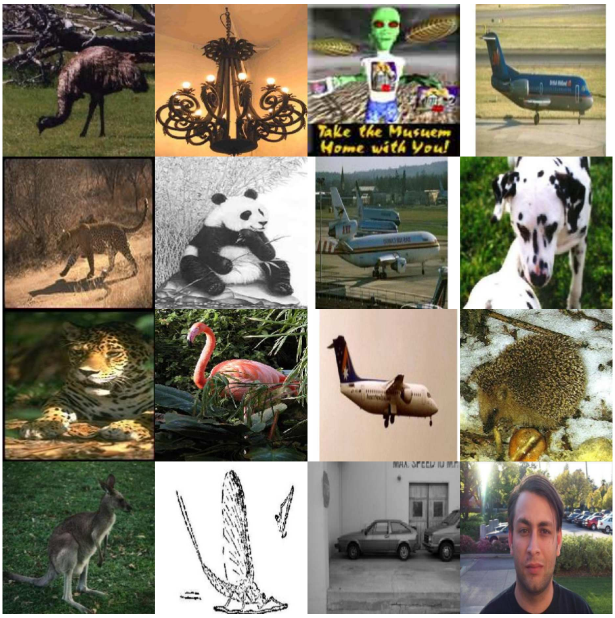


Figure 4.6: Sample images from Caltech 101 dataset [51].

4.2.2 Experimental Results

Measuring the resiliency of a deep CNN imply measuring the classification error rates of the approximated networks against those obtained by using the basic fine-tuning procedure. Tables 4.1, 4.2, 4.3 report the classification accuracy and the number of iterations needed to converge for AlexNet [95], varying the training optimiser (section 4.2.1) and the used sizing approach (section 4.1.1), for each considered dataset respectively (section 4.2.1). Tables 4.4 and 4.5 respectively report the number of parameters and occupied memory (in MB) for the same set of CNN, optimisation approach, sizing strategy and dataset.

Technique	Optimiser	Accuracy		Iterations	
		Mean	Median	Mean	Median
Base	SGDM	**0.9744±0.0021**	**0.9738±0.0026**	**4120**	**3800**
	ADAM	0.9658±0.0025	0.9656±0.0031	**7400**	**5000**
A-Rule	SGDM	0.9729±0.0024	0.9730±0.0030	5360	5000
	ADAM	**0.9664±0.0026**	**0.9660±0.0032**	11880	12000
Huang	SGDM	0.9729±0.0025	0.9722±0.0031	11120	13000
	ADAM	0.9663±0.0027	**0.9660±0.0033**	18160	17400

Table 4.1: AlexNet [95] 5-fold cross validation mean results for the Dogs vs Cats [49] dataset, with respective 95% confidence values when needed. In bold the best result for each combination.

Similarly, tables 4.6, 4.7, 4.8 report the classification accuracy and the number of iterations needed to converge for Vgg19 [166], varying the training optimiser (section 4.2.1) and the used sizing approach (section 4.1.1), for each considered dataset respectively (section 4.2.1). Tables 4.9 and 4.10 respectively report the number of parameters and occupied memory (in MB) for the same set of CNN, optimisation approach, sizing strategy and dataset.

Technique	Optimiser	Accuracy		Iterations	
		Mean	Median	Mean	Median
Base	SGDM	**0.9538±0.0095**	**0.9587±0.0119**	**1022**	**792**
	ADAM	**0.9601±0.0111**	**0.9621±0.0138**	192	**156**
A-Rule	SGDM	0.9531±0.0132	0.9495±0.0165	1190	1248
	ADAM	0.9588±0.0138	0.9495±0.0172	293	192
Huang	SGDM	0.9493±0.0113	0.9460±0.0141	1534	1368
	ADAM	0.9563±0.0156	0.9524±0.0195	**190**	168

Table 4.2: AlexNet [95] 5-fold cross validation mean results for the UIUC Sports Event [102] dataset, with respective 95% confidence values when needed. In bold the best result for each combination.

Technique	Optimiser	Accuracy		Iterations	
		Mean	Median	Mean	Median
Base	SGDM	0.9159±0.0100	0.9155±0.0125	**25258**	**24455**
	ADAM	**0.9334±0.0067**	**0.9338±0.0084**	9475	10512
A-Rule	SGDM	**0.9229±0.0073**	**0.9237±0.0091**	29667	29638
	ADAM	0.9324±0.0071	0.9335±0.0089	7680	7300
Huang	SGDM	0.9212±0.0071	0.9204±0.0089	31332	29273
	ADAM	0.9331±0.0082	0.9329±0.0103	**7227**	**6789**

Table 4.3: AlexNet [95] 5-fold cross validation mean results for the Caltech-101 [51] dataset, with respective 95% confidence values when needed. In bold the best result for each combination.

Although this work must be considered just as a proofs-of-concept, results show that it could be possible to reduce the number of neurons in the hidden layer without statistically affect the network performance, but with a significant reduction of the number of parameters (up to $\sim 27\%$ for AlexNet and $\sim 11\%$ for Vgg19) and required memory occupation (up to $\sim 41\%$ for

		#Parameters	Δ	Δ%
Dogs Vs Cats [49]	Base	56,876,418	-	-
	A-Rule	48,485,765	8,390,653	14.75%
	Huang	41,136,258	15,740,160	27.67%
UIUC Sports Event [102]	Base	56,901,000	-	-
	A-Rule	48,510,380	8,390,620	14.75%
	Huang	41,745,340	15,155,660	26.64%
Caltech-101 [51]	Base	57,282,021	-	-
	A-Rule	48,894,417	8,387,604	14.64%
	Huang	41,136,258	15,740,160	27.67%

Table 4.4: Summary of the number of AlexNet [95] parameters for each sizing technique and considered dataset. The table also report the numerical difference (Δ) and the percentage saving (Δ%) obtained by using the sized network w.r.t. the base fine-tuning approach.

		Memory (MB)	**Δ (MB)**	Δ%
Dogs Vs Cats [49]	Base	211	-	-
	A-Rule	177	34	16.11%
	Huang	124	87	41.23%
UIUC Sports Event [102]	Base	207	-	-
	A-Rule	176	31	14.98%
	Huang	152	55	26.57%
Caltech-101 [51]	Base	209	-	-
	A-Rule	178	31	14.83%
	Huang	166	43	20.57%

Table 4.5: Summary of the required AlexNet [95] memory for each sizing technique and considered dataset. The table also report the numerical difference (Δ) and the percentage saving (Δ%) obtained by using the sized network w.r.t. the base fine-tuning approach.

Technique	Optimiser	Accuracy		Iterations	
		Mean	Median	Mean	Median
Base	SGDM	**0.9887±0.0010**	**0.9884±0.0013**	**9160**	9800
	ADAM	0.9870±0.0009	0.9870±0.0011	6760	2200
A-Rule	SGDM	0.9881±0.0006	0.9882±0.0007	10360	10200
	ADAM	0.9864±0.0015	0.9858±0.0019	6200	5200
Huang	SGDM	0.9884±0.0009	0.9880±0.0011	9200	**8400**
	ADAM	**0.9873±0.0013**	**0.9872±0.0016**	**3280**	**1400**

Table 4.6: Vgg19 [166] 5-fold cross validation mean results for the Dogs vs Cats [49] dataset, with respective 95% confidence values when needed. In bold the best result for each combination.

Technique	Optimiser	Accuracy		Iterations	
		Mean	Median	Mean	Median
Base	SGDM	0.9658±0.0097	0.9621±0.0121	**377**	**276**
	ADAM	0.9734±0.0026	0.9716±0.0032	**84**	**72**
A-Rule	SGDM	0.9683±0.0066	0.9685±0.0082	386	372
	ADAM	**0.9740±0.0025**	**0.9746±0.0031**	245	96
Huang	SGDM	**0.9709±0.0043**	**0.9714±0.0054**	1022	1068
	ADAM	**0.9740±0.0044**	**0.9746±0.0054**	192	156

Table 4.7: Vgg19 [166] 5-fold cross validation mean results for the UIUC Sports Event [102] dataset, with respective 95% confidence values when needed. In bold the best result for each combination.

AlexNet and $\sim$ 10% for Vgg19). This would imply that, though fine-tuning can already be effectively used in many different contexts, other investigations are needed to develop improvements that allow unleashing its full potential. Moreover, the possibility of reducing the memory occupation without a direct

Technique	Optimiser	Accuracy		Iterations	
		Mean	Median	Mean	Median
Base	SGDM	0.9362±0.0035	0.9377±0.0043	11271	11242
	ADAM	0.9378±0.0063	0.9377±0.0078	**569**	**511**
A-Rule	SGDM	0.9478±0.0075	**0.9531±0.0094**	11811	11023
	ADAM	**0.9415±0.0065**	**0.9431±0.0082**	657	584
Huang	SGDM	**0.9483±0.0061**	0.9475±0.0076	**9724**	**8103**
	ADAM	**0.9415±0.0065**	**0.9431±0.0082**	657	584

Table 4.8: Vgg19 [166] 5-fold cross validation mean results for the Caltech-101 [51] dataset, with respective 95% confidence values when needed. In bold the best result for each combination.

		#Parameters	Δ	Δ%
Dogs Vs Cats [49]	Base	139,578,434	-	-
	A-Rule	131,187,781	8,390,653	6.01%
	Huang	123,838,274	15,740,160	11.28%
UIUC Sports Event [102]	Base	139,603,016	-	-
	A-Rule	131,212,396	8,390,620	6.01%
	Huang	124,447,356	15,155,660	10.86%
Caltech-101 [51]	Base	139,984,037	-	-
	A-Rule	131,596,433	8,387,604	5.99%
	Huang	123,838,274	15,740,160	11.28%

Table 4.9: Summary of the number of Vgg19 [166] parameters for each sizing technique and considered dataset. The table also report the numerical difference (Δ) and the percentage saving (Δ%) obtained by using the sized network w.r.t. the base fine-tuning approach.

impact on the network classification ability open new scenarios toward the application of powerful CNNs on embedded devices, as already done with Random Forest classifier [15].

		Memory (MB)	**Δ (MB)**	**Δ%**
Dogs Vs Cats [49]	Base	508	-	-
	A-Rule	488	20	3.94%
	Huang	461	47	9.25%
UIUC Sports Event [102]	Base	507	-	-
	A-Rule	477	30	5.92%
	Huang	452	55	10.85%
Caltech-101 [51]	Base	506	-	-
	A-Rule	482	24	4.74%
	Huang	465	41	8.10%

Table 4.10: Summary of the required Vgg19 [166] memory for each sizing technique and considered dataset. The table also report the numerical difference (Δ) and the percentage saving (Δ%) obtained by using the sized network w.r.t. the base fine-tuning approach.

4.3 Sizing and Adversarial Perturbations

In section 1.3 we introduced the problem of misleading a CNN by means of adversarial perturbations. Common defence techniques usually try to make CNNs more robust by either working on the data (to find out the adversarial samples [200] or to destroy the injected artefacts [47]) or on the way the model learns from it [117]. But is it possible that the shape of the network itself contributes to the effectiveness of such attacks? Our idea is that the shape of CNNs itself could be part of the reasons why they are susceptible to adversarial attacks.

Although recently, some authors have proposed adversarial attacks able to work in several domains [29, 159], in this section we will focus only on CNNs for image processing and on adversarial attacks applicable to them. This is because i) CNNs represents the most used deep neural networks and ii) all the adversarial attacks so far proposed in any domain are the same, or are an adaptation, of those intended against images. Therefore, we analyse the impact that the sizing strategy has on the robustness of CNNs against adversarial perturbation approaches. To this aim, table 4.11 and 4.12 respectively report the robustness of AlexNet [95] and Vgg19 [166] against two adversarial perturbation strategies (see section 1.3.1 for details) on the UIUC Sports Event Dataset [102], varying the sizing approach (section 4.1.1). The value ρ, introduced in [120] and related to the magnitude of the adversarial noise needed to mislead the CNN, is defined as

$$\rho = \frac{||N_a||_2}{||I||_2} \tag{4.1}$$

where N_a is the injected adversarial noise and I is the target image. The column *"Time"* refers to the time (in seconds) needed to craft the adversarial samples. It is worth noting that, to provide fair results, the CNNs have been trained of the training set, while the adversarial samples have been crafted only for images in the test set. Moreover, both *rho* and *"Time"* values have been measured only for successfully crafted adversarial samples.

	Fool	ρ		Time (s)	
		Mean	Median	Mean	Median
Base	FGSM	0.0183	0.0185	2690.45	2811.46
	DeepFool	0.0407	0.0419	706.83	703.33
A-Rule	FGSM	0.0193	0.0194	3260.09	3146.05
	DeepFool	0.0442	0.0452	722.80	754.96
Huang	FGSM	**0.0199**	**0.0202**	4745.09	4149.13
	DeepFool	**0.0488**	**0.0499**	558.67	554.58

Table 4.11: AlexNet [95] robustness to FGSM [64] and to DeepFool [120] adversarial perturbations attacks on the UIUC Sports Event Dataset [102], varying the sizing approach.

	Fool	ρ		Time (s)	
		Mean	Median	Mean	Median
Base	FGSM	0.0177	0.0182	18386.91	18634.11
	DeepFool	0.**1188**	0.0754	4305.76	4383.62
A-Rule	FGSM	0.0192	0.0190	27851.98	28652.63
	DeepFool	0.0713	0.0730	5151.23	5243.81
Huang	FGSM	**0.0200**	**0.0201**	22059.96	22084.38
	DeepFool	0.1000	**0.0882**	4132.82	4130.29

Table 4.12: AlexNet [166] robustness to FGSM [64] and to DeepFool [120] adversarial perturbations attacks on the UIUC Sports Event Dataset [102], varying the sizing approach.

Interestingly, results show that, apart for a single combination, sized CNNs needs a “stronger” adversarial noise to be mislead. This further motivate other investigations in the direction of CNNs layer sizing, since the reported analysis seems to suggest that the adversarial perturbation problem can be faced (or at least limited) by reducing the number of neurons/connections that actively take part in the network decision problem. This results, although preliminary, represent a novel contribution in the field of adversarial defense strategies since CNN sizing does not relay on the analysis of the data but could make CNNs intrinsically more robust by changing the neurons connection pattern. This will help the user in “sizing” the network accordingly to the desired levels of performance/robustness they needs to obtain on the basis of the risk associated with the task.

It is also worth to note that the approach is totally topic-agnostic, meaning that it is applicable in several contexts and for different tasks (i.e. user recognition, object detection, etc), helping researchers in choosing the most suitable solution without changes in the procedure. Moreover, one of the big strengths of the approach is that it can theoretically be used also on already developed network, since it only operates on the network structure, not on the data nor on the training procedure. This has a huge impact on its applicability, since it means that also already deployed CNN bases application could be made more robust to adversarial attacks and updated in a total transparent way from the user perspective, since the input/out of the network will remain totally unchanged.

Part III

Undermining Security and Fairness

As seen in part I, artificial intelligence, and in particular deep learning, is being involved in a wide range of application fields, including critical ones. As a consequence of this wide spreading, AI-based systems are becoming more and more target of attacks aimed in circumventing them (see section 1.3 for details). Indeed, if on one hand, industry is pushing toward a massive use of artificial intelligence enhanced solution, on the other it is not adequately supporting researches in end-to-end understating of capabilities and vulnerabilities of such systems. The results may be very (negatively) mediatic[20], especially when regarding borderline domain such as the reliability of autonomous driving systems [169]. Unfortunately, this contributes in further undermining people trust in AI, whose reputation is already tarnished by mass media[21].

Since AI is very likely to be an important part of our everyday life in the very next future, it is crucial to build trust in AI systems. Although the solution is not straightforward, a crucial step in that direction is to raise awareness about security and fairness threats of these systems, from a technical perspective as well as from the governance and from the ethical point of view. Several are the issues that must be faced, such as: designing systems that analyse people data ensuring privacy by default; analysing the limitations and the weaknesses that might affect an AI-based system, independently from the particular adopted technology or technical solutions; assessing the behaviours in the case of successful attacks and/or in presence of degraded environmental conditions.

In this part of the thesis, we will focus on these aspects. In particular, in chapter 5 we investigate the vulnerability of biometric based authentication systems, while in chapter 6 we consider the privacy and fairness concerns associated with automatic face analysis. In both cases we take advantage

[20] https://towardsdatascience.com/your-car-may-not-know-when-to-stop-adversarial-attacks-against-autonomous-vehicles-a16df91511f4

[21] https://www.theguardian.com/commentisfree/2016/apr/07/robots-replacing-jobs-luddites-economics-labor

of *adversarial perturbations*, making them the key factor the proposed approaches.

