UNIVERSITÀ DEGLI STUDI DI SIENA

Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

UNIVERSITÀ
DI SIENA
1240

Improving fake image detection through background analysis, facial segmentation, and model interpretability

Marco Tanfoni

PhD in Information Engineering and Science

Supervisors

Prof. Marco Maggini
Prof. Monica Bianchini

Examination Commitee

Prof. Federico Becattini
Prof. Carlo Colombo
Dr. Antonino Crivello
Prof. Fabrizio Silvestri

Thesis reviewers

Prof. Stefano Cagnoni, University of Parma
Prof. Carlo Colombo, University of Florence

Siena, 14th April 2025

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank the many people who were part of this journey—some from the very beginning or even before its start, others who joined along the way and helped me find the strength to carry it through when things got tough.

To Ambra, who shared with me not only this PhD but also the many ups and downs of everyday life. Her support, even in the hardest moments, helped me move forward when I was ready to stop (and still does), and I feel I owe her more than these few lines can say.

The same can be said for my mother and Silvia, whose presence, support, and love—whether expressed in a message, a phone call, or a meal together—have always been essential and have kept me going when I needed it the most. Also thanks to Mirko and their two wonderful daughters Nadia and Alba, whose cheerful energy have brightened many moments along the way.

I also feel the need to thank my father, who, although he won't be able to read these lines, continues to, and will always be, a part of who I am, through the example he set for me and that I will always try to follow.

To Filippo, my best friend and the brother I got to choose. For his unwavering presence, the experiences we have shared, and the way he always manages to make me feel understood, no matter what.

To Elia, who started as a colleague in the lab and became a true friend. Like many others, but more than most, he gave me a practical reason to stay when I was close to walking away from this PhD, thanks to the idea that eventually became the main topic of this thesis. From that point on, our collaboration was not only productive but also fun, and I can confidently say, without exaggeration, that I wouldn't be here without him.

To my crew: together with Filippo, Frel, friend since forever and companion in more adventures than I can count. There's almost nothing he says that doesn't make me laugh, an invaluable gift when things get heavy; and Stefano, for opening the doors of his legendary cuckhouse to me and the rest of the crew; thank you for the hospitality, the chaos, and the laughs.

To Lo, with whom I shared many years of fencing and many more moments beyond that, and who I don't see as often as I'd like, but who always finds the right words when I need advice or simply lends an ear during one of our fast–food catchups.

To Edoardo, for the chill evening games and the company during those quiet mornings that would've otherwise been spent alone, especially while writing this thesis.

To Tia, whose creativity has always been a source of inspiration and still manages to surprise me every single day.

To the *PC master reis* group: Frel (again), Caso, Draw, and Jack, for the endless games, the laughs, and the humiliating defeats. From Make Way to You Suck at Parking, Tokyo, Gang Beasts, and all the other dumb games we've played together: thanks for always keeping things fun (even when you're destroying me).

To all the lab buds: Nicco, Barbara, Paolo, Caterina, Filippo, Veronica, Simone, Pietro, Sara, Giacomo, and Fiamma for their help, discussions, and insights, but above all for creating an environment that felt human, supportive, and never cold. Working with them made even the more frustrating days easier to face, not to mention the pleasant experience in Seville —and the just–a–bit–less–pleasant one in Barcelona—we shared together with some of them.

To my two supervisors, Marco, for his support throughout the journey, and Monica, whose continuous guidance, insight, and encouragement proved invaluable at many key moments, from the months before the beginning of this journey to its very end. Their ideas were the seeds from which this thesis grew.

Lastly, I would like to thank Leonardo SpA for funding this PhD and for never interfering with the direction of my research, allowing me to carry it out with full intellectual freedom.

Ringraziamenti

Vorrei ringraziare le molte persone che hanno fatto parte di questo percorso—alcune fin dall'inizio o addirittura da prima che cominciasse, altre che si sono aggiunte lungo la strada e mi hanno aiutato a trovare la forza di portarlo a termine quando le cose si facevano difficili.

Ad Ambra, che ha condiviso con me non solo questo dottorato ma anche i tanti alti e bassi della vita quotidiana. Il suo supporto, anche nei momenti più duri, mi ha aiutato ad andare avanti quando ero pronto a fermarmi (e lo fa ancora), e sento di doverle più di quanto queste poche righe possano esprimere.

Lo stesso vale per mamma e Silvia, la cui presenza, il supporto e l'amore—che si manifestassero in un messaggio, una telefonata o un pasto insieme—sono sempre stati fondamentali e mi hanno sostenuto quando ne avevo più bisogno. Un grazie anche a Mirko e alle loro due meravigliose figlie Nadia e Alba, la cui allegria ha illuminato molti momenti lungo il cammino.

Sento anche il bisogno di ringraziare babbo, che, sebbene non potrà leggere queste righe, continua e continuerà a far parte di ciò che sono, attraverso l'esempio che mi ha lasciato e che cercherò sempre di seguire.

A Filippo, il mio migliore amico e il fratello mi sono scelto. Per la sua presenza costante, le esperienze condivise, e il modo in cui riesce sempre a farmi sentire compreso, qualunque cosa accada.