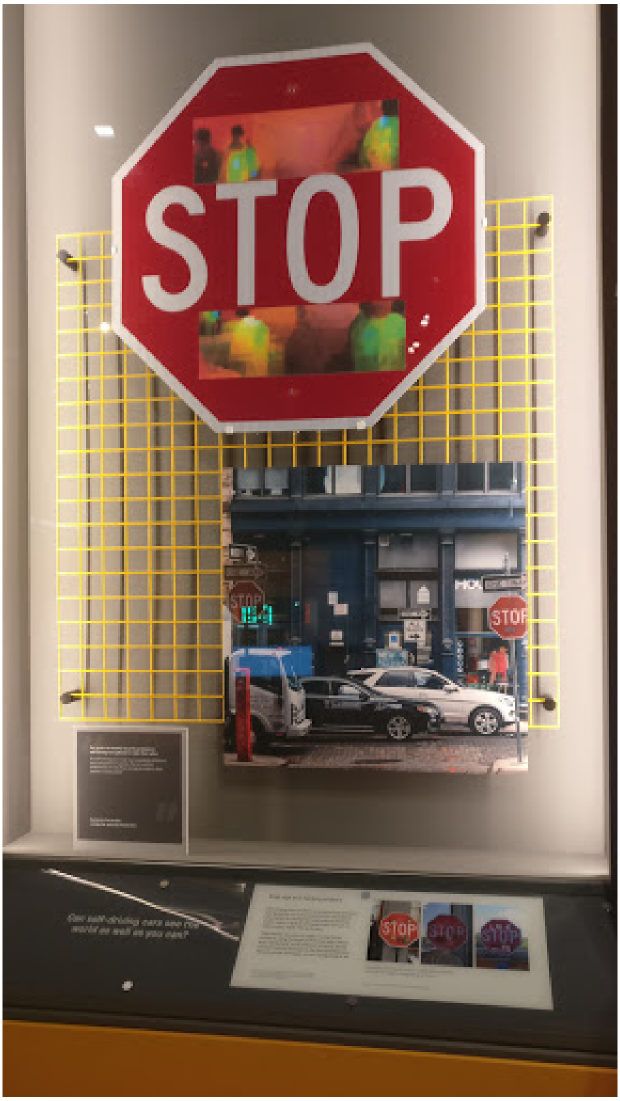


Figure 4.7: Installation at the London Science Museum on an adversarial attack against automatic traffic signs recognition.

5

Biometrics Authentication Systems

In a more and more connected world, one of the actions we perform more often is to verify our identity, in order to get access to our laptop, mobile phone, bank account, university services, etc. There are several ways to prove someone's identity, such as by using a certificate, a personal document, a password, a key, etc. All these means share the characteristics of having some sort of "secret" that only the real user knows/posses and, thus, providing it demonstrate the user's identity.

Over the years, as the risk associated with grating access to unauthorised users increased, industry moved toward users' identification bases on something the subject can not lend to anyone else [17]. Therefore, with the growing availability of small, cheap and reliable biometric acquisition scanners, the spread of Biometric-based Authentication Systems (BAS) in daily life consumer electronics (like smartphones and laptops) have been increasing [206]. Biometrics are becoming the security standard de-facto in several context, mainly because they are usually easy and safe to acquire [21], allowing the system to identify a subject (and thus to understand its authorisation levels) on the basis of characteristics that describe what the user is (such as a fingerprint), more than what a user own (like a key).

As for in many other domains, in the last years industry is pushing toward the use of artificial intelligence and deep learning in BAS [176]. The aim is to increase BAS reliability and versatility by leveraging machine learning approaches to relieve domain expert from designing new solutions. However, as seen in the introduction of this part of the thesis, it is of crucial importance to have and end-to-end understanding of capabilities and vulnerabilities of such AI-bases systems when used in critical domains. Therefore, in this chapter we investigate whether it is possible to circumvent a CNN-based BAS by exploiting adversarial perturbations (see section 1.3.1 for details).

5.1 Biometrics

Biometrics analysis, as the term suggest, refers to the "measure" of some "biological" characteristics of a subject to infer some property about they (e.g. the user identity). Biometrics can be grouped into hard (also known as primary) and soft (figure 5.1), with the first referring to physical [85], behavioral [119], and biological characteristics [156] directly measurable on the subject, and the latter concerning ancillary characteristics related, for example, to the subject nationality, gender, age and so on [3]. Over the years, researchers' interest is moving from hard to soft biometrics, at the beginning mainly with the aim of improving authentication system effectiveness [205, 4], then focusing on subject identification [186, 146].

Each biometrics has some pros and cons, usually related with the intrinsic security level or to the invasiveness of the acquisition procedure. For this reason, the choice for the most appropriate solution depends on the purpose the system has been designed for. For example, DNA can be considered ideal to verify someone's identity, but the acquisition procedure might be tedious; iris is usually easy to acquire, but the associated security level not very high; etc. Therefore, it is not uncommon the contemporaneous use of several biometrics (sometimes less effective, but cheaper) to cope with the

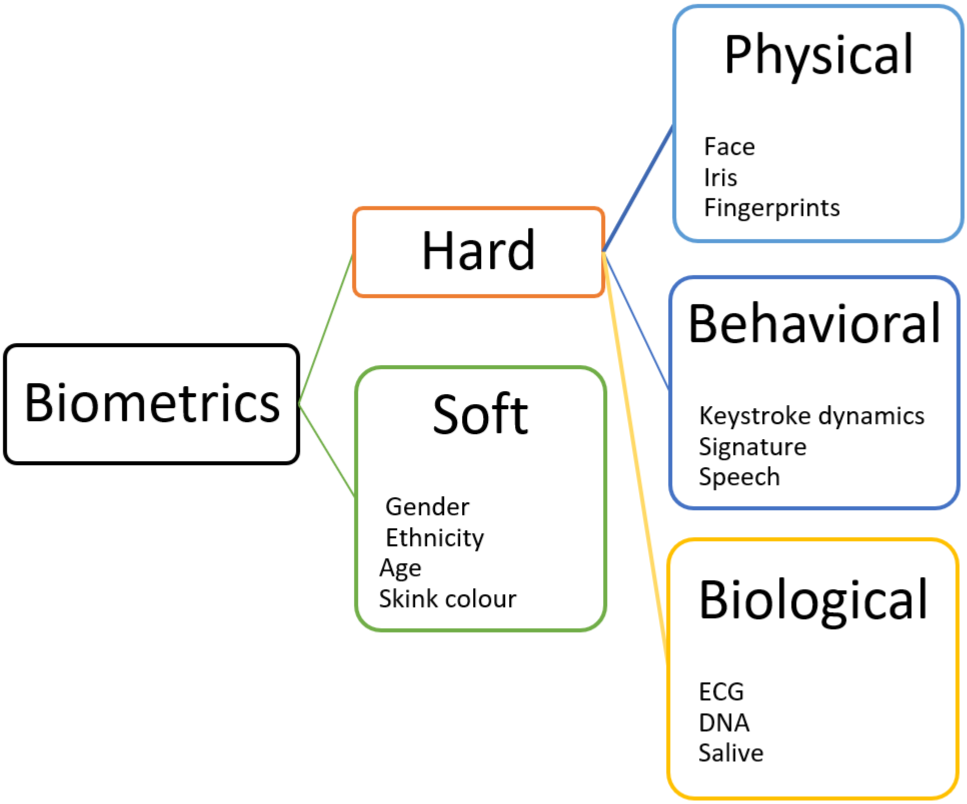


Figure 5.1: Biometrics classification schema in hard and soft, according to the grouping made in [3].

system requirements [27]. Unfortunately, as for any other authentication means, it is possible to attack a BAS by using a counterfeit replica of the target subject biometrics (see figure 5.2 for an example). Since the attack consists in "presenting" the fake replica to the scanner, this type of attacks goes under the name of *Presentation Attack* (PA). As the spread of BAS increased, the same happened to the effectiveness of presentation attacks, to the point of even starting a public debate about their usage[22]. Therefore, to

[22] https://www.bizjournals.com/washington/blog/techflash/2013/10/this-is-the-biometric-war-michael.html

face the problem, researches started developing *Liveness Detectors* (LDs), namely methods aimed in detecting whether the acquired biometrics belongs to a *live* (and thus real) subject or no.

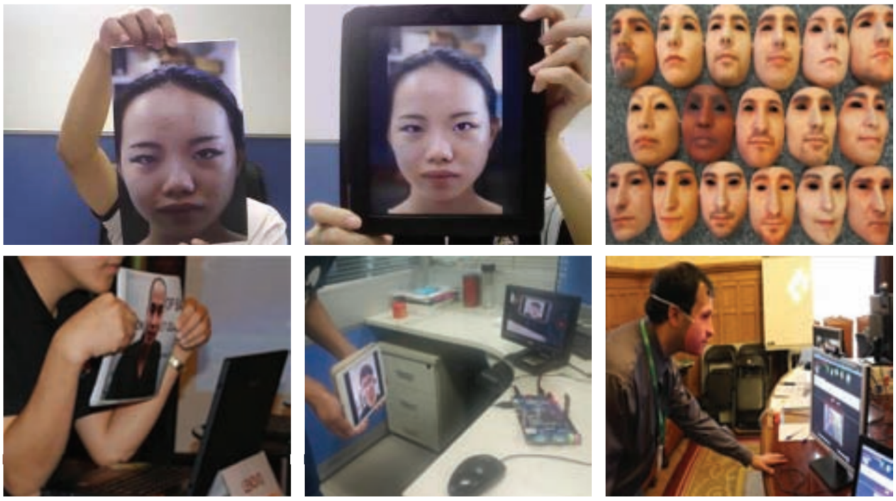


Figure 5.2: Example of a face presentation attack. From left to right: a printed face; a face reproduced by means of an electronic device; a face 3D mask. Top row show the fake biometrics replicas, while bottom row show the attack by using the corresponding replica. Images taken from [101].

Detecting fake biometrics usually involve the analysis of characteristics that, evident in the case of real acquisitions, are hard to replicate on counterfeit replicas (e.g. heartbeat in a finger, eye blinking for a face, etc). These characteristics can be identified by means of external hardware (e.g. a depth camera to analyse whether a face is tridimensional or just printed) or by simply analysing the acquired biometry by using an ad-hoc software. If, on one hand, the former tends to be more accurate and reliable, on the other it also might be hard to be put in practice (e.g. because too expensive, too big, etc). Thus, on the basis of the desired security level, it is possible to use a different approach for liveness detection against presentation attacks.

5.2 Fingerprint-based Authentication Systems

The last years growing availability of small, cheap and reliable fingerprint acquisition scanners has been resulting in an increasing spread of Fingerprint-based Authentication Systems (FAS) in consumer devices, such as smartphones and laptops. The success of FAS with the mass audience is mostly related to their high users' acceptability, thanks to the fact that fingerprints are considered secure (both in terms of subject identification reliability and of spoofing difficulty) and their acquisition not invasive [21]. The flip side of the coin is that this wide-spreading has giving rise to a new wave in research on smarter spoofing attacks, namely procedures aimed to bypass a FAS by using a counterfeit fingerprint (see section 5.1 for details).

As happened for other biometrics, to face this problem researches are focusing on the development of more effective Liveness Detectors (LDs) approaches to discern authentic (i.e. acquired by using a "live" finger) fingerprints from artificial replicas. As a consequence, a modern FAS (Fig. 5.3) usually comprises a Liveness Detector (LD) stage [131]: it is clear that detecting fake fingerprints as soon as possible is crucial [5] to reduce the computational burden and to limit the probability of unauthorised access. Determining whether a fingerprint comes from a live finger or not is a task that can be faced by exploiting additional hardware or by only relying on the data coming from the scanner. Despite the use of external sensors data (e.g. temperature, humidity, etc) may be more effective [115], using hardware-based LD it usually results in higher costs and bigger scanner. For this reason, over the last years researches mostly focused on the development of LD based on the analysis of the fingerprint image as acquired by the scanner.

Performing the attack at sensor level implies that the liveness detection can be addressed in the context of computer vision. As for many other computer vision problems, over the years machine learning (ML) based approaches demonstrated to be reliable and effective for fingerprint liveness detection,

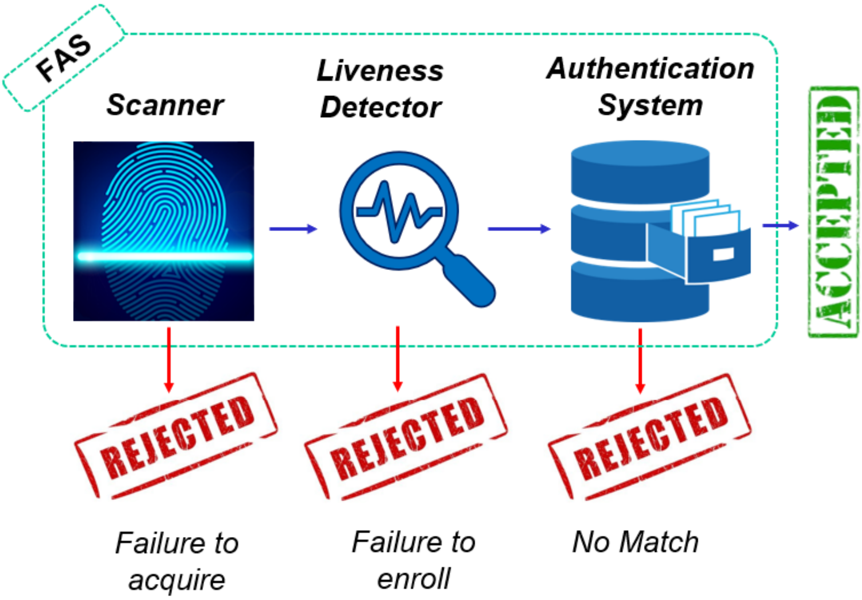


Figure 5.3: Exemplification of a modern FAS working schema. On the left, the scanner acquires the fingerprint: if there is any acquisition problem, the acquisition in rejected (Failure to acquire), otherwise the acquired image is passed to the next stage. In the middle, the liveness detector checks whether the input sample comes from a live (real) finger or not: in the former case it is passed to the next stage; the latter case it is rejected (Failure to enrol). On the right, the authentication system verifies if the fingerprint belongs to an authorised user, granting or forbidding the access.

with recent deep Convolutional Neural Networks (CNN) obtaining state-of-the-art performance in detecting a wide number of spoofing approaches [60, 35]. However, as seen in section 1.2, the term deep refers to the number of stacked layers and thus, indirectly, to the total number of neurons in the network. To estimate millions of parameters without incurring in over-fitting, a huge number of annotated samples is usually required. Unfortunately, *collecting a big fingerprint dataset could be difficult, expensive and time-consuming*. Therefore, one of the most adopted solutions [21] is fine-tuning, consisting in using a CNN pre-trained on a different task (having a suitable amount of available training samples) and performing a *re-training* of some of the layers (usually last ones) to adapt the network to the new task while preserving part of the past learned knowledge (see section 1.2 for details). However, although fine-tuning demonstrated to be very effective [204], the use of a pre-trained CNN does not bring only benefits but also cause inherit its weaknesses that could nullify the effectiveness of a liveness detector. Among all, the most critical are represented by adversarial perturbation [181], i.e. the ability of ad-hoc crafted noise to mislead a CNN (see section 1.3 for details).

Despite natural images and fingerprints belongs to different domains, this weak-point could open new attack scenario never considered before. Therefore, to shed lights on this problem and raise domain experts awareness, *we want to design an adversarial perturbation based attack to arbitrarily cause state-of-the-art CNN-based liveness detectors to misclassify a fake fingerprint*. In particular, the aim is to understand if and to what extent adversarial perturbations can affect FASs by:

- exploring the effectiveness of some adversarial attacks on a CNN based fingerprint liveness detection system;
- *analysing the impact of the injected adversarial perturbation on the fingerprint authentication algorithm*;
- proposing a strategy to improve the attack success rate. Moreover, in order to design an attack procedure usable in a real-world scenario, we also propose some constraints to make the generated fake fingerprint printable.

It is worth noticing that the need to preserve fingerprint key authentication characteristics while injecting a perturbation able to mislead the liveness detector is a way of exploiting adversarial perturbation that, to the best of our knowledge, has never been so far proposed.

5.2.1 Fingerprint Liveness Detection

As stated in the previous section, fingerprints are the most popular biometrics since they have been proven to be reliable, easy to implement and to use [21]. Moreover, the reduction of both the size and the cost of fingerprints sensors have been causing a spread of biometric authentication systems based on fingerprints. As a consequence, a growing number of attacks have been designed to circumvent the authentication system and thus providing unauthorized access. As seen in the previous section, a common strategy is emphpresentation attack, consisting in the submission (presentation) of an artificial replica of the finger to the sensor. Several materials can be used for this purpose (including cheap and very accessible ones, such as wood glue and play-doh), as long as they fairly emulate a finger skin characteristics (figure 5.4). A fake replica can be made with (consensual), without (unconsensual) or with the partial (semi-consensual) authorization of the real user [60], starting, for example, from high-resolution photos, fingerprints left on an object, etc.

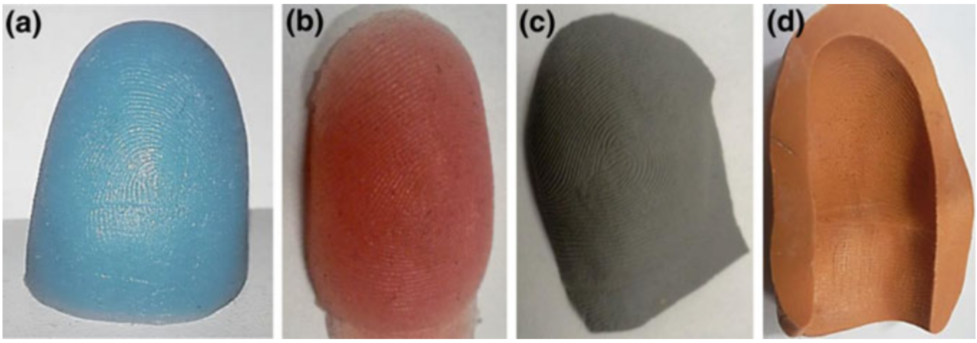


Figure 5.4: Artificial finger replicas made using GLS (a), Ecoflex (b), Liquid Ecoflex (c) and Modasil (d). Images taken from [21].

In this context, *Liveness Detection (LD)* (section 5.1) is the task of determining whether a fingerprint belongs to a real (live) finger or to a fake replica, before its submission to the authentication system. This task can both i) leverage only on the fingerprint image acquired by the scanner or ii) exploit data coming from additional sensors (such as temperature, blood pressure, humidity etc.). If, on one hand, the information provided by external hardware can improve the recognition rate, on the other its acquisition could not be always viable (e.g. it is not possible to use this approach for already deployed sensors). Thus, much effort has mainly been dedicated to developing image-based (software) techniques.

Over the years, increasingly sophisticated LDs approaches have been proposed to cope with increasingly challenging spoofing techniques. To support this process, in 2009 was started the first Liveness Detection Competitions [110], a biennial contest in which participants are challenged to identify spoof biometrics (iris and fingerprints) from live samples. The urge is to allow researches to compare on a standardised, common experimental protocol, providing them with a large quantity of fake and live samples, collected in a well-known environment, in the viewing of sustaining results reproducibility (see section 3 for details). The task complexity and liveness detectors performance improved over LivDet editions. The 2015 edition [122] set a turning point not only because the test-set spoof fingerprints were produced by also using materials not included in the training set, but mainly because it was the first time that a CNN was used for fingerprint liveness detection, resulting in accuracy levels, intra-materials and intra-sensors generalization ability never reached before. In particular, it is also worth noting that, despite the runner-up [65] used a sophisticated approach able to exploit both spatial and frequency domain features, the competition was won by an approach [126] relaying on a Vgg19 CNN [166] pre-trained on ImageNet and then fine-tuned on the fingerprint liveness detection task.

Thereafter, not only livdet top performers always include CNN-bases approaches, but interestingly, similar solutions resulted effective also with other biometrics [142, 77]. This give rise to some questions: how vulnerable are CNN-based fingerprint liveness detectors to adversarial perturbations? Is the counterfeit fingerprint still correctly recognised by the authentication system or had its main characteristics totally been destroyed by the adversarial attack? To answer these questions, analysing the effectiveness of adversarial perturbation approaches against FAS is of crucial importance.

5.3 Adversarial Presentation Attack

The potential vulnerability of CNN-bases liveness detection systems could open new scenarios in which an attacker could be able to make a fake fingerprint being recognised as live by exploiting adversarial perturbations, giving rise to an *adversarial presentation attack*. It is worth noting that, since a FAS must be considered as composed of two subsystems (figure 5.3), to be considered successful a FAS outbreak must not only be able to circumvent the liveness detector, but also have to preserve the biometry as clean as possible in order to be still able to break the authentication system. Therefore, in this work we propose to evaluate FAS attacks in two steps (figure 5.5): first, we evaluate the effectiveness of the attack against the LD subsystem and then we verify the robustness of the AS to fake adversarial replicas that were able to mislead the liveness detector.

5.3.1 Attacking the Liveness Detector

As stated in section 1.3.1, several adversarial perturbation attacks were so far proposed. However, to the best of our knowledge, *adversarial perturbation was never used as a method to mask a presentation attack for fingerprints, neither for any other biometry*. In this work *we thus propose to attack the fingerprint liveness detector system by determining an adversarial perturbation over a fake fingerprint image such that it can be recognized as live and thus submitted to the matching process, while preserving as minutiae as possible in order to be still able to break the authorization system.* Since none

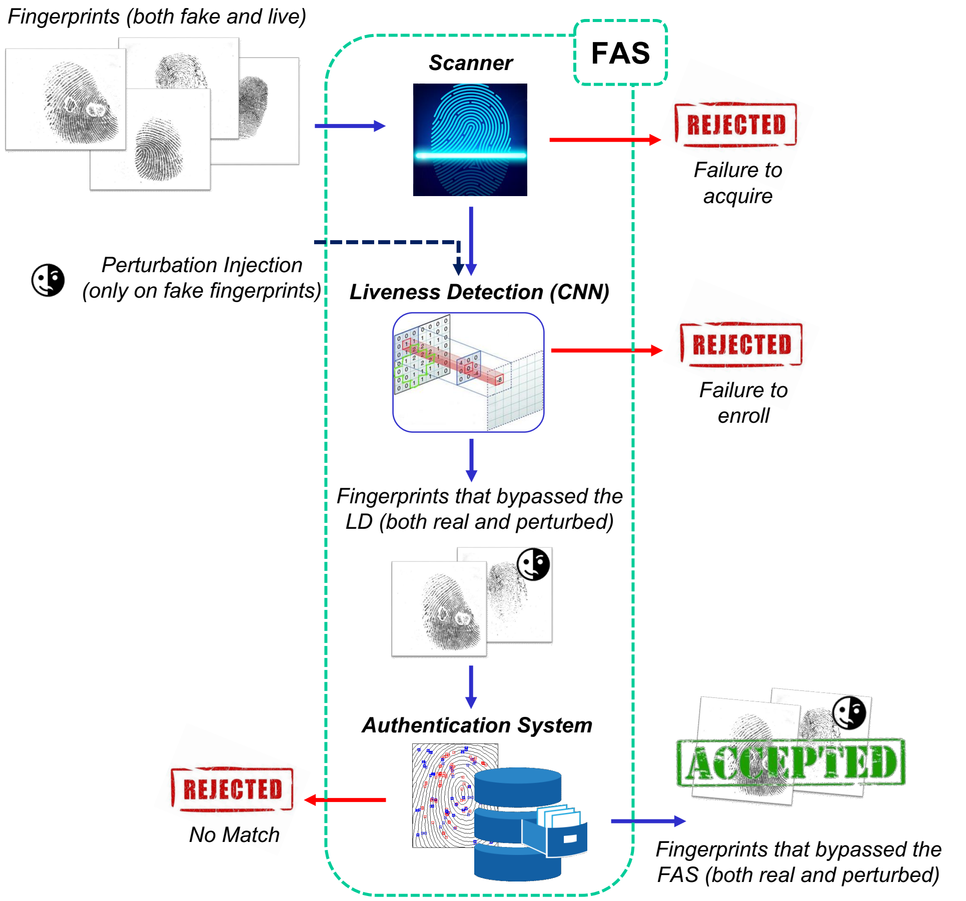

Fingerprints (both fake and live)
FAS
Scanner
REJECTED
Failure to acquire
Perturbation Injection (only on fake fingerprints)
Liveness Detection (CNN)
REJECTED
Failure to enroll
Fingerprints that bypassed the LD (both real and perturbed)
Authentication System
REJECTED
No Match
ACCEPTED
Fingerprints that bypassed the FAS (both real and perturbed)

Figure 5.5: Exemplification of the proposed attack schema. On the left, a set of fingerprints, some of which (the marked ones) perturbed in order to try to mislead the FAS. In the centre, after the LD some fake fingerprints are still present and will be submitted to the AS. Finally, on the right, the set of authorized accesses contains a fake fingerprint, meaning that the adversarial attack was successful.

of the perturbation approaches so far developed demonstrated to be the most effective, in order to better understand the vulnerability of FASs according to the definition proposed at the beginning of section 5.3, in this work, we analysed the effectiveness of three perturbation techniques:

1) As a baseline, we considered the iterative version of the FGSM proposed by [97], in order to asses if a very intuitive and simple to apply attack can break the liveness detector. For the attack, we used an initial standard deviation $\varepsilon = 0.01$ and an increment of 0.01;

2) DeepFool [120] is then used to analyze if the local linearisation approach stands also against a fine-tuned binary CNN;

3) Finally, in order to explore the effectiveness of an almost uninformed attack, we considered the evolutive approach introduced by [174]. In particular, since preliminary exploration demonstrated that the one-pixel attack is useless against the LD (at least under the imposed maximum number of iteration), in this work we removed the low-number modified pixels constraints, allowing the algorithm to modify up to 2000 points (that implies modifying $\leq 2\%$ of pixels in the worst case).

5.3.2 Fingerprints Adversarial Perturbations

As seen in section 1.3.1, in the contest of natural images $I \in \mathbb{R}^{(w,h,3)}$, given a classifier $f_C : I \rightarrow \{1..n\}$ mapping I to one of the possible n labels, an *adversarial perturbation* r is defined as

$$r \in \mathbb{R}^{(w,h,3)} : f_C(I) \neq f_C(I+r) \tag{5.1}$$

where r is usually required to be as little as possible (to be subtle for the human perception, as in figure 1.8). However, natural images and fingerprints are different in natures and thus applying adversarial perturbation approaches as are may not be effective (figure 5.6).

According to the adversarial perturbation idea, the injected noise should be as reduced as possible, to be almost invisible to human eyes. However, natural and fingerprint images differ for some characteristics, making this requirement harder to achieve with fingerprint images since: in the former, the

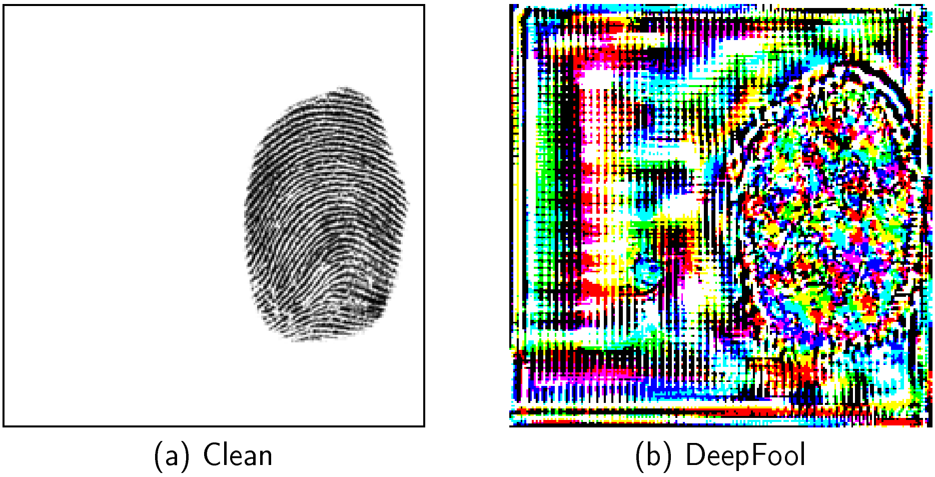


Figure 5.6: Example of using DeepFool [120] to generate the adversarial perturbation for a fingerprint from LivDet2015 [122] competition.

whole image carries information, usually in a colour (RGB) space; in the latter, the information is limited to a portion of the image (where the fingerprint is) and only the differences between background and the fingerprint are of interest (thus a grey-level like space). This implies that, even though an adversarial presentation attack would be able to fool the liveness detector, not only a simply masking or a threshold operation will destroy the attack, but a human operator will be definitively able to spot the attack.

Since we want to produce print-robust adversarial fingerprints, to take into account these differences, we constrained adversarial perturbation approaches:

- to inject a grey-level (i.e. the same for all the channels in the case of RGB acquisitions) noise $r \in [0,255]$
- to apply it only to the Region of Interest (ROI) delimiting the actual fingerprint [113].

If, on one hand, these constraints make the attack harder to perform, on the other are needed to conduct the experiments in a more realistic fashion.

5.3.3 Experimental Setup

As stated in section 5.2.1, the need for a common experimental protocol started the gathering of fingerprints datasets. In this work, we will make use of the one provided with LivDet2015 [122] competition. The variety of used sensor and of used spoofing material, the availability of well-defined training and test datasets (the latter produced also by using spoofing materials not available in the training) collected under clear circumstances, and the availability of open-source top-performer liveness detectors trained on it, are the main reasons motivating our choice.

Table 5.1 briefly reports the main characteristics of the LivDet2015 dataset. For each scanner, the table reports the size (in pixels) of the acquired fingerprint, and the number of live and fake fingerprints. The latter are grouped based on the used spoofing material. A hyphen in a cell indicates that the corresponding material has not been used to generate fake fingerprints for the corresponding scanner.

The FAS considered in this work consists of the cascade of following two sub-systems (see section 5.2 for details):

- The first to analyse the input fingerprint image is the *liveness detector* (LD). Aimed in detecting fake fingerprints, it is the first to analyze the input images. In this work we consider the approach proposed by the LivDet 2015 winner [126], based on a VGG19 CNN pre-trained on ImageNet and then fine-tuned on the fingerprint liveness detection task. The choice made by the authors was not only guided by the enthusiasm for deep learning, but the result of a precise analysis of its suitability. In particular, authors proved that their approach obtains better results when compared both to other CNN (AlexNet and an ad-hoc generated CNN were evaluated) and also to other non-deep state-of-the-art approaches [59], producing results able to outperform the runner-up. To prevent over-fitting, authors adopted dataset augmentation by extracting from

Scanner	Biometrika	CrossMatch	DigitalPersona	GreenBit
Image Size (px)	1000x1000	640x480	252x324	500x500
Live	1000	1500	1000	1000
Body Double	-	300	-	-
Ecoflex	250	270	250	250
Gelatine	250	300	250	250
Latex	250	-	250	250
Liquid Ecoflex	250	-	250	250
OOMOO	-	297	-	-
Playdoh	-	281	-	-
RTV	250	-	250	250
Woodglue	250	-	250	250

Table 5.1: LivDet2015 dataset characteristics. For each scanner, the acquired fingerprint size, and the number of live and fake fingerprints (for each spoofing materials) images are reported. The hyphen in a cell indicates that the corresponding material has not been used to generate fake fingerprints for the corresponding scanner.

each fingerprint five smaller images obtained considering the 80% of the original image from each corner and one at the centre. Each patch is then horizontally reflected, obtaining a final dataset 10 times bigger than the original one. Finally, simple resizing was adopted to fit fingerprints (whose size ranges from 252x324 pixels up to 1000x1000) to the VGG19 input layer (that expects three channels images with 224 pixels height and width).

- The second is the *authentication system* (AS), aimed in determining whether the biometry is valid for accessing the system. To this aim, it provides access if the pattern of ridges, furrows and minutiae on the surface of the presented fingerprint matches the pattern of a reference one. Fingerprint-based reliable authentication can be a challenging problem and its accuracy strongly depends on the image quality and on the fingerprint orientation. We consider as AS the work of [85], a hybrid AS that extract shape and orientation descriptor to filter false and unnatural minutiae pairings while exploiting ridge orientation as an adjunct parameter for the matching. This choice was guided by its closeness to the FBI standards [89] and by its common usage in literature as a baseline.

The liveness detector [126] was trained on the training set, using 5-fold cross-validation to determine the best number of training iterations and the augmentation technique proposed in the original paper; then, the adversarial attack was performed by using images from the test set. Only fake fingerprints have been considered since an attacker is usually interested in making fake replicas recognized as live. It is worth noting that the LD is not perfect, thus implying that some fake fingerprints are recognized as live. Therefore, in order to produce fair results, only fingerprint correctly recognized as fake were used for the adversarial perturbation attack. Finally, only perturbed images able to mislead the LD was submitted to the authentication system. Table 5.2 summarises, for each LivDet2015 fingerprint scanner, the number and relative percentage of correctly recognised and authorised fake fingerprints.

5.3.4 Adversarial Presentation Attack Results

This section reports the results for the fingerprint adversarial presentation attack, by varying the adversarial perturbation approach. Tables 5.3 to 5.5 report the results of the attack, for each sensor, for the FGSM, DeepFool and Evolutionary adversarial perturbation approaches respectively. Results show that DeepFool is the most effective in breaking the LD ($\sim$ 36.7% of average success rate), but the worst against the AS ($\sim$ 70% of average success rate). The FGSM based attack shows an average success rate of 2.3% for the LD, but injected perturbation preserves enough fingerprint characteristics to make it able to break the AS in about the 90.6% of the cases (on average).

Scanner	**#FF**	**#FF\#Fake**	**#Auth.**	**#Auth.\#Fake**
Biometrika	1435	95.66%	1058	73.73%
CrossMatch	1406	97.09%	1366	97.16%
DigitalPersona	1424	94.93%	1049	73.67%
GreenBit	1426	95.06%	1064	74.61%

Table 5.2: Results of a classical presentation attack (i.e. without the use of any perturbation technique to mislead the liveness detector). For each scanner, the Table reports: the number of fake fingerprints (and relative percentage with respect to the total number of available fingerprints in the dataset) correctly recognised as fake by the liveness detector (#FF); the number of fake fingerprints able to by-pass the authentication system (and the relative percentage with respect to the total number of fake fingerprints able to mislead the liveness detector).