A Elia, che è partito come collega di laboratorio ed è diventato un vero amico. Come molti altri, ma più della maggior parte, mi ha dato un motivo concreto per restare quando ero vicino ad abbandonare tutto, grazie all'idea che è poi diventata l'argomento principale di questa tesi. Da quel momento in poi, la nostra collaborazione non è stata solo produttiva ma anche divertente, e posso dire con sicurezza, senza esagerare, che senza di lui non sarei qui.

Alla mia cricca: oltre a Filippo, Frel, amico da sempre e compagno in più avventure di quante riesca a ricordare. Non c'è quasi nulla che dica che non mi faccia ridere, un dono inestimabile quando le cose si fanno pesanti; e Stefano, per aver aperto le porte della sua leggendaria cuckhouse a me e al resto della banda; grazie per l'ospitalità, il caos e le risate.

A Lo, con cui ho condiviso molti anni di scherma e molti più momenti al di fuori di essa, e che non vedo quanto vorrei, ma che trova sempre le parole giuste quando ho bisogno di un consiglio o semplicemente mi presta ascolto durante uno dei nostri ritrovi fast-food.

A Edoardo, per le tranquille serate di gioco e la compagnia durante quelle mattine silenziose che altrimenti avrei passato da solo, specialmente mentre scrivevo questa tesi.

A Tia, la cui creatività è sempre stata una fonte di ispirazione e continua a sorprendermi ogni singolo giorno.

Al gruppo dei *PC master reis*: Frel (di nuovo), Caso, Draw e Jack, per le infinite partite, le risate e le sconfitte umilianti. Da Make Way a You Suck at Parking, Tokyo, Gang Beasts, e tutti gli altri giochi scemi a cui abbiamo giocato insieme: grazie per aver sempre reso tutto divertente (anche quando mi stavate distruggendo).

A tutti i compagni di laboratorio: Nicco, Barbara, Paolo, Caterina, Filippo, Veronica, Simone, Pietro, Sara, Giacomo e Fiamma per l'aiuto, le discussioni e gli spunti, ma soprattutto per aver creato un ambiente umano, accogliente e mai freddo. Lavorare con loro ha reso più affrontabili anche le giornate più frustranti, senza contare la piacevole esperienza a Siviglia—e quella solo–un–po'–meno–piacevole a Barcellona—che ho condiviso con alcuni di loro.

Ai miei due tutor, Marco, per il suo supporto lungo tutto il percorso, e Monica, la cui guida costante, intuizioni e incoraggiamento si sono rivelati preziosi in molti momenti chiave, dai mesi precedenti l'inizio di questo percorso fino alla sua conclusione. Le loro idee sono stati i semi da cui è nata questa tesi.

Infine, desidero ringraziare Leonardo SpA per aver finanziato questo dottorato e per non aver mai interferito con la direzione della mia ricerca, permettendomi di portarla avanti con piena libertà intellettuale.

Part I

Foundations and context

Chapter 1

Introduction

The advent of Artificial Intelligence (AI) has been crucially impactful in various domains, both scientific and industrial, and computer vision stood out as one of its fundamental areas, since it allowed machines to interpret and process visual data, opening applications in a vast range of fields, from autonomous driving and healthcare to content moderation and security. This field has evolved rapidly, shifting from manually designed algorithms to the widespread adoption of Deep Learning (DL) techniques, which have revolutionized how visual information is processed and understood. Current computer vision algorithms rely on DL, and particularly on Convolutional Neural Networks (CNNs), which quickly became their backbone in most of the cases. On one side, this shift allowed DL tools to reach unprecedented accuracy in all the already tackled tasks, such as edge detection or feature matching; on the other side, it also unlocked new possibilities in more complex tasks, such as object detection, image segmentation, or video analysis.

With these technologies becoming more and more integrated into delicate applications such as healthcare, personalized advertising, and social media content curation, ensuring their ethical usage has become a crucial concern. More specifically, with the advent of the newest, powerful, generative models, such as Generative Adversarial Networks (GANs), which are capable of generating highly realistic synthetic images, there is a growing need of innovative approaches to effectively detect such content.

This thesis addresses the problem of detecting images depicting human faces that have been synthetically generated via such models, with particular attention to understanding how specific regions of the image influence the decision. This aspect is examined through a twofold investigation: by analysing if and how much the image background plays a role in determining whether it has been generated or not, also implementing a well–known explainability technique to visually identify the most important areas of the image for the decision, and by verifying if the information needed to segment a human face into different semantic areas is useful for improving detection performance.

1.1 Background and motivation

The field of computer vision is one of the most relevant topics in the AI domain and has rapidly evolved in recent years, largely replacing traditional techniques, which relied on manually designed algorithms to interpret visual information. Indeed, techniques such as feature matching [1], edge detection [2], contour–based segmentation [3], and so on, had to be adapted to each different application, thus limiting their applicability to multiple tasks. Instead, modern computer vision algorithms introduced the use of DL, which, as of today, has become central

to these methods, allowing them to capture visual information with a level of accuracy and flexibility that was previously unattainable.

One of the fundamental technologies that allows researchers to analyse visual features is that of Convolutional Neural Networks (CNNs), whose conceptual foundation was first introduced by Fukushima with the Neocognitron [4] and later advanced by LeCun [5], Krizhevsky [6], and He [7]. The models in this family are capable of automatically learn features from data, parsing them in a progressive fashion and learning to locate their most informative fractions. In the computer vision field, this means that such models progressively develop the ability to identify patterns and structures within images, ranging from simple edges and textures in the initial layers [8], to complex shapes and objects in deeper layers [9]. CNNs have proven to be highly versatile since their introduction, providing state–of–the–art performance in analysing any kind of information in the form of images, but also in different tasks, such as video processing [10], natural language processing —through visual embeddings [11]— time–series classification, via an appropriate data conversion [12], and complex domains, including 3D data analysis [13] and multimodal learning [14].