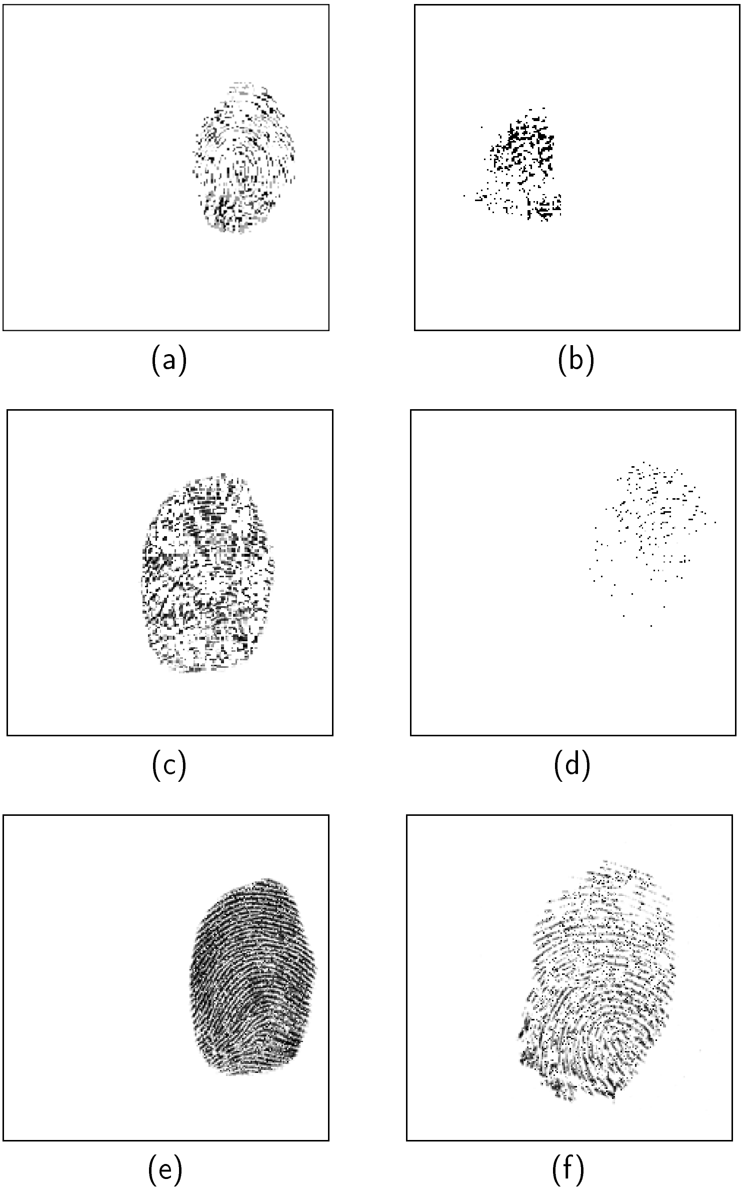
Although the Evolutionary attack performs only a little bit better w.r.t. the FGSM approach against the LD (excluding the GreenBit scanner, with a success rate of 65.79%), perturbed fingerprints are very likely able to fool the AS ($\sim 95.16\%$ of average success rate). For each attack, the column R in the same table reports the ratio between the number of successful adversarial perturbed fingerprints (against LD and AS) over the number of successful clean fake fingerprints (against only the AS), indicating the relative percentage of fake fingerprints that, despite the injected noise, are still able to break the AS. For the sake of completeness, table 5.2 reports, for each sensor, the result of a standard presentation attack (without any perturbation injected) in order to define a baseline, considering only correctly recognised fake fingerprints (as stated in session 5.3). It is worth noting that, although not reported here, during our experiment the considered AS demonstrated to be perfect in recognising users on the basis of live fingerprints. Finally, to better understand the attack effectiveness, figure 5.7 reports some examples of successful and unsuccessful adversarial attacks for each analysed adversarial perturbation approach. Among all the images, it is worth noting that the one produced by the Evolutionary attack appears to be the less perturbed one, resulting in a just slightly visible salt-pepper like noise (Figure 5.7-e).

Scanner	LD (%)	AS (%)	R (%)
Biometrika	2.67	86.84	3.15
CrossMatch	2.45	85.71	2.82
DigitalPersona	2.16	93.55	2.74
GreenBit	1.92	96.30	1.90

Table 5.3: Results of the adversarial attack, for each scanner, by using the FGSM adversarial perturbation approach. The Table reports the attack success rate against the Liveness Detector (LD%) and the Authentication System (AS% - evaluated with reference to the number of images that have passed the Liveness Detector) and the ratio (R%) between the number of successful adversarial perturbed fingerprints (against LD and AS) over the number of successful clean fake fingerprints (against only the AS).

Scanner	LD (%)	AS (%)	R (%)
Biometrika	62.15	73.67	62.15
CrossMatch	35.06	72.20	33.93
DigitalPersona	1.60	69.57	1.51
GreenBit	52.92	68.01	37.04

Table 5.4: Results of the adversarial attack, for each scanner, by using the DeepFool adversarial perturbation approach. The Table reports the attack success rate against the Liveness Detector (LD%) and the Authentication System (AS% - evaluated with reference to the number of images that have passed the Liveness Detector) and the ratio (R%) between the number of successful adversarial perturbed fingerprints (against LD and AS) over the number of successful clean fake fingerprints (against only the AS).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.7: Successful (left) and unsuccessful (right) adversarial perturbation attack on a fake fingerprint using FGSM (a-b), DeepFool (c-d) and the Evolutionary approach (e-f).

Scanner	LD (%)	AS (%)	R (%)
Biometrika	3.58	96.08	4.67
CrossMatch	7.22	91.26	8.83
DigitalPersona	8.78	94.44	11.25
GreenBit	65.79	98.16	66.47

Table 5.5: Results of the adversarial attack, for each scanner, by using the Evolutive adversarial perturbation approach. The Table reports the attack success rate against the Liveness Detector (LD%) and the Authentication System (AS% - evaluated with reference to the number of images that have passed the Liveness Detector) and the ratio (R%) between the number of successful adversarial perturbed fingerprints (against LD and AS) over the number of successful clean fake fingerprints (against only the AS).

The analysis of the three considered attacks (namely, FGSM, DeepFool and the Evolutionary one) shows that the FGSM approach is the less effective against the LD, but introduced perturbation is able to obtain good performance against the AS. On the other hand, DeepFool is the best in misleading the LD (excluding the GreenBit scanner), while the Evolutionary approach is the best against the AS: the result is that FGSM is the less effective considering the FAS as a whole, with DeepFool and the Evolutionary approach performing better on two sensors respectively. This is very interesting because despite FSGM and DeepFool had a perfect knowledge of the LD, they are not effective when considering the FAS as a whole, while the Evolutionary based attack does not only consider the FAS as a black-box, but it is also very robust to the settings and acquisition scanners variations, so demonstrating to be a good starting point for designing a black-box scanner-independent attack.

It is worth noting that, although this work must be considered as a preliminary step to the development of an *adversarial presentation attack*, results seem to support the feasibility of the proposed idea. In particular, the reduced number of successfully perturbed images by the Evolutionary approach, together with their high likely AS attack success rates, suggest that an appropriate tuning of the attack parameters (maximum number of iteration, population size and differential evolution parameters) is likely to have a strong impact on its effectiveness. However, further studies are needed to analyse the applicability of such an approach in a real-world scenario, taking into account different scanner, fake replicas made using unknown materials and so on.

5.4 Transferring Perturbation Attack

A common assumption in adversarial perturbation scenarios is that the attacker has white-box access to the target CNN and to the used dataset. Indeed, in section 5.3 we attacked the Vgg16-based liveness detector introduced in [126], showing how to effectively bypass it. In that case, the attacker had white-box access to the trained CNN used for the LD and limited knowledge on the dataset since some spoofing materials used for the test set are not used for the training.

However, in the biometrics authentication system context, this might be a stretch limiting the applicability of the attack. Therefore, in this section, we *want to make a step further by analysing whether it is possible to transfer a perturbation across different CNN liveness detectors in the case of a target LD very different from the one used to derive the perturbations*. In particular, we want the attacker having no clues about the target liveness detector, but that it is CNN-based. The knowledge the attacker has about the dataset is the same considered in our past experiment (i.e. the used fingerprint scanner and training spoofing materials). The aim is to perform a black-box perturbation attack against CNN fingerprint liveness detectors, analysing if and to what extent the scanner and the spoofing material combinations affect the success rate of the attack. Both points, to the best of our knowledge, have never been so far addressed.

As seen in section 5.3, misleading a CNN-bases liveness detector requires the attacker to modify a fake fingerprint such that it is recognised as real. Using adversarial perturbations to this aim in a black-box scenario implies the crafting of the adversarial fingerprint by using a CNN that is different from the one actually used by the target liveness detector. Thus, to analyse the transferability of adversarial perturbation in the fingerprint context, it is important i) to understand how to adapt adversarial perturbation approaches, ii) to define a fingerprints dataset and iii) to set the liveness detectors used for the experimentation. To face this scenario, we propose i) to use a "shadow" (i.e. ad-hoc crafted and trained by the attacker) CNN-based liveness detector to create the perturbations that ii) will be transferred to the fingerprint analysed by the target black-box liveness detector.

As showed by the LivDet competitions, over the years researches faced the liveness detection problem adopting increasingly sophisticated approaches, usually based on the available computer-vision state-of-the-art techniques. Two years later the end of LivDet2015 competition, a new approach raised the bar for the LivDet2015 dataset: Finferprint SpoofBuster [35]. Although itself based on the fine-tuning of a (more recent) CNN named MobileNet [79], the authors argue that resizing the fingerprint image as a whole, to match the net input size, introduces noise leading to severe information loss. Therefore, to face this, they introduce the idea of analysing local fingerprint's regions of 96 pixels height and width, each centred on minutiae (i.e. ridges and pores). Each region is then rotated based on the minutiae orientation and resized to match the CNN input layer. The network is then fine-tuned by using RMSProp optimiser [22] and a batch size of 100, after applying a data augmentation procedure. The probabilistic output from the so trained network is used as a "spoofness score" for the current region, with 0.5 used as a threshold between live and fake minutiae (Fig. 5.8).

At the time of writing this paper, the two described approaches represent the state-of-the-art for all LivDet competitions, since all the approaches that

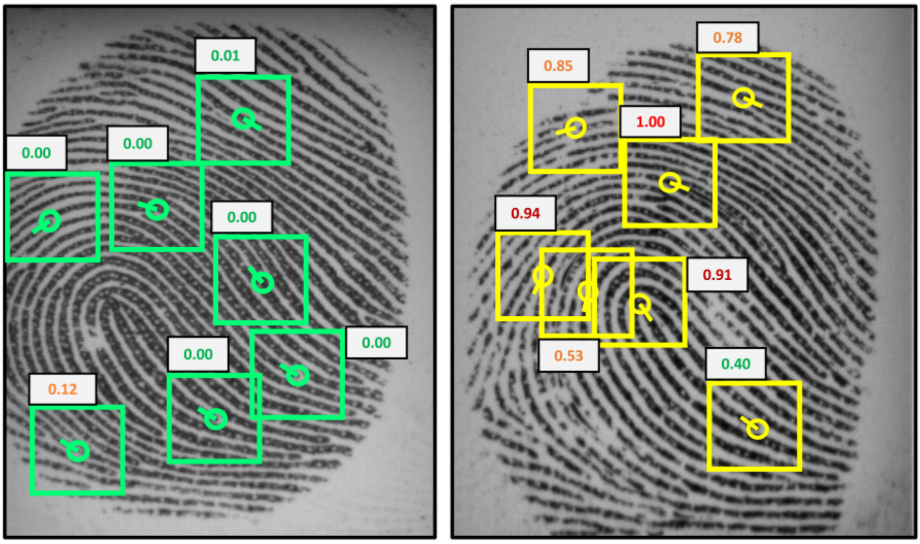


Figure 5.8: Example of minutiae based regions for a live (left) and for a fake (righ) fingerprint. The number represents the spoofness score. The circle with the line is the minutiae orientation. Image taken from the original SpoofBuster paper [35].

obtained better results are neither publicly available nor published. Therefore, in this work we consider SpoofBuster [35] as the target black-box liveness detector, and the Vgg16 based approach proposed in [126] as our shadow model.

It is worth to note that, in the described scenario, the attacker has no clues about the target liveness detector. Thus, the only viable approach for the attacker is to generate an adversarial noise having the same size as the fingerprint acquired by the scanner. However, since adversarial samples have the same size of the target CNN input layer, *there is the need for a stage to adapt the noise crafted by using the shadow liveness detector to the fingerprint scanner acquisition size*. Once the attacker creates the adversarial samples by using the shadow LD, there are two viable approaches to obtain an adversarial sample having the scanner output size (figure 5.9):

- **Image resize**, in which the attacker directly resizes the crafted adversarial sample;
- **Noise resize**, in which the attacker resizes only the adversarial noise, adding it to the original fingerprint acquired by the scanner.

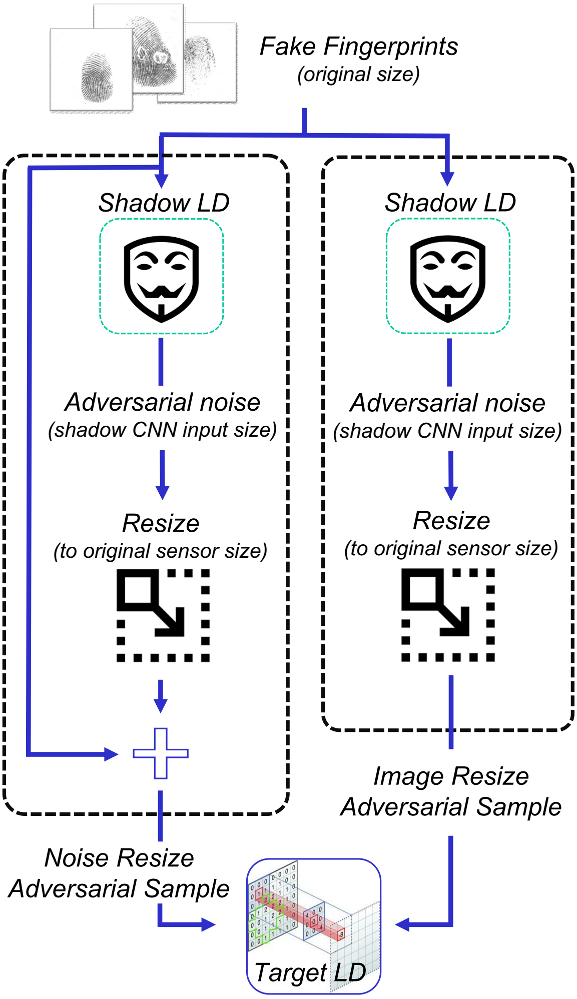


Figure 5.9: Transfer perturbation attack scenarios. Top: the image resize attack procedure. Bottom: the noise resize attack procedure.

Since there is no reason to prefer one above the other, we analyse the effectiveness of both procedures. To this aim, both the target and the shadow liveness detectors are trained in a 5-fold CV fashion on the official LivDet2015 training dataset. To evaluate the attack success rate, we consider only *fake* fingerprints form the official LivDet2015 test dataset (since an attacker is usually interested only in making fake replicas recognised as live). Finally, as none of the perturbation approaches so far developed demonstrated to be the most effective, we consider three different algorithms to craft the adversarial samples: iterative FGSM [97], with an initial standard deviation $\varepsilon = 0.01$ and an increment of 0.01; DeepFool [120]; OnePixel [174], modified to relax the low-number modified pixels constraint to allow modifying up to 2000 pixels (i.e. $\leq 2\%$ of the total number of pixels on the smallest fingerprint).

5.4.1 Experimental Results

Since the different sensor may differ a lot in terms of acquisition size (see table 5.1 for details), a different liveness detector (both target and shadow) is trained for each sensor. It is worth to note that since obtained LDs are not perfect (although closely matching the performance reported in the corresponding papers) some fake fingerprints in the test dataset are already recognized as live. Therefore, in order to produce fair results, *only fingerprint correctly recognized as fake by the target liveness detector* were used to evaluate the attack, obtaining 1472 (98.13%), 1436 (99.17%), 1445 (96.33%) and 1456 (97.06%) fingerprints for Biometrika, CrossMatch, DigitalPersona and GreenBit scanners respectively (refer to table 5.2 for the number of samples in each dataset).

Table 5.6 reports the transfer perturbation attack success rates obtained by using the “image resize” approach. Results show that, although there are some critic combinations for which the attack is ineffective, the effectiveness of the attack is related to the scanner more than to the used perturbation algorithm.

Scanner	FGSM	DeepFool	OnePixel
Biometrika	3,16%	2,94%	3,51%
CrossMatch	96,88%	97,82%	96,66%
DigitalPersona	3,51%	3,65%	0,28%
GreenBit	19,99%	20,97%	0,08%

Table 5.6: Transfer adversarial attack success probability against the target Liveness Detector under the "image resize" scenario, for each scanner and for each adversarial perturbation approach.

This behaviour is expected since, although all optical, each LivDet2015 scanner have different characteristics (e.g. sensor, lens, acquisition plate, etc.) that result in distinct artefacts in the acquired fingerprints [50]. A clear example of this is reported in figure 5.10, where two fingerprints from the same finger of the same user, but acquired by using two different scanners, are reported. It is interesting to note that the two scanners proven more robust against this attack are those having the smaller and the largest fingerprint sizes, while the weakest is the scanner for which training and test spoofing material did not overlap. Using a different sensor also affects the acquired fingerprint histogram (figure 5.11) due to the different ways each sensor acquires the light wavelengths. As a consequence, since a CNN essentially performs liveness detection by analysing fingerprints high-frequency texture details, some sensors may be more resilient to perturbation attacks.

Table 5.7 reports the success rates obtained in the "noise resize" scenario, with results clearly showing that the approach is not effective (except for a single scanner/algorithm combination). This is probably motivated by the fact that adversarial perturbation algorithms determine the noise to inject with respect to a target sample. Therefore, resizing only the noise probably causes the resulting image losing the details able to mislead the CNN-based LD.

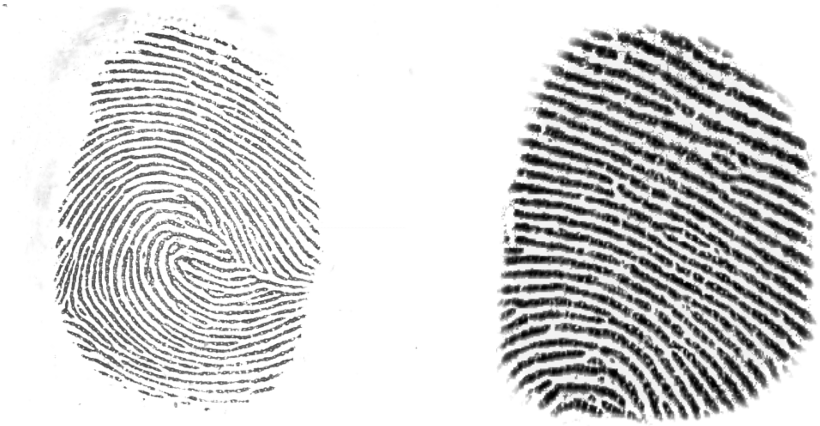


Figure 5.10: Example of different geometric distortions introduced by acquiring the same fingerprint by using two different scanners: GreenBit (left) and DigitalPersona (right).