Another fundamental technology covered in this work is the so–called Generative Adversarial Network (GAN) [15]. The models of this class are able to generate virtually any kind of synthetic data by emulating real distributions provided during the training phase. In the context of computer vision, GANs are used to produce highly realistic images of various subjects, including environments, animals, and people. Regarding the latter, this ability to create human faces that are almost indistinguishable from their real counterparts has raised critical concerns regarding authenticity and reliability, especially in sensitive applications [16]. For example, GAN–generated images pose significant security risks in fields such as digital security, where synthetic content can be exploited to create deep–fakes, potentially leading to identity fraud, impersonation, or the spread of misinformation [17]. Another potential risk lies in the field of social media, where the dissemination of fake visual content can undermine trust in digital platforms and fuel misinformation campaigns [18]. Finally, GANs could pose a threat to biometric authentication systems by generating synthetic identities capable of bypassing security measures, thereby compromising system integrity [19].

Existing works already achieve close to perfection detection performance, but they often overlook the decision–making process of their model. In the author's opinion, understanding which parts of an image contribute most to the classification is a crucial task, particularly if one considers the fact that the background is largely recognized as a weak point regarding face–generation models, often being blurry or containing visual artefacts. Humans themselves, when trying to detect generated images, usually rely on background inconsistencies that stand out at a first glance, thus making the background a key indicator for an image being synthetic. To the author's knowledge, there has been very scarce research towards this direction. On the other hand, the usage of particular parts of the face to enhance detection rates has already been introduced in the literature, for example focusing on the eyes, like in [20], where a Siamese Neural Network exploit certain inter–eye symmetries and inconsistencies to detect generated human faces, achieving performance that are comparable or even superior to other state–of–the–art methods that analyse the entire face. Other studies have instead employed a texture–based method for detection purposes, like [21], where some particular pixel patterns that occur frequently in GAN–generated images are leveraged. Also from this point of view, there is

still the possibility to investigate whether the ability to recognize some areas of the face, such as the nose, the eyes, the mouth and so on, is useful when trying to separate AI–generated faces from real ones. Again, this is inspired by the behaviour of humans, who may focus on specific characteristics of the image, observing each part of the face individually and looking for anomalies in each of them, when approaching the task of distinguishing synthetic from legitimate images.

These challenges provide the motivation for this thesis, which aims to contribute developing a more robust, transparent and interpretable method of detecting synthetic human faces.

1.2 Research scope and objectives

This research focuses on the detection of synthetic human faces generated by GANs, with particular attention, on one side, on the trustworthiness of the detection method, which is investigated by understanding the role of different image regions and their impact on the proposed detection model, and, on the other side, on the exploration of the effectiveness of the transfer learning paradigm in this context. This work does not aim to surpass the detection performance of existing state–of–the–art methods in terms of mere metrics; instead, it highlights a question that the author considered crucial from the beginning: how much the detection process is really based on facial features rather than on the background elements of the image to understand whether further research in this direction is justified, with the broader goal of advancing explainability and interpretability in synthetic face detection systems

The main objectives of this thesis are as follows.

1. **To investigate the role of the background in synthetic face detection:** understand how background inconsistencies contribute to the detection of GAN–generated faces and explore whether removing the background affects model performance.

2. **To assess the potential of leveraging transfer learning:** explore how knowledge gained during a facial segmentation task can be leveraged to improve detection performance.

3. **To analyse the interpretability of the proposed method:** introduce a well–known explainability technique in the pipeline, to determine the most influential regions of the image and provide insights into whether detection decisions are driven by facial features or background elements.

1.3 Research hypotheses

The main hypotheses that this thesis aim to prove are the following:

H1 **The background significantly helps the detection mechanism.** Since the background in GAN–generated images often contains artifacts and other kind of inconsistencies, a noticeable drop in classification performance is expected when the facial area of the image is isolated from the rest.

H2 **Transfer learning from facial segmentation tasks improves detection performance.** The knowledge acquired from a prior training on a semantic segmentation task, that is, learning to classify facial regions like eyes, nose, and mouth, is useful to improve the model's performance.

H3 **Explainability techniques can reveal the most influential regions for classification.** Leveraging this kind of tools provides insights into the model's decision–making process, highlighting image areas that are most relevant for detecting synthetic faces.

1.4 Methodology overview

In order to prove the hypotheses presented in Section 1.3, this work involved various steps, that will be described in detail in the following sections. What follows is a brief overview of said steps.

Data gathering: The data for this research originated from two primary sources: for the segmentation task, the paid version of a dataset called Mut1ny was used. The images in this dataset, approximately 70,000, are labelled at the pixel–level into multiple classes, making it particularly suitable for the task. On the other hand, for the classification task, a combination of real and synthetic facial images, drawn from a collection called ArtiFact and publicly available online, was employed, reaching a total of more than 300,000 files. For a more precise description of the data, refer to Section 3.1.

Segmentation model training: The semantic segmentation model here presented is based on DeepLabV3+ [22] and trained on the Mut1ny dataset. This part of the procedure is necessary both to isolate the facial regions from the background and to obtain the information needed for the transfer learning method. An in–depth description of this phase is included in Section 3.3.