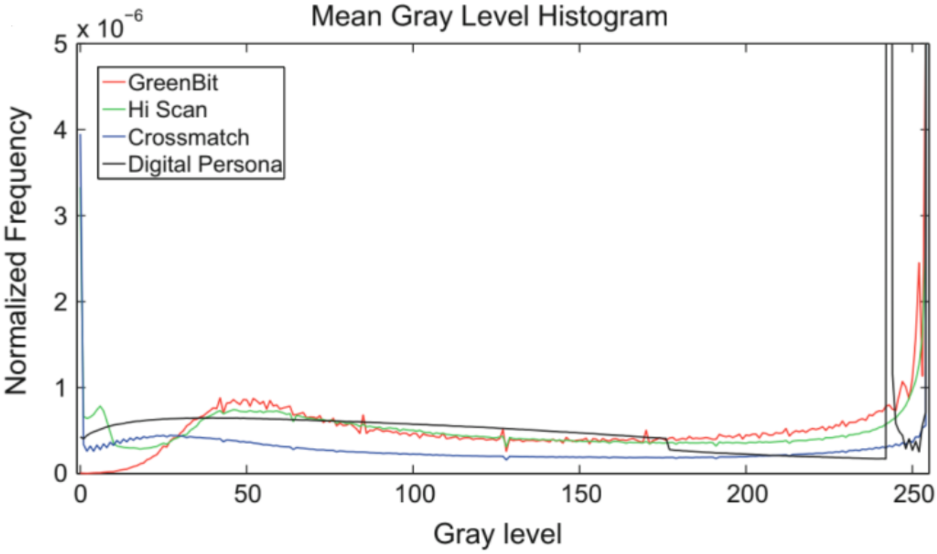


Figure 5.11: Mean histograms of greyscale for all the LivDet2015 scanner ("Hi Scan" refers to the Biometrika scanner). Image taken from [50].

Scanner	FGSM	DeepFool	OnePixel
Biometrika	1,08%	1,15%	0,00%
CrossMatch	0,07%	1,16%	99,93%
DigitalPersona	0,00%	0,00%	0,00%
GreenBit	0,00%	0,07%	0,00%

Table 5.7: Transfer adversarial attack success probability against the target Liveness Detector under the “noise resize” scenario, for each scanner and for each adversarial perturbation approach.

Proven “image resize” as the best strategy, to better understand to what extent the used spoofing material affect the liveness detection effectiveness, tables 5.8 to 5.11 report the success rates (one scanner per table), under the “image resize” scenario, grouped by spoofing material.

Scanner	FGSM	DeepFool	OnePixel
Ecoflex	0,40%	0,39%	0,00%
Gelatin	1,60%	1,62%	1,60%
Latex	0,80%	0,81%	0,80%
Liquid Ecoflex	0,00%	0,00%	0,80%
RTV	6,41%	19,60%	6,01%
Wood Glue	8,40%	8,00%	8,40%

Table 5.8: Biometrika scanner per-material transfer adversarial attack success probability against the target Liveness Detector under the “image resize” scenario, for each adversarial perturbation approach.

Scanner	**FGSM**	**DeepFool**	**OnePixel**
Body Double	96,65%	96,68%	96,67%
Ecoflex	98,52%	98,89%	98,53%
Gelatin	90,33%	91,67%	90,33%
OOMOO	94,24%	94,95%	94,28%
Playdoh	79,36%	80,43%	80,07%

Table 5.9: Crossmatch scanner per-material transfer adversarial attack success probability against the target Liveness Detector under the “image resize” scenario, for each adversarial perturbation approach.

Scanner	**FGSM**	**DeepFool**	**OnePixel**
Ecoflex	0,80%	0,81%	0,00%
Gelatin	0,80%	0,82%	0,00%
Latex	0,40%	0,42%	0,00%
Liquid Ecoflex	9,60%	9,62%	1,60%
RTV	0,80%	0,79%	0,00%
Wood Glue	7,60%	7,61%	0,00%

Table 5.10: Digital Persona scanner per-material transfer adversarial attack success probability against the target Liveness Detector under the “image resize” scenario, for each adversarial perturbation approach.

Scanner	FGSM	DeepFool	OnePixel
Ecoflex	14,80%	15,60%	0,00%
Gelatin	2,80%	2,82%	0,00%
Latex	10,80%	11,20%	0,00%
Liquid Ecoflex	31,20%	30,41%	0,00%
RTV	16,40%	19,60%	0,00%
Wood Glue	37,60%	40,41%	0,40%

Table 5.11: Green Bit scanner per-material transfer adversarial attack success probability against the target Liveness Detector under the “image resize” scenario, for each adversarial perturbation approach.

Results shows that there is no a single substance always performing the best, but rather that each scanner shows a different robustness degree against different materials. Together, these results seem to suggest that the combination of the fingerprint scanner and spoofing material is crucial for the success of the attack. This result is expected since the different characteristics of fingerprint scanners affects the texture of the acquired images.

To better understand the result of adversarial presentation attack transfer, figures 5.12 and 5.13 report some examples of successful and unsuccessful attacks for each analysed adversarial perturbation approach.

As a final consideration, we highlight that an attack against a liveness detector should be performed in *reasonable* time to be considered useful (e.g. because of security policies timing). For the target liveness detector considered in this work (SpoofBuster), this point is even more critic since it analyses a fingerprint on local patches, making the white-box attack we introduced in section 5.3 extremely time-consuming. In this case, transferring the adversarial perturbations from a CNN easier (less time-consuming) to

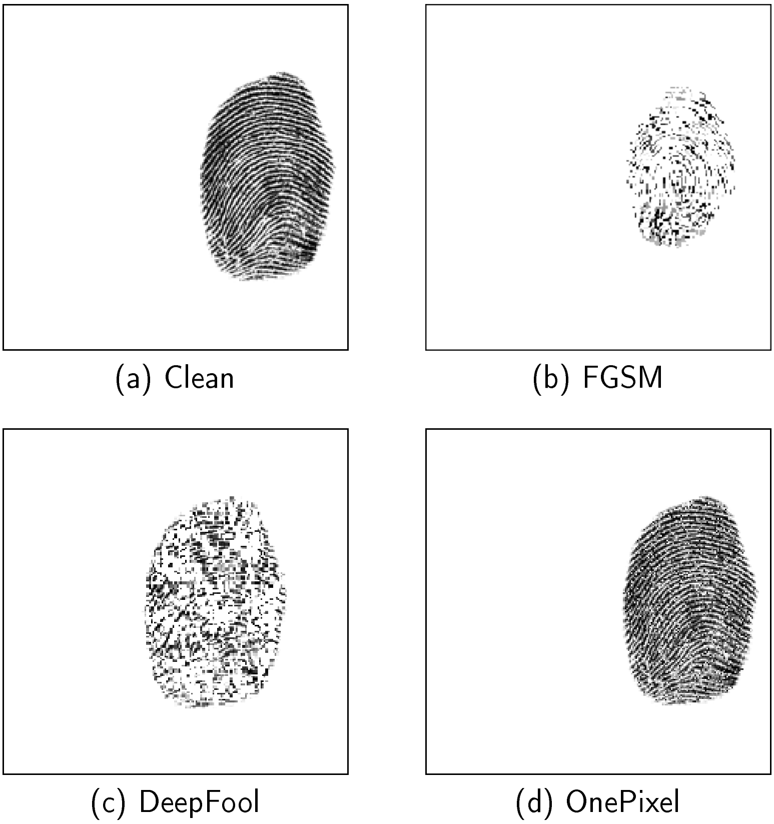
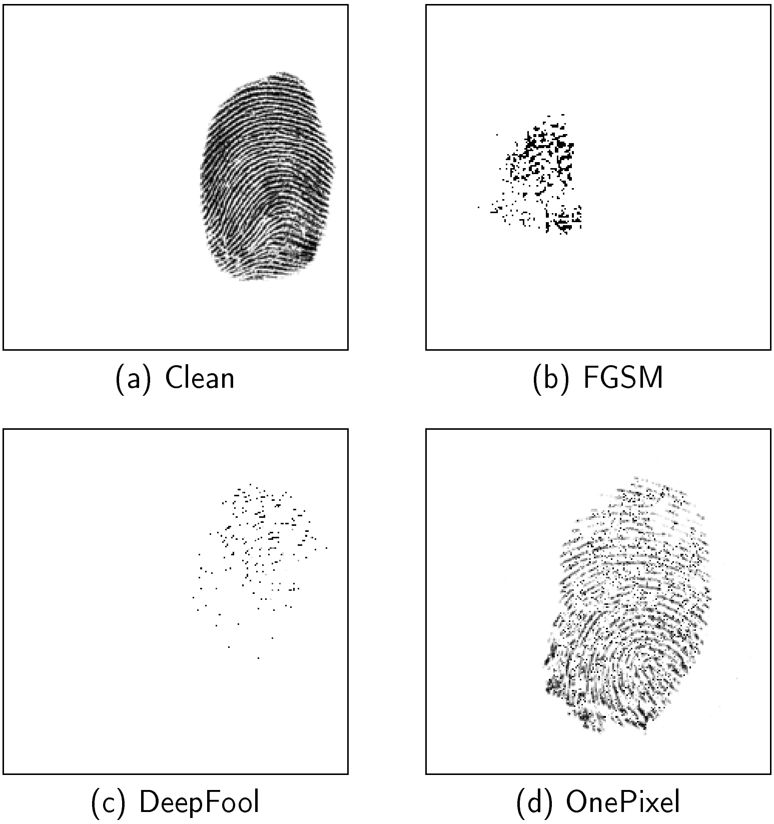


Figure 5.12: Successful adversarial perturbation attacks on a clean fingerprint (a), using FGSM (b), DeepFool (c) and OnePixel (d) under the "image resize" scenario. Please note that no successful example exist for the FGSM approach.

attack, to a harder one, might be the best choice. To sustain this claim, we run both the original white-box attack and the introduced transfer attack by using a server hosted in our HPC facility [23] equipped with 2 x Intel(R) Xeon(R) Intel(R) 2.13GHz CPUs (4 cores each), 32GB RAM and an Nvidia Titan Xp GPU having 12GB DDR5 GRAM. For the white-box, the computation needed about 576 hours, while the "image resize" transfer attack took only about 33.5 hours ($\sim 17x$ times faster).

[23] www.scope.unina.it


(a) Clean

(b) FGSM

(c) DeepFool

(d) OnePixel

Figure 5.13: Unsuccessful adversarial perturbation attacks on a clean fingerprint (a), using FGSM (b), DeepFool (c) and OnePixel (d) under the “image resize” scenario. Please note that, for visualization reasons, the fingerprint used for the OnePixel example is different from the one used in the other cases.

In conclusion, *this work is a first proof-of-concept showing the viability of adversarial transfer perturbations against CNN-based liveness detectors under certain combinations of fingerprint scanner and used spoofing material.*

6 Fairness and Privacy in Face Analysis

The improvements in artificial intelligence, sustained by the rise of the big data paradigm and of social media, allowed the spread of several *smart assistants* in our daily life activities. Tools that appears to be totally harmless, such as automatic image tagging (e.g. those based on face recognition), voice control and personalised advertising, process enormous amounts of data (often remotely due to the huge computational effort required) rich in sensitive information. Literature is full of approaches aimed at exposing subject privacy, with several researchers trying to develop defensive strategies in the view of a secure and private AI [162, 14].

However, those privacy threats are usually perceived as far from us or, even worse, able to affect only our digital alter-ego. Unfortunately, this is a false belief. Indeed, especially when it comes to derive subjects' soft biometrics from data spontaneously published by target users (e.g. a profile picture on a social media, a blog post, a product review, etc.), there are many very effective and sneaky (i.e. not perceived by users) attacks able to extract our soft biometrics with a single glance in a real environment, maybe also without our explicit consensus [93, 31] . The impact that privacy-related issues could have on our life is becoming more and more severe as AI improves. In recent years, face analysis is becoming a central theme for subject privacy and fairness with AI. Indeed, although face recognition was a task already addressed in a successful way before the advent of deep learning [193], it was only with CNNs that researchers were able to “close the gap with respect to human-level performance” [182, 175, 134].

Face recognition and analysis is a particularly sensitive topic even without involving AI. Indeed, it has been proven that attractive people tends to get more financial and social benefit [106]. Despite apparently related to humans social and evolutionary psychology, “lookism” [195] (the term used to describe this behaviour) is extrimely controversial, especially when it results in discriminative decisions. With the increasing use if AI for face analysis, could there any risk to fell for a similar affair? According to a very recent scandal[24] the concerns seem to be more realistc than expected.

As already showed in chapter 2, the problem is not in artificial intelligence, but in the way we, as humans, make use it. However, with AI increasing usage in everyday tool (smartphone, ads, loan, etc.), how can we be sure not only that our privacy information will be not disclosed, but that they will not be used against us? Therefore, in this chapter we investigate whether it is possible to exploit adversarial perturbations (see section 1.3.1) as a mean to protect against extraction of sensitive information from face images, our most disclosed identity biometrics.

[24] https://www.washingtonpost.com/technology/2019/10/22/ai-hiring-face-scanning-algorithm-increasingly-decides-whether-you-deserve-job/

6.1 Defending Against Face Soft-Biometrics

As seen in this chapter beginning, the misuse of AI when it comes to privacy concerning context may results in issues that could impact our real life when users lose control of their data. With the spread of facial recognition and of all the soft-biometrics information associated with it (see section 5.1 for details), having concerns about a (potentially malicious) misuse of deep learning based face analysis techniques is no so conspirational[25].

However, as already seen in section 1.3.1, CNNs have blindspots that can be leveraged to mislead human biometrics analysers (section 5.2). Therefore, the potential vulnerability of CNNs to adversarial perturbations could open new possibilities for privacy protection by exploiting them to create an object that can be used in the real world to fool automatic soft biometrics systems based on CNNs. *The aim of this section is thus to understand if and to what extent adversarial perturbation can be used to protect subjects against unwanted soft biometric detection by automatic means.*

The idea is to create an adversarial patch [25] that, once printed, is able to fool CNNs by simply "wearing" it, in the shape, for example, of a sticker, a clip or a pendant. By definition, an adversarial perturbation should be as invisible as possible and this constraint is usually met since the injected noise is distributed over the whole image. On the contrary, in our case, it is strongly preferable to trade a very visible perturbation in exchange for having the opportunity to apply it on a very limited portion of the image.

Among all the soft biometrics, probably gender and ethnicity are the most crucial ones, mainly because of their past bad experiences [153, 39]. In this work, as a case of study, we will focus on the generation of a *general adversarial patch* specifically designed to fool a CNN for ethnicity recognition

[25] https://www.telegraph.co.uk/technology/2019/09/20/government-ai-rules-require-diverse-teams-prevent-racist-sexist/

in a real-world application. In particular, the aim is to create a patch that works across different subjects, including those never seen by the CNN.

6.1.1 CNNs for Ethnicity Classification

As described in the beginning of the chapter, although face recognition was a task already addressed in a successful way before the advent of deep learning [193], it was only with CNNs that researchers were able to "close the gap with respect to human-level performance" [182, 175, 134]. Thanks to the availability of large labeled dataset, in the last years researchers started to explore the effectiveness of CNNs for soft biometrics detection, both by using ad-hoc architectures [171, 114] or by fine-tuning a pre-trained CNN [124].

In this work we follow the latter approach, by fine-tuning an ImageNet [152] pre-trained Vgg16 [166] CNN on the ethnicity classification task. This choice was guided by some preliminary experiments that showed the higher effectiveness of fine-tuning w.r.t. training from scratch. Since Vgg16 was intended to face a thousand classes problem, we replaced the classification layer with one having the desired number of classes (ethnicity to be recognized) before performing a *re-training* of all the layers to adapt the network to the new classification task.

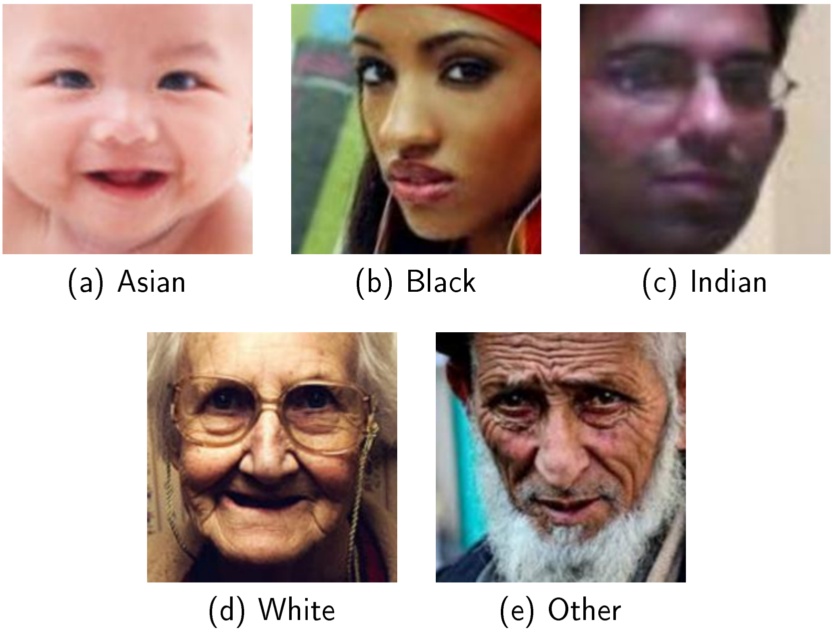
As dataset we considered UTKFace[26] [207], a publicly available large-scale face dataset containing over $20,000$ images with annotations (obtained with the DEX algorithm [150] and double checked by a human annotator) of age (ranging from 0 to 116 years old), gender and ethnicity (Asian, Black, Indian, White, Others), covering large variation in pose, facial expression, illumination, occlusion, resolution, etc. Although it is divided into five different ethnic clusters (table 6.1), in this work, as a case of study, we chose to focus only on the ’Black’ vs. ’White’ task.

[26] https://susanqq.github.io/UTKFace/

Ethnicity	Male Count	Female Count	Avg. Age
Asian	1575	1859	26
Black	2319	2209	34
Indian	2261	1715	32
White	5477	4601	38
Others	760	932	23

Table 6.1: UTKFace dataset characteristics.

It is worth noticing that *we did not impose any further restriction on subject age, pose, expression, illumination and occlusion, in order to obtain a model and an adversarial patch able to work in real environmental conditions*. To show the differences in resolution, illumination, position, etc., in figure 6.1 we report some samples extracted from the UTKFace dataset.


(a) Asian (b) Black (c) Indian

(d) White (e) Other

Figure 6.1: Images from the UTKFace dataset. Please note the variety of pose, illumination, age, resolution, expression and accessories.

To train the Vgg16 model, images were randomly divided into a training and a test set (80/20); following past similar works, the ADAM optimizer [88] was used, with a batch size of 32 and a $5 * 10^{-4}$ L2 regularization term. Training was performed for 10 epochs, with initial learning rate set to 10^{-3} and a step-wise decay strategy of 0.1 each 2 epochs. Images do not undergo any kind of pre-processing: a simple image resizing is performed to match the Vgg16 expected input height and width ($224 * 224$). Under these settings, we obtained a CNN able to determine the subject ethnicity with an accuracy, on the test dataset, of 95.59%.

6.1.2 Face Global Adversarial Patch

As stated in section 1.3.1, several adversarial perturbation attacks was so far proposed. However, to the best of our knowledge, adversarial perturbation was never used as a method to mask a subject ethnicity (nor any other soft biometry). The designed iterative procedure, intended to obtain the general adversarial patch invariant to position and subject, consists in the following steps (synthesized by the Algorithm 1):

1. A mask is generated to force the perturbation algorithm to work only in a restricted region of the image

2. An adversarial perturbation is determined for the first image, over the previously generated mask

3. The second point is repeated for all the remaining images, by starting, for each image, from the perturbation calculated over the immediately preceding image

4. The mask is randomly moved in order to generate an adversarial patch invariant to its position

5. Steps 3 and 4 are repeated (including also the first image) until the perturbation is able to tamper all the images ethnicity, or until a termination condition (such as the maximum number of iterations) was met

It is worth noting that it is not strictly needed that the perturbation algorithm is able to determine a good patch for each image in a single attempt, since the same image is seen many times. As a consequence, the maximum number of allowed iteration has to be tuned according to the desired performance level. Moreover, the proposed schema is totally independent of the chosen mask shape or perturbation algorithm: the first can be freely selected to match the desired object shape (for example a circle for a pendant); for the latter, the only strict requirement is to force the algorithm to inject only integer noise in the range $[0, 255]$, in order to make the obtained patch printable.

```
mask = createMask();
pert = randPert(mask);
pertCount, iterCount = 0;
while pertCount $\leq$ ths & iterCount $\leq$ maxIter do
    pertCount = 0;
    for Image img in Dataset do
        mask,pert = randomMove(mask,pert);
        pert = calculatePert(pert, mask, img);
        if classify(img) $\neq$ classify(img + pert) then
            pertCount++;
        end
        maxIter++;
    end
end
```

Algorithm 1: General adversarial patch creation. Please note that the maxIter and ths (threshold) values were not set to highlight that they are user-defined parameters.

6.1.3 Experimental Results

The adversarial presentation attack was performed using images from the test set. In particular, we consider only images whose ethnicity was correctly classified in the absence of perturbation, in order to produce fair results not influenced by false positives. We used DeepFool [120] as adversarial perturbation algorithm; the maximum iteration value was set to 500, while the mask was shaped in a circle. We performed the patch generation separately for black and white subjects; MATLAB R2018a was used to perform the experiments.

Table 6.2 reports the success rates for each ethnicity, by varying the mask size, showing that the approach is effective also when the mask size is reduced. It is interesting to note that the performance drop associated with the mask size is more significant for black subjects: this could be due to many factor, including problems related to a smaller diversity of the dataset (as reported in table 6.1, the number of black subjects is almost the half of white ones) or, on the contrary, to a greater variation between black subject images (that resulted in a trained CNN with a higher generalization ability in classifying black individuals).

Ethnicity	Circle Radius		
	10	15	20
Black	67.81%	88.56%	95.75%
White	97.49%	98.82%	99.95%

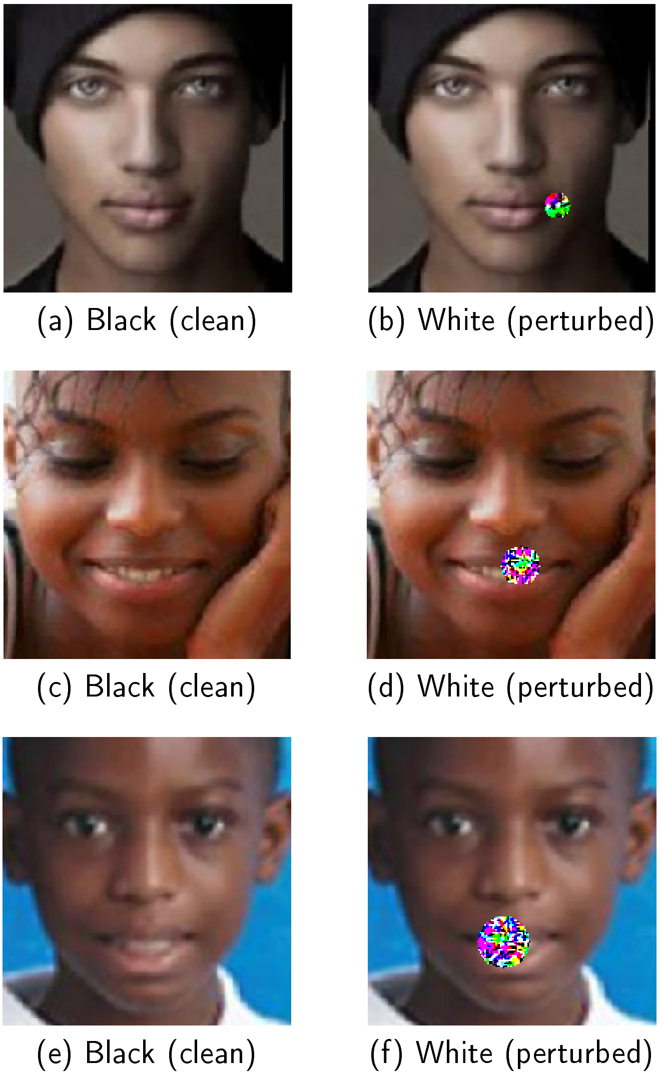
Table 6.2: Success rates of the general adversarial patch for the black vs white classification task, as the patch radius varies (in pixel).

Figures 6.2 and 6.3 report some examples of images with and without the adversarial perturbation applied, respectively for black and for white subjects, as the mask size varies. Finally, figure 6.4 reports the effect of evaluating the patch on images acquired in real-time with a laptop webcam, while figure 6.5 reports the effect of printing the patch and acquiring the resulting perturbed printed image with a webcam.

It is worth noting that although the idea of creating an adversarial patch that, once printed, is able to fool CNNs in a real context is not new [25], nor is new to apply it on face images [161], this work makes a step further because:

- We do not impose constraints on the patch position, making it insensible to its positioning (i.e. on a cap, as a pendant, etc.)
- We do not perform any kind of pre-processing, making the patch invariant to gender, age, illumination, expressions, etc., i.e. all conditions that usually happen in a real application
- We performed an analysis of the impact that the patch size has on the approach effectiveness
- We do not impose any constraint on the used adversarial perturbation algorithm, under the only condition to inject an integer, printable, noise
- Our approach allows us to generate a patch that works for several different subjects, so that it is possible to calculate it once and apply it many times

As shown in Figures 6.2, 6.3 and 6.5, despite the fact that for very small-sized patch the performance drops in the case of black individuals, the patch always occupies just a very reduced portion of the input image, so that it is possible to use it in a real scenario as an accessory. Moreover, it is worth noticing that the binary ethnicity problem has been chosen just as a proof-of-concept: the proposed approach is general and applicable to any kind of soft biometric and CNN.



(a) Black (clean)

(b) White (perturbed)

(c) Black (clean)

(d) White (perturbed)

(e) Black (clean)

(f) White (perturbed)

Figure 6.2: Example of the general adversarial path effects on same images of black subjects from the UTKFace dataset. Left columns, clean (unperturbed) images; right column, the same images with a 10 (d), 15 (e), 20 (d) pixels radius patch applied. Please note the variety of pose, illumination, age, resolution and expression.

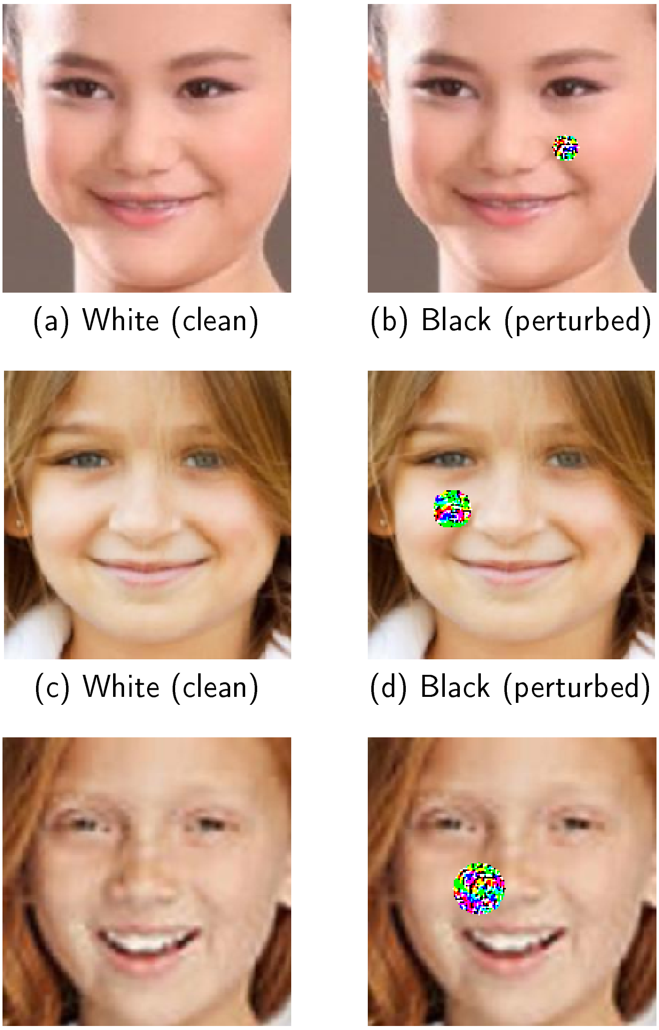
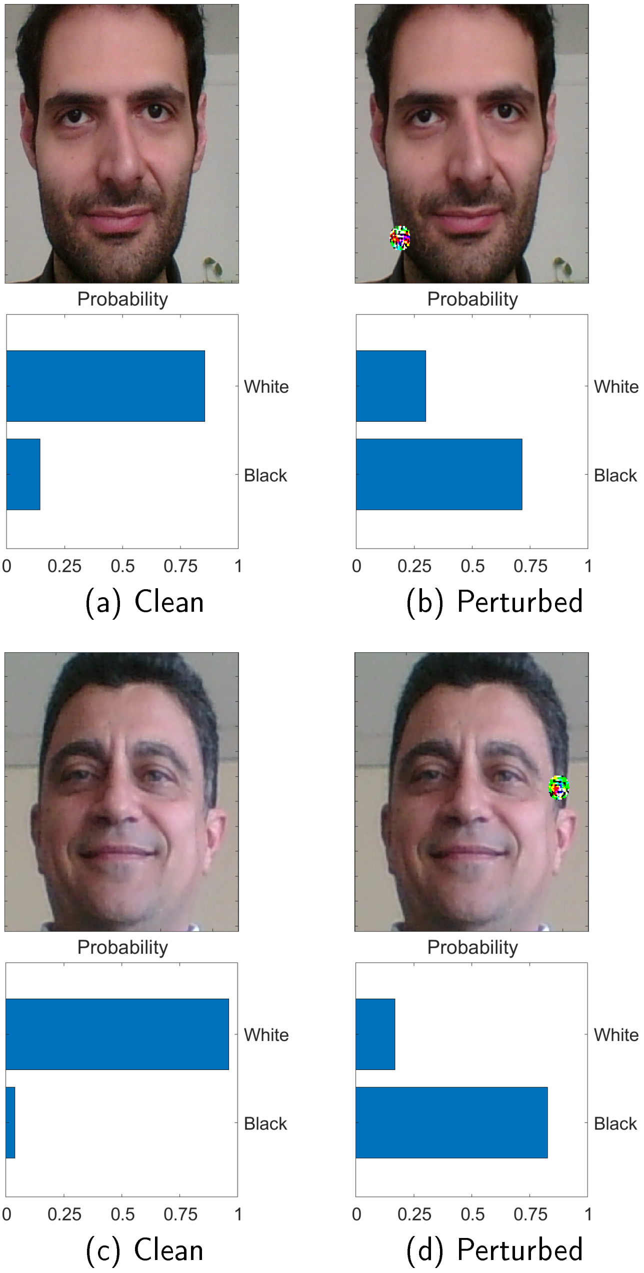

(a) White (clean)
(b) Black (perturbed)
(c) White (clean)
(d) Black (perturbed)
(e) White (clean)
(f) Black (perturbed)

Figure 6.3: Example of the general adversarial patch effects on the same images of white subjects from the UTKFace dataset. Left column, clean (unperturbed) images; right column, the same images with a 10 (d), 15 (e), 20 (d) pixels radius patch applied. Please note the variety of pose, illumination, age, resolution and expression.



(a) Clean

(b) Perturbed

(c) Clean

(d) Perturbed

Figure 6.4: Example of the effectiveness of the adversarial patch on authors' images acquired using a laptop webcam: the top row reports clean (a, c) and resulting perturbed (b, d) images, while the bottom row reports the associated probabilities.

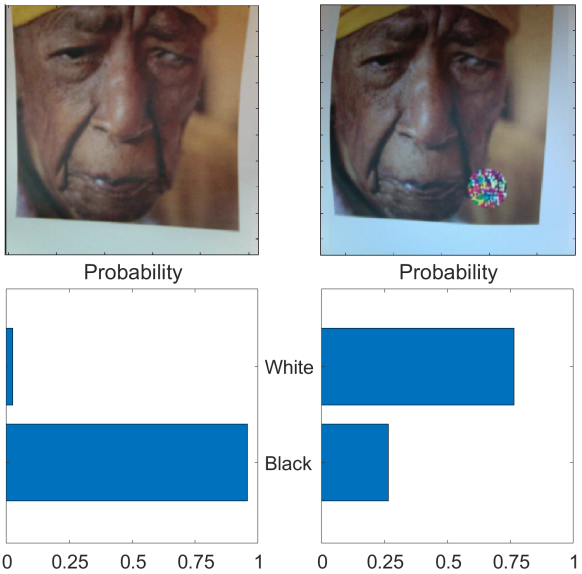


Figure 6.5: Example of the effectiveness of the adversarial patch after printing: on the left, the clean image is printed and classified by using a laptop webcam; on the right, image with the patch is printed and classified by using a laptop webcam.

Discussions and Open Issues

The spread of artificial intelligence in critical domains (e.g. facial recognition, biometric verification, autonomous driving, etc.) rises questions related to the consequences that its misuse (malicious or not) can lead to, such as unethical or unfair decisions (e.g. discriminating on the basis of ethnicity or gender) as well as violating people's privacy. Trough the text we have several times highlighted that AI is not to blame since, being just a tool, the consequences resulting from its misuses can not be accounted to the medium, but must be instead attributed to its operator.

Nonetheless, we claim that in order to develop a more ethical, fair and secure use of artificial intelligence, all the involved actors (in primis users, developers and legislators) must have a very clear idea about some critical questions, such as *"what is AI?"*, *"what are the ethical implications of it improper usage?"*, *"what are its capabilities and limits?"*, *"is it safe to use AI in critical domains?"*, and so on. Moreover, since AI is very likely to be an important part of our everyday life in the very next future, it is crucial to build trustworthy AI systems.