Background removal and fake detection: After identifying the faces in each image, an alternative version of the classification dataset was created with the background removed. A slightly modified version of the model used for the segmentation phase performed the fake detection separately on both of the versions of the dataset, and the performance are compared. This procedure is discussed in Sections 3.3 and 3.4.

Transfer learning: The weights obtained from the segmentation task were reused in the binary classification task to detect synthetic faces. This approach, explored in Section 3.5, aims to demonstrate how transfer learning improves the model's ability to differentiate between real and generated images.

Explainability analysis: An explainability technique, SHapley Additive exPlanations (SHAP) [23], was employed to highlight the most influential areas of the images for the classification. In particular, this method shows how the model focuses on the background when it is available, and how it is forced to look at the actual facial features when it is removed. This procedure can be found in Section 3.6.

1.5 Thesis outline

This thesis is structured into four main parts:

Part I: Foundations and context: This part, specifically Chapter 2, covers the fundamental knowledge in the AI field needed to understand the topics treated in this work. In particular, basic principles of CNNs and GANs are introduced, together with those of transfer learning and the mathematical principles behind the adopted explainability technique, SHAP.

Part II: Methodology and implementation: Chapter 3 details the proposed approach, both in its theoretical aspects and in its practical implementation, describing how segmentation, fake detection, transfer learning and explainability were performed, also including the description of the evaluation metrics used to assess the performance of the approach, as well as some implementation details such as model training setup and hardware configuration.

Part III: Experimental results and analysis: This part presents the experimental setup and discusses the results of the three main research direction covered in this work: facial segmentation, fake detection and explainability, respectively in sections 4.1, 4.2, and 4.3.

Part IV: Conclusions and future work: The final part summarizes the findings of the thesis, highlights its contributions and limitations (Chapter 5) and briefly discusses two additional works that were published during the doctoral program but are not part of this research (Chapter 6).

1.6 Publications

All the candidate's publications are listed below, both relating to the main research line described in this thesis and to various studies carried out before and during the doctorate.

1. Tanfoni, M., Ceroni, E. G., Marziali, S., Pancino, N., Maggini, M., and Bian-chini, M. (2024). **Generated or Not Generated (GNG): The Importance of Background in the Detection of Fake Images**. *Electronics*, 13(16), 3161.
https://doi.org/10.3390/electronics13163161

2. Tanfoni, M., Ceroni, E. G., Pancino, N., Bianchini, M., Maggini, M. (2024). **Facial Segmentation in Deepfake Classification: a Transfer Learning Approach**. *Procedia Computer Science*, 246, pp. 4160–4168.
https://doi.org/10.1016/j.procs.2024.09.255

3. Tanfoni, M., Ceroni, E. G., Maggini, M., Pancino, N., Bianchini, M. **A Hybrid Deep Learning Approach for Liver Tumor Segmentation Using DeepLabV3+ and Hidden Markov Models**. *Proc. of the IEEE International Symposium on Systems Engineering, ISSE 2024*, Perugia, Italy, pp. 1–5.
https://doi: 10.1109/ISSE63315.2024.10741139

4. Landi, E., *et al.* **A MobileNet Neural Network Model for Fault Diagnosis in Roller Bearings**. *Proc. of the 2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Kuala Lumpur, Malaysia, pp. 1–6.
https://doi: 10.1109/I2MTC53148.2023.10176049

5. Pathak, G. A., *et al.* (2022). **A first update on mapping the human genetic architecture of COVID-19**. *Nature*, 608, pp. E1–E10.
https://doi.org/10.1038/s41586-022-04826-7

6. Kousathanas, A., *et al.* (2022). **Whole-genome sequencing reveals host factors underlying critical COVID-19**. *Nature*, 607(7917), pp. 97–103.
https://10.1038/s41586-022-04576-6

7. Fallerini, C., *et al.* (2022). **Common, low-frequency, rare, and ultra-rare coding variants contribute to COVID-19 severity** (2022). *Human Genetics*, 141(1), pp. 147–173.
https://doi: 10.1007/s00439-021-02397-7

8. Niemi, M. E. K., *et al.* (2021). **Mapping the human genetic architecture of COVID-19**. *Nature*, 600(7889), pp. 472–477.
https://10.1038/s41586-021-03767-x

9. D'Antonio, M., *et al.* (2021). **SARS-CoV-2 susceptibility and COVID-19 disease severity are associated with genetic variants affecting gene expression in a variety of tissues**. *Cell Reports*, 37(7), 110020.
https://10.1016/j.celrep.2021.110020

Chapter 2

Theoretical foundations

This chapter introduces the neural network models used in the thesis, with particular emphasis on convolutional networks and generative adversarial networks. Furthermore, the transfer learning paradigm is presented and the importance of explainability and interpretability in security–sensitive problems is explained. The chapter concludes with a presentation of the most recent literature in fake detection.

2.1 Convolutional neural networks (CNNs)

CNNs were first introduced by LeCun et al. in [5], evolving from Fukushima's Neocognitron [4], and were originally designed for tasks involving spatially structured data, specifically for image processing. With respect to the previous fully connected neural networks, CNNs introduce the concept of convolutional layers (from which their name) with local connections, allowing this kind of models to capture spatial features significantly reducing the computational cost. Since their original release, CNNs' field of application has evolved from just image analysis to a much wider scope, thanks to their ability to learn hierarchical representations also useful for different types of data.