Therefore, in this thesis we tried to make a step towards the crucial need for raising awareness about security, ethical and fairness threats associated with AI systems, from a technical perspective as well as from the governance and from the ethical point of view. To this aim, this thesis is divided into three main parts: in part I (chapters 1 and 2) we introduced the concept of AI and the related ethical implications; in part II (chapters 3 to 4) we presented some crucial issues associated with the use of deep learning, showing how a proper network design can be used to limit their effects; in part III (chapter 5 and 6) we addressed the implications that an AI misuse can cause in a critical domain such as biometrics, proposing attacks properly intended for the aim. In particular:

- in Chapter 1, with the aim of providing the reader with the basic concepts needed to fully understand our contribution, we introduced the concept of Artificial Intelligence, illustrating the differences between shallow and deep architecture, motivating the reasons making Deep Learning bloom only in the last years, finally highlighting the burden associated with the use of deep neural networks. In particular, we introduced "transfer learning", showing how this approach can be effectively used to leverage deep neural networks (consisting of millions of parameters to optimise) also in task with limited amounts of labelled samples. We also introduced "adversarial perturbations", a term referring to the techniques intended to deceive AI systems by injecting a small perturbation (noise, often totally imperceptible to the human being) into the data. This represents the cornerstone of the whole thesis, since although adversarial perturbations are a considerable concern to domain experts, on the other hand, the fuels new possibilities to both favour a fair use of artificial intelligence systems and to better understand the "reasoning" they follow in order to reach the solution of a given problem. Therefore, the chapter first introduces the concept both from a mathematical and from the effects of such attacks, and then provide a review of some of the most famous approaches so far proposed, grouped on the basis of the exploited characteristics or on the basis of the obtained effects;

- in Chapter 2 we focused on the AI ethical aspects, first by reporting some detrimental (and often unintended) consequences that could arise with DL misuses, second by exposing and motivating our personal point of view. As can be seen from the reported examples, it is no wonder that reports of discriminatory effects in data-driven applications litter the news. On the other hand, the same examples also show that the problem is in humans not properly training AI models or maliciously teach them the worst of our mind. Essentially, AI is neither good or evil. It is just a tool designed to learn from example, also in the case of biased (e.g. racist) labels. The other question arising is whether we should adapt the notion of ethics to take into account decisions (totally or partially) made by human artefacts. Our opinion is that the most effective way to deal with ethics in machine learning is to consider the humans and the AI agents as a strictly coupled entity. This can allow to actively provide the system (human + machine) ethical judgement, to closely monitor for problematic emergent behaviours, and to be prepared to quickly react when problems arise. It is worth noticing that Asimov reached the same results many years ago: indeed, although Asimov doesn't mention Kant or refer to the word "deontological" anywhere in his works, it is clear from their formulation that the three robotic laws are Kantian in spirit, in the sense that they are universal and context independent;

- Chapter 3 is centred on the reproducibility of results, a crucial aspect of any scientific research. Focusing in particular on deep neural networks, we shed some lights on the intrinsic lack of determinism associated with the use of cuDNN, the NVIDIA library for GPU accelerated deep learning used (at the time of writing the thesis) in frameworks. Since, despite the workaround proposed by NVIDIA, the problem could unexpectedly arise during experimentation, especially in critical domains it is of crucial importance to take this into account and to put into practice all the means needed to guarantee a fail-safe situation. To this aim, we investigated the reproducibility of deep learning across different hardware and software configurations, both at training and at inference time. This reproducibility assessment is important in order to determine whether the resulting model produces good or bad outcomes just because of luckier or blunter environmental training conditions. As a case of study, we considered a biomedical image processing problem for the consequences associated with a misdiagnosed patient. Results show that the reproducibility issue can be effectively shifted from a strictly combinatorial problem to a statistical one, in order to validate the model robustness and stability more than its perfect outcomes predictability. Thus, in order to avoid providing not totally reproducible claims, it is very important to shifts the attention from a pure performance point-of-view to a statistical validity of the obtained outcomes. Indeed, a model showing large variations in results will have wider confidence intervals with respect to a more stable, and thus reproducible, one;

- Chapter 4 focused on the concept of *approximate computing*, a field involving the study of resilience (i.e. the ability of a system to provide correct results also in the presence of degraded working conditions) to reduce the resources needed by a system. The chapter introduced the concept of “sizing a CNN”, namely a procedure intended in removing some neurons to reduce the number of trainable parameters, in the context of fine-tuning. The aim is to show how a naive use of deep neural networks might not be the best solution. Indeed, although preliminary, results not only show that it is possible to reduce the number of parameters and memory usage without statistically affecting the performance, but also that the obtained network is more robust against adversarial perturbations;

- Chapter 5 aims to raise domain experts awareness on the potential consequences associated with the use of deep learning in security critical domain, such as biometric authentication systems. To this aim, after introducing the concept of biometrics, of authentication system, of presentation attacks and of liveness detection, the chapter proposed two attacks against CNN based authentication systems. The core idea is to exploit adversarial perturbations to modify fake fingerprints enough to be recognised as real by a liveness detector while preserving the user characteristics needed to bypass the authentication system. Since natural and fingerprint images are different, the considered adversarial perturbations approaches had been suitably adapted. Moreover, in order to design an attack procedure usable in a real-world scenario, we also propose some constraints to make the generated fake fingerprint printable. It is worth noticing that the need to preserve fingerprint key authentication characteristics while injecting a perturbation able to mislead the liveness detector is a way of exploiting adversarial perturbation that, to the best of our knowledge, has never been so far proposed. Moreover, with the aim of performing a more realist attack, we also introduce the concept of shadow model, indicating a liveness detector trained by the attacker for the sole purpose of generating the adversarial samples to submit to the actual target LD. All the results and experiments had been performed by using a public fingerprint dataset, considering as target CNNs its current top performers;

- in Chapter 6 we explored the insidious problem of protecting user privacy when it comes to derive subjects' soft biometrics from data spontaneously published by target users (e.g. a profile picture on a social media, a blog post, a product review, etc.). As a case of study, we focused on subject ethnicity detection based on the analysis of a face picture. To address the problem, we used adversarial perturbations to create a patch that trades visibility for a wide user usability.

It should be clear that several are the issues that must be faced, such as: designing systems that analyse people data ensuring privacy by default; analysing the limitations and the weaknesses that might affect an AI-based system, independently from the particular adopted technology or technical solutions; assessing the behaviours in the case of successful attacks and/or in presence of degraded environmental conditions; etc. Indeed, if on one hand, the industry is pushing toward a massive use of artificial intelligence enhanced solution, on the other it is not adequately supporting researches in end-to-end understating of capabilities and vulnerabilities of such systems. The results may be very (negatively) mediatic, especially when regarding borderline domains related to subjects privacy, ethics and fairness, such as users profiling, fake news generation and reliability of autonomous driving systems.

As shown, since AI is extremely pervasive in our life, there is a high risk that the choices made by using such models may have a significant impact on society. Therefore, it is becoming more and more crucial to quickly understand how to properly regulate artificial intelligence [197]. But, is the legislator able to cope it? The solution is not straightforward, not only due to the difficulties arising trying to put in practice AI policies [94], but also because it is a problem that must be addressed internationally, and not on a local scale.

Unfortunately, this is a very hard matter, since opinions about it are extremely discordant even within the same country. For example, in the USA, on one hand, FBI claims that their AI algorithms are effective and reliable to the point of being usable as scientific evidence[27], on the other Google is

[27] https://www.propublica.org/article/with-photo-analysis-fbi-lab-continues-shaky-forensic-science-practices

pushing toward the development of a suitable AI regulation[28]. In Europe, the situation appears a little more uniform, mainly thanks to General Data Protection Regulation (GDPR), a document through which the European Parliament has proposed, in 2016, a set of rules to regulate the activity of any company operating with data belonging to citizens from any European country [194].

In conclusion, AI represents without any doubt one of the greatest achievement made by humans. It has the power of really changing our world and to help people, even more than fire and electricity did[29]. However, since "with great power comes great responsibility", we must learn how to properly use it, developing methods and enacting laws that support its fair, secure and ethical usage for all people around the world.

[28] https://www.ft.com/content/3467659a-386d-11ea-ac3c-f68c10993b04

[29] https://www.cnbc.com/2018/02/01/google-ceo-sundar-pichai-ai-is-more-important-than-fire-electricity.html

Bibliography

[1] Aamoth, D. (2014). Interview with eugene goostman, the fake kid who passed the turing test. *Time. Disponible en.*

[2] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318.

[3] Abdelwhab, A. and Viriri, S. (2018). A survey on soft biometrics for human identification. In *Machine Learning and Biometrics*. IntechOpen.

[4] Ailisto, H., Vildjiounaite, E., Lindholm, M., Mäkelä, S.-M., and Peltola, J. (2006). Soft biometrics—combining body weight and fat measurements with fingerprint biometrics. *Pattern Recognition Letters*, 27(5):325–334.

[5] Akhtar, Z., Micheloni, C., and Foresti, G. L. (2015). Biometric liveness detection: Challenges and research opportunities. *IEEE Security & Privacy*, 13(5):63–72.

[6] Ali, S., Rauf, A., Islam, N., Farman, H., and Khan, S. (2017). User profiling: A privacy issue in online public network. *Sindh University Research Journal-SURJ (Science Series)*, 49(1).

[7] Amato, F., Barbareschi, M., Cozzolino, G., Mazzeo, A., Mazzocca, N., and Tammaro, A. (2017a). Outperforming image segmentation by

exploiting approximate k-means algorithms. In *International Conference on Optimization and Decision Science*, pages 31–38. Springer.

[8] Amato, F., Marrone, S., Moscato, V., Piantadosi, G., Picariello, A., and Sansone, C. (2017b). Chatbots meet ehealth: Automatizing healthcare. In *WAIAH@ AI* IA*, pages 40–49.

[9] Amato, F., Marrone, S., Moscato, V., Piantadosi, G., Picariello, A., and Sansone, C. (2019). Holmes: ehealth in the big data and deep learning era. *Information*, 10(2):34.

[10] Amoroso, D. and Tamburrini, G. (2017). The ethical and legal case against autonomy in weapons systems. *Global Jurist*, 18(1).

[11] Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias. *ProPublica, May*, 23:2016.

[12] Assael, Y. M., Shillingford, B., Whiteson, S., and de Freitas, N. (2016). Lipnet: End-to-end sentence-level lipreading.

[13] Ateniese, G., Felici, G., Mancini, L. V., Spognardi, A., Villani, A., and Vitali, D. (2013). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *arXiv preprint arXiv:1306.4447*.

[14] Bae, H., Jang, J., Jung, D., Jang, H., Ha, H., and Yoon, S. (2018). Security and privacy issues in deep learning. *arXiv preprint arXiv:1807.11655*.

[15] Barbareschi, M., Papa, C., and Sansone, C. (2017). Approximate decision tree-based multiple classifier systems. In *International Conference on Optimization and Decision Science*, pages 39–47. Springer.

[16] Bardsley, E. and Donaldson, A. F. (2014). Warps and atomics: Beyond barrier synchronization in the verification of gpu kernels. In *NASA Formal Methods Symposium*, pages 230–245. Springer.

[17] Barkadehi, M. H., Nilashi, M., Ibrahim, O., Fardi, A. Z., and Samad, S. (2018). Authentication systems: A literature review and classification. *Telematics and Informatics*, 35(5):1491–1511.

[18] Barreto, L., Amaral, A., and Pereira, T. (2017). Industry 4.0 implications in logistics: an overview. *Procedia Manufacturing*, 13:1245–1252.

[19] Bates, C. L., Chen, J. C.-T., Garbow, Z. A., and Young, G. E. (2014). Advertising in virtual environments based on crowd statistics. US Patent 8,924,250.

[20] Belz, S., Sullivan, M. A., and Pratt, J. (2009). System and method for sending targeted marketing data using proximity data. US Patent App. 11/958,825.

[21] Bhanu, B. and Kumar, A. (2017). *Deep learning for biometrics*. Springer.

[22] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[23] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[24] Brem, R. F., Baum, J., Lechner, M., Kaplan, S., Souders, S., Naul, L. G., and Hoffmeister, J. (2003). Improvement in sensitivity of screening mammography with computer-aided detection: a multiinstitutional trial. *American Journal of Roentgenology*, 181(3):687–693.

[25] Brown, T., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial patch. arxiv e-prints (dec. 2017). *arXiv preprint cs.CV/1712.09665*, 1(2):4.

[26] Burton, E., Goldsmith, J., Koenig, S., Kuipers, B., Mattei, N., and Walsh, T. (2017). Ethical considerations in artificial intelligence courses. *arXiv preprint arXiv:1701.07769*.

[27] Canuto, A. M., Pintro, F., and Xavier-Junior, J. C. (2013). Investigating fusion approaches in multi-biometric cancellable recognition. *Expert Systems with Applications*, 40(6):1971–1980.

[28] Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE.

[29] Carlini, N. and Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE.

[30] Chen, J.-C., Patel, V. M., and Chellappa, R. (2016). Unconstrained face verification using deep cnn features. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE.

[31] Cheng, N., Chandramouli, R., and Subbalakshmi, K. (2011). Author gender identification from text. *Digital Investigation*, 8(1):78–88.

[32] Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., and Shelhamer, E. (2014). cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*.

[33] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

[34] Chollet, F. et al. (2015). Keras. https://keras.io.

[35] Chugh, T., Cao, K., and Jain, A. K. (2018). Fingerprint spoof buster: Use of minutiae-centered patches. *IEEE Transactions on Information Forensics and Security*, 13(9):2190–2202.

[36] Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al. (2013). The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6):1045–1057.

[37] Co, K. T., Muñoz-González, L., de Maupeou, S., and Lupu, E. C. (2019). Procedural noise adversarial examples for black-box attacks on deep convolutional networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 275–289.

[38] Correia-Silva, J. R., Berriel, R. F., Badue, C., de Souza, A. F., and Oliveira-Santos, T. (2018). Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[39] Datta, A., Tschantz, M. C., and Datta, A. (2015). Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112.

[40] Desmarais, S. L., Johnson, K. L., and Singh, J. P. (2016). Performance of recidivism risk assessment instruments in us correctional settings. *Psychological Services*, 13(3):206.

[41] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[42] Dirican, C. (2015). The impacts of robotics, artificial intelligence on business and economics. *Procedia-Social and Behavioral Sciences*, 195:564–573.

[43] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655.

[44] Dong, Y., Liao, F., Pang, T., Su, H., Hu, X., Li, J., and Zhu, J. (2017). Boosting adversarial attacks with momentum. arxiv preprint. *arXiv preprint arXiv:1710.06081*.

[45] Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

[46] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

[47] Dziugaite, G. K., Ghahramani, Z., and Roy, D. M. (2016). A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*.

[48] Eigen, D., Rolfe, J., Fergus, R., and LeCun, Y. (2013). Understanding deep architectures using a recursive convolutional network. *arXiv preprint arXiv:1312.1847*.

[49] Elson, J., Douceur, J. J., Howell, J., and Saul, J. (2007). Asirra: a captcha that exploits interest-aligned manual image categorization.

[50] Evans, N. (2019). *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection*. Springer.

[51] Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

[52] Fenton, J. J., Taplin, S. H., Carney, P. A., Abraham, L., Sickles, E. A., D'Orsi, C., Berns, E. A., Cutter, G., Hendrick, R. E., Barlow, W. E., et al. (2007). Influence of computer-aided detection on performance of screening mammography. *New England Journal of Medicine*, 356(14):1399–1409.

[53] Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333.

[54] Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. (2014). Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32.

[55] Freeman, K. (2016). Algorithmic injustice: How the wisconsin supreme court failed to protect due process rights in state v. loomis. *North Carolina Journal of Law and Technology*, 18:75–180.

[56] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

[57] Ganju, K., Wang, Q., Yang, W., Gunter, C. A., and Borisov, N. (2018). Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–633.

[58] Gensler, H. J. (1985). Ethical consistency principles. *The Philosophical Quarterly (1950-)*, 35(139):156–170.

[59] Ghiani, L., Marcialis, G. L., and Roli, F. (2012). Fingerprint liveness detection by local phase quantization. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 537–540. IEEE.

[60] Ghiani, L., Yambay, D. A., Mura, V., Marcialis, G. L., Roli, F., and Schuckers, S. A. (2017). Review of the fingerprint liveness detection (livdet) competition series: 2009 to 2015. *Image and Vision Computing*, 58:110–128.

[61] Gilbert, F. J., Astley, S. M., McGee, M. A., Gillan, M. G., Boggis, C. R., Griffiths, P. M., and Duffy, S. W. (2006). Single reading with computer-aided detection and double reading of screening mammograms

in the united kingdom national breast screening program. *Radiology*, 241(1):47–53.

[62] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

[63] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning book. *MIT Press*, 521(7553):800.

[64] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

[65] Gragnaniello, D., Poggi, G., Sansone, C., and Verdoliva, L. (2015). Local contrast phase descriptor for fingerprint liveness detection. *Pattern Recognition*, 48(4):1050–1058.

[66] Gravina, M., Marrone, S., Piantadosi, G., Sansone, M., and Sansone, C. (2019). 3tp-cnn: Radiomics and deep learning for lesions classification in dce-mri. In *International Conference on Image Analysis and Processing*, pages 661–671. Springer.

[67] Gromet, M. (2008). Comparison of computer-aided detection to double reading of screening mammograms: review of 231,221 mammograms. *American Journal of Roentgenology*, 190(4):854–859.

[68] Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. (2019). Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

[69] Guiochet, J., Machin, M., and Waeselynck, H. (2017). Safety-critical advanced robots: A survey. *Robotics and Autonomous Systems*, 94:43–52.

[70] Gundersen, O. E., Gil, Y., and Aha, D. W. (2018). On reproducible ai: Towards reproducible research, open science, and digital scholarship in ai publications. *AI Magazine*, 39(3):56–68.

[71] Guynn, J. (2015). Google photos labeled black people ‘gorillas’. *USA Today, July*.

[72] Han, D., Liu, Q., and Fan, W. (2018). A new image classification method using cnn transfer learning and web data augmentation. *Expert Systems with Applications*, 95:43–56.