2.1.1 Fundamental principles of CNNs

To process input data, a sequence of specialized layers are leveraged in CNNs. First, a **convolutional layer** applies a set of filters (kernels) to the input and computes a weighted sum at every position, producing a feature map. Mathematically, this operation can be expressed as:

$$(f * x)(i,j) = \sum_m \sum_n x(i-m, j-n) f(m,n), \tag{2.1}$$

where x is the input data, f is the filter and (i,j) are spatial coordinates (e.g. the coordinates of the pixel with respect to the dimensions of the image). In this case, the key aspect that the network must learn are the filters, which are typically smaller than the input data — common dimensions are 3×3 or 5×5 —, and are applied to the input by sliding over it. Each filter specializes on recognizing certain patterns, and its parameters are learned during training. For example, if the input data is a standard three–channel image, filters are set to operate across all the channels simultaneously, and each of them learns to detect specific spatial patterns, such as edges, textures, or repetitive structures. All the information gathered by the filters is combined in a single feature map, which represents the spatial activation of a specific pattern detected by each of the filters.

A comprehensive representation of the computational flow of a CNN is shown in Figure 2.1.

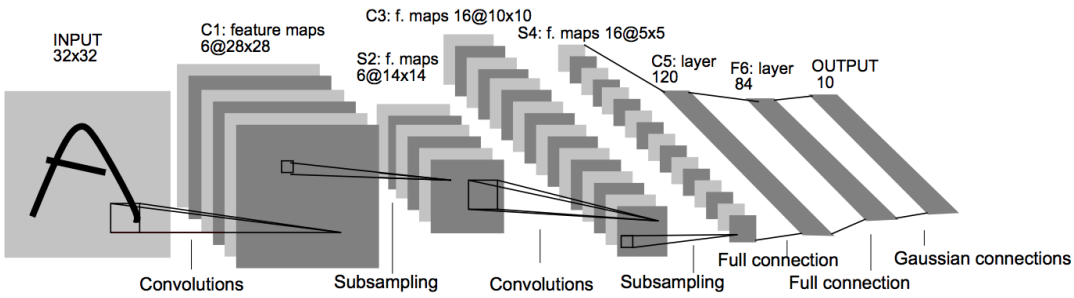

Figure 2.1: Architecture of a CNN, as presented in [5].

The convolutional layer's behaviour is influenced by multiple design choices, such as the **stride**, representing the size of the step with which the filter moves across the input, the chosen **padding**, used to preserve the spatial dimensions of the data and ensure that the filter can process the borders of the input, and the **number of filters**, each producing a distinct feature map. The feature maps are then combined into the final multi–channel feature representation, with each channel representing a unique and different aspect of the input data.

After the convolutional layer, a nonlinear layer, also known as **activation function**, is introduced into the network, to model spatial relationships in the feature map obtained from the previous step. The activation function first proposed in [5] is the sigmoid function, defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

which was widely used at the time to introduce nonlinearities in the models, and ensures that the output values range between 0 and 1, especially useful when interpreting the results as probabilities (although the values of the activation outputs do not necessarily sum to 1). However, sigmoid functions suffer from some drawbacks, such as the vanishing gradient problem, where, for inputs being significantly large or small, the gradient tends to 0 during backpropagation, thus slowing down the learning process for deep networks. Additionally, the computation of the exponential function increases the computational cost, making it less efficient for large–scale networks.

One of the most commonly used activation functions nowadays is the Rectified Linear Unit (ReLU), first introduced in [24], but named as such after [25], defined as:

$$\mathrm{ReLU}(x) = \max(0, x),$$

which is computationally efficient and helps reducing the vanishing gradient problem. ReLU only activates a neuron if its input is positive, introducing sparsity into the network, which can improve computational efficiency and generalization. Moreover, its piecewise linear nature makes it faster to compute compared to the sigmoid, which involves more complex mathematical operations. One limitation of ReLU is the the so–called "dying ReLU" problem, where neurons can become inactive if their inputs are always negative, making them less efficient. To overcome this issue, several alternative activation functions have been introduced, such as Swish [26]

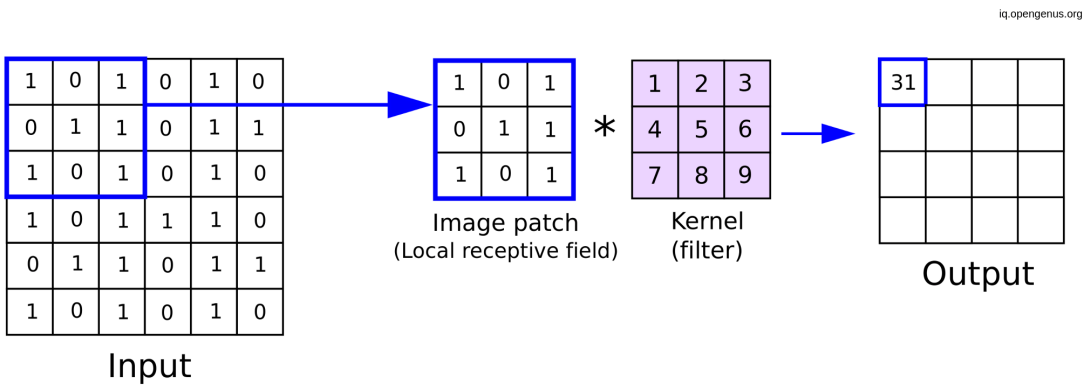

Figure 2.2: Visual representation of the convolution operation on an input matrix. Source.

and Leaky ReLU [27], variants of, respectively, sigmoid and ReLU, and proven to improve the performance for certain architectures and tasks.

Next, CNNs leverage **pooling layers** to reduce the spatial dimensions of feature maps and decrease computational complexity, mitigating overfitting and also improving the network's robustness by ensuring its invariance to small translations in the input data. The choice of the pooling strategy is determined a priori during the network design and typically involves one of two methods: max pooling, which selects the maximum value within a defined window to preserve the most prominent features, or average pooling, which computes the mean value within the window to provide a smoother representation.

Finally, a certain number of **fully connected layers** take the high–level features extracted by convolutional layers and map them to the desired output, such as class probabilities in an image classification task. These layers perform a weighted sum of the input features followed by an activation function, effectively acting as a traditional neural network classifier.

2.1.2 CNN applications in image analysis

CNNs have been fundamental for many image analysis tasks across various domains. Some examples are listed in the following.

- **Object detection and localization:** frameworks like YOLO [28] and Faster R–CNN [29] use CNNs to identify and locate objects in images with high accuracy and efficiency.
- **Image segmentation:** models such as U–Net [30] and DeepLab [22] leverage CNNs to partition images into meaningful regions, facilitating applications in medical imaging and autonomous driving.
- **Facial recognition:** architectures like FaceNet [31] utilize CNNs to extract embeddings that represent facial features, enabling robust face verification and identification.
- **Generative models:** CNNs serve as the building blocks for Generative Adversarial Networks (GANs) [15], discussed in Section 2.2.2, and their extensions, enabling tasks such

as image synthesis and super–resolution.

2.2 Generative models

Unlike discriminative models, which predict labels or outputs given an input ($P(y|x)$), generative models are a class of machine learning methods that aim to generate new samples resembling the input data, by learning their underlying probability distribution $P(x)$ or $P(x, y)$, thus making them suitable for tasks such as data synthesis and augmentation or unsupervised learning. This section covers the evolution of generative models, with particular attention to the one central for this work.

2.2.1 Evolution of generative models

First generative models relied in probabilistic approaches, like Restricted Boltzmann Machines (RBMs) [32, 33] consisting of shallow, stochastic neural networks, were built on energy–based models and used Gibbs sampling to generate data. Deep Belief Networks (DBNs) [34] extended RBMs by stacking multiple layers, enhancing their representation power. The main drawback of these techniques was the high computational cost, making their scaling to complex datasets unfeasible in practice.

Unlike RBMs and DBNs, VAEs [35] employed an encoder–decoder structure combined with probabilistic reasoning. The encoder maps input data into a latent space, and the decoder generates new data by sampling from this latent representation. Their probabilistic nature allows for smooth interpolation in the latent space, making them particularly suited for applications such as image synthesis and anomaly detection.

Generative Adversarial Networks (GANs) adopt an adversarial training framework, where two neural networks — a generator and a discriminator — compete in a zero–sum game, allowing GANs to generate highly realistic data without explicitly modelling the data distribution. Due to their importance in the context of this work, GANs are explored in greater detail in 2.2.2.

More recently, diffusion models [36,37] have emerged as a promising alternative for generative tasks. These models work by simulating a forward diffusion process that gradually adds noise to the data until it is indistinguishable from pure noise. After this preliminary step, a reverse process denoises its output until the original data is obtained. Diffusion models are demonstrating state–of–the–art performance in generating high–quality data and are increasingly being adopted in applications such as text–to–image synthesis and audio generation.

2.2.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks were introduced by Ian Goodfellow et al. in [15] as a new paradigm for generative modelling. The major novelty with respect to the state–of–the–art at the time was the introduction of the adversarial training scheme that gives the name to these networks. When GANs were released, probabilistic models such as RBMs or DBMs relied on computationally intensive algorithms, making them unfeasible for practical use cases, except for the most trivial scenarios, often forcing them to being processed via approximation techniques such as Markov Chain Monte Carlo (MCMC) sampling. Another approach was, for example,

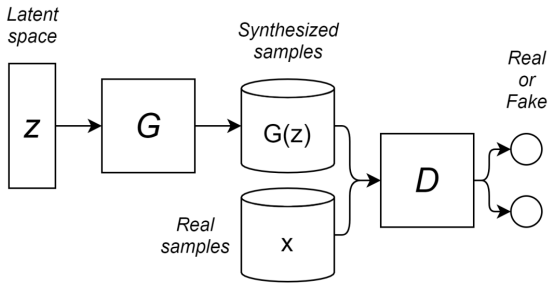

Figure 2.3: A summary of the GAN architecture composition, as reported in [39].

noise–contrastive estimation (NCE) [38], which relied on predefined noise distributions, limiting its effective functioning.

GANs overcame these issues by introducing a pipeline in which a generative model is directly trained, without explicitly modelling the data distribution. In fact, in the new approach presented, two neural networks — the generator and the discriminator — compete in a zero–sum game, allowing for a much more lightweight training phase from a computational point of view with respect to traditional approaches. GANs have made data generation possible in many real–world applications, such as image synthesis or data augmentation.