[73] Han, S., Mao, H., and Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149.*

[74] Hawking, S., Tegmark, M., Russell, S., and Wilczek, F. (2014). Transcending complacency on superintelligent machines. *Huffington Post, April*, 19.

[75] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[76] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[77] Hoffman, S., Sharma, R., and Ross, A. (2018). Convolutional neural networks for iris presentation attack detection: Toward cross-dataset and cross-sensor generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1620–1628.

[78] Horton, H. (2016). Microsoft deletes 'teen girl'ai after it became a hitler-loving sex robot within 24 hours. *The Telegraph*, 24.

[79] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861.*

[80] Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3.

[81] Huang, G.-B. (2003). Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, 14(2):274–281.

[82] Humphrys, M. (2009). How my program passed the turing test. In *Parsing the Turing Test*, pages 237–260. Springer.

[83] Hutson, M. (2018). Artificial intelligence faces reproducibility crisis.

[84] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM.

[85] Joshua, A., Paul, K., and Junbin, G. (2011). Fingerprint matching using a hybrid shape and orientation descriptor. *State of the art in Biometrics, ISBN*, pages 978–953.

[86] Kalmet, P. H., Sanduleanu, S., Primakov, S., Wu, G., Jochems, A., Refaee, T., Ibrahim, A., Hulst, L. v., Lambin, P., and Poeze, M. (2020). Deep learning in fracture detection: a narrative review. *Acta Orthopaedica*, pages 1–6.

[87] King, T. C., Aggarwal, N., Taddeo, M., and Floridi, L. (2019). Artificial intelligence crime: An interdisciplinary analysis of foreseeable threats and solutions. *Science and engineering ethics*, pages 1–32.

[88] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[89] Ko, K. (2007). User's guide to nist biometric image software (nbis). Technical report.

[90] Kolberg, D. and Zühlke, D. (2015). Lean automation enabled by industry 4.0 technologies. *IFAC-PapersOnLine*, 48(3):1870–1875.

[91] Kolosnjaji, B., Zarras, A., Webster, G., and Eckert, C. (2016). Deep learning for classification of malware system call sequences. In *Australasian Joint Conference on Artificial Intelligence*, pages 137–149. Springer.

[92] Koopman, P. and Wagner, M. (2016). Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24.

[93] Kosinski, M., Stillwell, D., and Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805.

[94] Krafft, P., Young, M., Katell, M., Huang, K., and Bugingo, G. (2019). Defining ai in policy versus practice. *arXiv preprint arXiv:1912.11095*.

[95] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[96] Kumar, V., Khattar, D., Gupta, S., Gupta, M., and Varma, V. (2017). User profiling based deep neural network for temporal news recommendation. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 765–772. IEEE.

[97] Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

[98] Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE.

[99] Lee, J., Davari, H., Singh, J., and Pandhare, V. (2018). Industrial artificial intelligence for industry 4.0-based manufacturing systems. *Manufacturing letters*, 18:20–23.

[100] Levman, J., Leung, T., Causer, P., Plewes, D., and Martel, A. L. (2008). Classification of dynamic contrast-enhanced magnetic resonance breast lesions by support vector machines. *IEEE Transactions on Medical Imaging*, 27(5):688–696.

[101] Li, L., Xia, Z., Jiang, X., Roli, F., and Feng, X. (2018). Face presentation attack detection in learned color-liked space. *arXiv preprint arXiv:1810.13170*.

[102] Li, L.-J. and Fei-Fei, L. (2007). What, where and who? classifying events by scene and object recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

[103] Lom, M., Pribyl, O., and Svitek, M. (2016). Industry 4.0 as a part of smart cities. In *2016 Smart Cities Symposium Prague (SCSP)*, pages 1–6. IEEE.

[104] Long, Y., Bindschaedler, V., Wang, L., Bu, D., Wang, X., Tang, H., Gunter, C. A., and Chen, K. (2018). Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*.

[105] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

[106] Maestripieri, D., Henry, A., and Nickels, N. (2017). Explaining financial and prosocial biases in favor of attractive people: Interdisciplinary perspectives from economics, social psychology, and evolutionary psychology. *Behavioral and Brain Sciences*, 40.

[107] Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. (2018). Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196.

[108] Manogaran, G., Thota, C., Lopez, D., and Sundarasekar, R. (2017). Big data security intelligence for healthcare industry 4.0. In *Cybersecurity for Industry 4.0*, pages 103–126. Springer.

[109] Manyika, J. and Bughin, J. (2018). The promise and challenge of the age of artificial intelligence. *McKinsey Global Institute Executive Briefing*.

[110] Marcialis, G. L. and Roli, F. (2009). Liveness detection competition 2009. *Biometric Technology Today*, 17(3):7–9.

[111] Marrone, S., Olivieri, S., Piantadosi, G., and Sansone, C. (2019). Reproducibility of deep cnn for biomedical image processing across frameworks and architectures. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE.

[112] Marrone, S., Piantadosi, G., Fusco, R., Petrillo, A., Sansone, M., and Sansone, C. (2013). Automatic lesion detection in breast dce-mri. In *International Conference on Image Analysis and Processing*, pages 359–368. Springer.

[113] Marrone, S. and Sansone, C. (2019). An adversarial perturbation approach against cnn-based soft biometrics detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[114] Masood, S., Gupta, S., Wajid, A., Gupta, S., and Ahmed, M. (2018). Prediction of human ethnicity from facial images using neural networks. In *Data Engineering and Intelligent Computing*, pages 217–226. Springer.

[115] Memon, S., Manivannan, N., Noor, A., Balachadran, W., and Boulgouris, N. V. (2012). Fingerprint sensors: Liveness detection issue and hardware based solutions. *Sensors & Transducers*, 136(1):35.

[116] Minsky, M. L. (1967). *Computation*. Prentice-Hall Englewood Cliffs.

[117] Miyato, T., Maeda, S.-i., Koyama, M., Nakae, K., and Ishii, S. (2015). Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*.

[118] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

[119] Monrose, F. and Rubin, A. D. (2000). Keystroke dynamics as a biometric for authentication. *Future Generation computer systems*, 16(4):351–359.

[120] Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582.

[121] Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., and Roli, F. (2017). Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38. ACM.

[122] Mura, V., Ghiani, L., Marcialis, G. L., Roli, F., Yambay, D. A., and Schuckers, S. A. (2015). Livdet 2015 fingerprint liveness detection competition 2015. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*, pages 1–6. IEEE.

[123] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

[124] Narang, N. and Bourlai, T. (2016). Gender and ethnicity classification using deep learning in heterogeneous face recognition. In *Biometrics (ICB), 2016 International Conference on*, pages 1–8. IEEE.

[125] Nasr, M., Shokri, R., and Houmansadr, A. (2018). Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 634–646.

[126] Nogueira, R. F., de Alencar Lotufo, R., and Machado, R. C. (2016). Fingerprint liveness detection using convolutional neural networks. *IEEE transactions on information forensics and security*, 11(6):1206–1213.

[127] Nvidia, C. (2011). Nvidia cuda c programming guide. *Nvidia Corporation*, 120(18):8.

[128] Oliveira, F., Santos, A., Aguiar, B., and Sousa, J. (2014). Gamefoundry: social gaming platform for digital marketing, user profiling and collective behavior. *Procedia-Social and Behavioral Sciences*, 148:58–66.

[129] Olsen, O. and Gøtzsche, P. C. (2001). Cochrane review on screening for breast cancer with mammography. *The Lancet*, 358(9290):1340–1342.

[130] Orekondy, T., Schiele, B., and Fritz, M. (2019). Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4954–4963.

[131] Orrù, G., Casula, R., Tuveri, P., Bazzoni, C., Dessalvi, G., Micheletto, M., Ghiani, L., and Marcialis, G. L. (2019). Livdet in action-fingerprint liveness detection competition 2019. *arXiv preprint arXiv:1905.00639*.

[132] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[133] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016). The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE.

[134] Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In *BMVC*, volume 1, page 6.

[135] Penatti, O. A., Nogueira, K., and dos Santos, J. A. (2015). Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on*, pages 44–51. IEEE.

[136] Piantadosi, G. (2017). Breast cancer analysis in dce-mri.

[137] Piantadosi, G., Marrone, S., Fusco, R., Sansone, M., and Sansone, C. (2018a). Comprehensive computer-aided diagnosis for breast t1-weighted dce-mri through quantitative dynamical features and spatio-temporal local binary patterns. *IET Computer Vision*, 12(7):1007–1017.

[138] Piantadosi, G., Marrone, S., and Sansone, C. (2018b). On reproducibility of deep convolutional neural networks approaches. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 104–109. Springer.

[139] Piantadosi, G., Sansone, M., and Sansone, C. (2018c). Breast segmentation in mri via u-net deep convolutional neural networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3917–3922. IEEE.

[140] Polletta, F. and Callahan, J. (2019). Deep stories, nostalgia narratives, and fake news: Storytelling in the trump era. In *Politics of meaning/meaning of politics*, pages 55–73. Springer.

[141] Pyrgelis, A., Troncoso, C., and De Cristofaro, E. (2017). Knock knock, who's there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*.

[142] Raghavendra, R., Venkatesh, S., Raja, K. B., and Busch, C. (2017). Transferable deep convolutional neural network features for fingervein presentation attack detection. In *Biometrics and Forensics (IWBF), 2017 5th International Workshop on*, pages 1–5. IEEE.

[143] Rahman, M. A., Rahman, T., Laganière, R., Mohammed, N., and Wang, Y. (2018). Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy*, 11(1):61–79.

[144] Ras, G., van Gerven, M., and Haselager, P. (2018). Explanation methods in deep learning: Users, values, concerns and challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 19–36. Springer.

[145] Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE.

[146] Reid, D. A., Nixon, M. S., and Stevenage, S. V. (2014). Soft biometrics; human identification using comparative descriptions. *IEEE Transactions on pattern analysis and machine intelligence*, 36(6):1216–1228.

[147] Riley, P. (2019). Three pitfalls to avoid in machine learning.

[148] Rojko, A. (2017). Industry 4.0 concept: background and overview. *International Journal of Interactive Mobile Technologies (iJIM)*, 11(5):77–90.

[149] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[150] Rothe, R., Timofte, R., and Van Gool, L. (2015). Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–15.

[151] Rudin, C., Waltz, D., Anderson, R. N., Boulanger, A., Salleb-Aouissi, A., Chow, M., Dutta, H., Gross, P. N., Huang, B., Ierome, S., et al. (2011). Machine learning for the new york city power grid. *IEEE transactions on pattern analysis and machine intelligence*, 34(2):328–345.

[152] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[153] Rutkin, A. (2016). Digital discrimination.

[154] Sabour, S., Cao, Y., Faghri, F., and Fleet, D. J. (2015). Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122*.

[155] Sadilek, A., Kautz, H., and Silenzio, V. (2012). Modeling spread of disease from social interactions. In *Sixth International AAAI Conference on Weblogs and Social Media*.

[156] Safie, S. I., Soraghan, J. J., and Petropoulakis, L. (2011). Electrocardiogram (ecg) biometric authentication using pulse active ratio (par). *IEEE Transactions on Information Forensics and Security*, 6(4):1315–1322.

[157] Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., and Backes, M. (2018). Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*.

[158] Sample, I. and Hern, A. (2014). Scientists dispute whether computer 'eugene goostman'passed turing test. *The Guardian*, 9.

[159] Sato, M., Suzuki, J., Shindo, H., and Matsumoto, Y. (2018). Interpretable adversarial perturbation in input embedding space for text. *arXiv preprint arXiv:1805.02917*.

[160] Schlingensiepen, J., Nemtanu, F., Mehmood, R., and McCluskey, L. (2016). Autonomic transport management systems—enabler for smart cities, personalized medicine, participation and industry grid/industry 4.0. In *Intelligent transportation systems–problems and perspectives*, pages 3–35. Springer.

[161] Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM.

[162] Shokri, R. and Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM.

[163] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

[164] Siegel, R. L., Miller, K. D., and Jemal, A. (2018). Cancer statistics, 2018. *CA: a cancer journal for clinicians*, 68(1):7–30.

[165] Simmhan, Y., Aman, S., Kumbhare, A., Liu, R., Stevens, S., Zhou, Q., and Prasanna, V. (2013). Cloud-based software platform for big data analytics in smart grids. *Computing in Science & Engineering*, 15(4):38.

[166] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[167] Sinwar, D., Sharma, M. K., and Verma, H. (2020). Remote sensing classification under deep learning: A review. In *Smart Systems and IoT: Innovations in Computing*, pages 813–823. Springer.

[168] Sirignano, J., Sadhwani, A., and Giesecke, K. (2016). Deep learning for mortgage risk. *arXiv preprint arXiv:1607.02470*.

[169] Sitawarin, C., Bhagoji, A. N., Mosenia, A., Chiang, M., and Mittal, P. (2018). Darts: Deceiving autonomous cars with toxic signs. *arXiv preprint arXiv:1802.06430*.

[170] Song, C., Ristenpart, T., and Shmatikov, V. (2017). Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601.

[171] Srinivas, N., Atwal, H., Rose, D. C., Mahalingam, G., Ricanek, K., and Bolme, D. S. (2017). Age, gender, and fine-grained ethnicity prediction using convolutional neural networks for the east asian face dataset. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 953–960. IEEE.

[172] Steinhardt, J., Koh, P. W. W., and Liang, P. S. (2017). Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529.

[173] Stock, T. and Seliger, G. (2016). Opportunities of sustainable manufacturing in industry 4.0. *Procedia Cirp*, 40:536–541.

[174] Su, J., Vargas, D. V., and Kouichi, S. (2017). One pixel attack for fooling deep neural networks. *arXiv preprint arXiv:1710.08864*.

[175] Sun, Y., Chen, Y., Wang, X., and Tang, X. (2014). Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996.

[176] Sundararajan, K. and Woodard, D. L. (2018). Deep learning for biometrics: a survey. *ACM Computing Surveys (CSUR)*, 51(3):65.

[177] Sweeney, L. (2013). Discrimination in online ad delivery. *Queue*, 11(3):10.

[178] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[179] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al. (2015). Going deeper with convolutions. Cvpr.

[180] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

[181] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

[182] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708.

[183] Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., and Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312.

[184] Takeichi, N., Kaida, R., Shimomura, A., and Yamauchi, T. (2017). Prediction of delay due to air traffic control by machine learning. In *AIAA Modeling and Simulation Technologies Conference*, page 1323.

[185] Thuemmler, C. and Bai, C. (2017). Health 4.0: Application of industry 4.0 design principles in future asthma management. In *Health 4.0: How virtualization and big data are revolutionizing healthcare*, pages 23–37. Springer.

[186] Tome, P., Fierrez, J., Vera-Rodriguez, R., and Nixon, M. S. (2014). Soft biometrics and their application in person recognition at a distance. *IEEE Transactions on Information Forensics and Security*, 9(3):464–475.

[187] Triepels, R., Daniels, H., and Feelders, A. (2018). Data-driven fraud detection in international shipping. *Expert Systems with Applications*, 99:193–202.

[188] Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer.

[189] Twellmann, T., Saalbach, A., Muller, C., Nattkemper, T. W., and Wismuller, A. (2004). Detection of suspicious lesions in dynamic contrast enhanced mri data. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 454–457. IEEE.

[190] Valentino-Devries, J., Singer-Vine, J., and Soltani, A. (2012). Websites vary prices, deals based on users' information. *Wall Street Journal*, 24.

[191] Van de Velde, E. F. (1994). *Concurrent scientific computing*, volume 16. Springer Science & Business Media.

[192] van Miltenburg, E. (2016). Stereotyping and bias in the flickr30k dataset. *arXiv preprint arXiv:1605.06083*.

[193] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.

[194] Voigt, P. and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*.

[195] Warhurst, C., Van den Broek, D., Hall, R., and Nickson, D. (2009). Lookism: The new frontier of employment discrimination? *Journal of Industrial Relations*, 51(1):131–136.

[196] Watkins, D., Gallardo, G., and Chau, S. (2018). Pilot support system: A machine learning approach. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pages 325–328. IEEE.

[197] Winfield, A. F. and Jirotka, M. (2018). Ethical governance is essential to building trust in robotics and artificial intelligence systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2133):20180085.

[198] Wolf, M. J., Miller, K., and Grodzinsky, F. S. (2017). Why we should have seen that coming: comments on microsoft's tay experiment, and wider implications. *ACM SIGCAS Computers and Society*, 47(3):54–64.

[199] Xie, Q., Hovy, E., Luong, M.-T., and Le, Q. V. (2019). Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*.

[200] Xu, W., Evans, D., and Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

[201] Yang, S., Han, K., Zheng, Z., Tang, S., and Wu, F. (2018). Towards personalized task matching in mobile crowdsensing via fine-grained user profiling. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 2411–2419. IEEE.

[202] Yao, G., Lei, T., and Zhong, J. (2019a). A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118:14–22.

[203] Yao, Y., Li, H., Zheng, H., and Zhao, B. Y. (2019b). Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2041–2055.

[204] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

[205] Zewail, R., Elsafi, A., Saeb, M., and Hamdy, N. (2004). Soft and hard biometrics fusion for improved identity verification. In *Circuits and Systems, 2004. MWSCAS'04. The 2004 47th Midwest Symposium on*, volume 1, pages I–225. IEEE.

[206] Zhang, D. D. (2013). *Automated biometrics: Technologies and systems*, volume 7. Springer Science & Business Media.

[207] Zhang, Z., Song, Y., and Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5810–5818.

[208] Zhao, H.-x. and Magoulès, F. (2012). A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, 16(6):3586–3592.

[209] Zoorob, R., Anderson, R., Cefalu, C., and Sidani, M. (2001). Cancer screening guidelines. *American family physician*, 63(6):1101–1112.

List of figures

List of tables