Mathematical framework

Below are some definitions that establish the nomenclature and theoretical framework for characterizing GANs, with Figure 2.3 providing a visual representation of the features here described.

Definition 2.1. The **latent space**, denoted here as $\mathcal{Z} \subseteq \mathbb{R}^d$, is a d–dimensional vector space whose elements, referred to as **noise vectors**, are sampled from a predefined probability distribution $p_z(z)$, typically uniform or Gaussian.

Definition 2.2. Given a noise vector $z \in \mathcal{Z}$, sampled from a prior distribution $p_z(z)$ defined over the latent space $\mathcal{Z}$, the **generator** G is a differentiable function $G : \mathcal{Z} \to \mathbb{R}^n$, parametrized by θ_g, where θ_g represents the weights and biases of the neural network implementing G. The generator transforms z into a synthetic sample $G(z; \theta_g)$ in the data space $\mathbb{R}^n$.

Definition 2.3. The **discriminator**, denoted as D, is a differentiable function $D : \mathbb{R}^n \to [0, 1]$, parametrized by θ_d, where θ_d represents the weights and biases of the neural network implementing D. The output $D(x; \theta_d)$ indicates the probability that a sample $x \in \mathbb{R}^n$ is real (i.e., drawn from the true data distribution $p_{\text{data}}(x)$) rather than generated by G.

In other words, the latent space $\mathcal{Z}$ provides a compact and abstract representation of the data distribution, from which noise vectors z are sampled and used as input to the generator. The generator is trained to approximate the training data distribution $p_{\text{data}}(x)$ by generating samples $G(z)$ that cannot be distinguished from real data by a discriminator function D, trained to maximize its ability to correctly classify real samples as close to 1 and synthetic samples as close to 0.

This adversarial dynamic forms the core of GAN training, driving both networks to improve iteratively and enabling the generator to approximate the true data distribution $p_{\text{data}}(x)$.

The adversarial interaction is formalized as a min–max optimization problem:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z\sim p_z(z)} [\log(1 - D(G(z)))] .$$

At equilibrium, the generator produces samples indistinguishable from real data, and the discriminator's optimal output becomes:

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}.$$

Training alternates between optimizing D and G using gradient–based methods. The discriminator maximizes $\log D(x)$ for real samples and $\log(1 - D(G(z)))$ for synthetic samples, improving its classification performance. The generator, in turn, minimizes $-\log D(G(z))$, encouraging the production of samples that are more likely to be classified as real by D. This iterative process ensures that G progressively captures the true data distribution.

Technical challenges in GANs

GANs faced multiple challenges, in particular during their early stages. One of the major issues has been the instability of the training process, mostly due to its characteristic adversarial nature. The min–max optimization framework used for training, in fact, needs a perfect balancing of the generator and the discriminator networks, which have to compete almost evenly, since if either one of them dominates the training process, it tends to become unstable, leading to poor performance. For example, if D becomes too good in finding the reconstructed samples, G may struggle to produce realistic samples, since it will not obtain enough information from D's performance, which would be useful to improve its reconstruction ability. Conversely, if G becomes too effective in generating realistic samples, D may struggle to distinguish them from real data, making the adversarial process ineffective and leading to a lack of meaningful feedback for G to improve further. One of these two situations could lead to a stall in the learning process, as highlighted in [15] itself. Additionally, GANs are characterized by another stability issue, also due to the adversarial nature of training, that is the difficulty in determining whether it actually ends, due to its often oscillatory behaviour. This makes it challenging to identify when the model has reached equilibrium, also because the loss function is non–convex in nature while GANs are highly sensitive to hyperparameters tuning. Therefore, the tuning phase must be carried out carefully to avoid instability or suboptimal convergence.

Another typical issue is the so–called **mode collapse**, where the generator just focuses on deceiving the discriminator without actually diversifying the generated examples, resulting in a lack of variety and the reproduction of only a limited subset of the data distribution. Mode collapse has been widely observed both in early stages of GANs and in more recent literature (see Section 2.2.2). Early GAN architectures also struggled with maintaining fine details and realistic textures in generated samples, particularly at higher resolutions. The generator's ability to synthesize complex patterns often degraded as the dimensionality of the data increased.

Finally, evaluating GAN performance is often challenging, since they lack a well–defined objective for assessing the quality of the reconstructed samples, unlike standard supervised learning models, which benefit from clear metrics such as accuracy or mean squared error. Instead, GAN evaluation relies on surrogate metrics that try to capture specific aspects of generative performance, such as the diversity and realism of the generated samples.

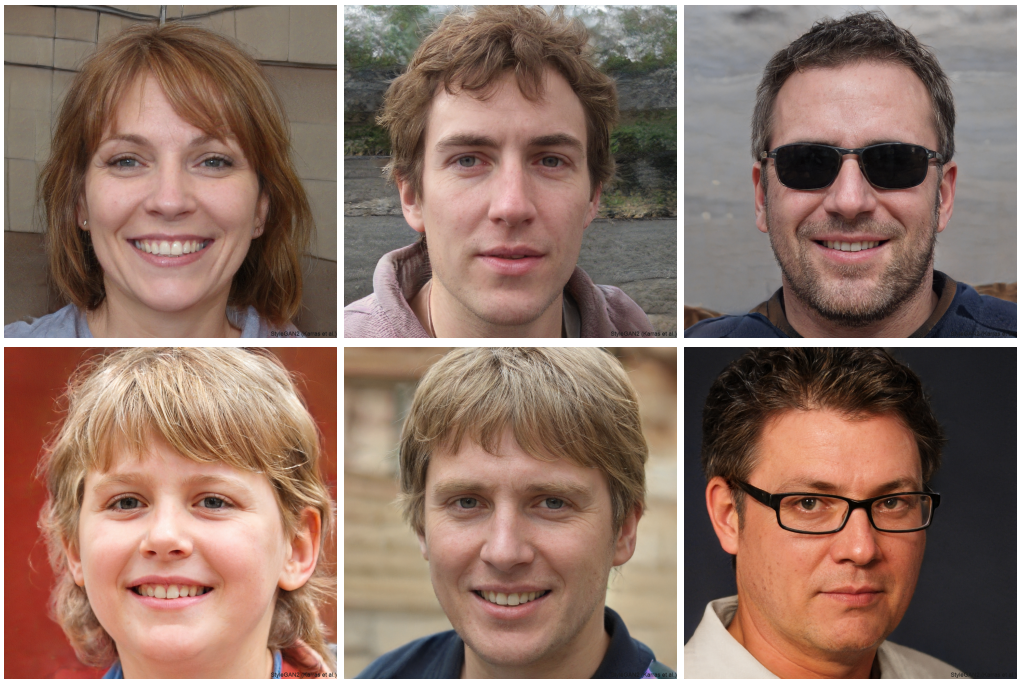

Figure 2.4: Examples of human faces generated with StyleGAN2, randomly selected sourcing from the website ThisPersonDoesNotExist.com.

Evolution of GANs

To address the issues pointed out in Section 2.2.2, the structure and the training procedure of GANs have gradually evolved. For example, Arjovsky et al. [40] replaced the traditional loss function with the Wasserstein distance — from which the name Wasserstein GANs (WGANs) — improving training stability by providing smoother gradients even when the discriminator is not optimal. WGAN–Gradient Penalty (WGAN–GP) [40] incorporated a gradient penalty, to further reduce vanishing gradient problems and ensure better regularization. The oscillatory behaviour is still an open issue in GANs' training, but an important breakthrough in this direction was achieved with Deep Convolutional GANs (DCGANs) [41], which introduced convolutional architectures, batch normalization and stride convolutions in place of pooling layers, to provide more reliable feedback for the generator and stabilize the overall training process.

Mode collapse has also been a persistent problem in GANs. To address this, approaches like Manifold–Guided GANs (MGGANs) [42] encouraged the generator to explore diverse modes within the data distribution, promoting variety in the generated samples. Moreover, SoftGAN [43] mitigated this issue by softening the discriminator's decision boundary, allowing the generator to generalize more effectively and produce a broader range of outputs.

As for the generation of data in the form of images, which is one of the most common usages of this architecture, early GANs often struggled to reproduce fine details and realistic textures at higher resolutions. Progressive Growing GANs (PGGANs) [44] revolutionized the field by changing how the training is performed (more details about this approach are given

in the following). Starting from PGGANs, BigGANs [45] further enhanced the performance in generating high resolution (HR) images through the use of larger batch sizes, architectural refinements, and advanced training protocols; moreover, Self–Attention GANs (SAGAN) [46] introduced self–attention mechanisms to capture long–range dependencies, to provide more consistent images.

Finally, GANs' evaluation commonly relies on metrics such as the Fréchet Inception Distance (FID), which measures the similarity between the distributions of real and generated data in a feature space, and the Inception Score (IS), which evaluates both the quality and variety of generated samples. Both of these metrics fail to represent the actual human perception and often require significant computational resources [47]. Additionally, models like CycleGAN [48] and StarGAN [49], while primarily focused on tasks such as image–to–image translation, highlighted the need for robust measures of fidelity and diversity, inspiring further research in this area.

StyleGAN family

StyleGAN models build on the PGGAN, where common training instability issues, typical in GANs, are specifically addressed. The proposed training scheme involves the growing of the network during training: the network first learns how to generate low–resolution 4×4 images, which grow to high–detail 1024×1024 images by adding deep layers to the network, dividing the task in smaller and easier to learn sub–problems. This technique allows the network to first learn core structures and, subsequently, fine details, stabilizing and accelerating the model training.

The first StyleGAN model has been a major breakthrough in realistic synthetic face generation technology, incorporating style transfer methodologies [50] to improve the generator's architecture. Indeed, contrary to what usually happens, in StyleGANs the latent vector z is not fed directly to the generator, but the images are generated from a learned constant vector of fixed size, while the z vector is mapped into an intermediate space, allowing independent control of different image features.

StyleGAN2 significantly improved the quality of generated images (some of which are illustrated in Figure 2.4) with respect to the original StyleGAN, especially in regard to the presence of minor but recognizable artefacts, such as droplets of colour blobs.

This was obtained by: (1) introducing an improved regularization loss focused on controlling the smoothness of the mapping from the latent space to the output; (2) reorganizing and simplifying the normalization technique used for the original model; (3) removing the constraint of generating images at each resolution defined during the progressive growth procedure and replacing it with a sum of outputs at different resolutions; (4) implementing skip connections [16].

The latest model of the family, StyleGAN3, improves its predecessor by addressing its propensity to fix finer details (such as hair texture) to specific image coordinates, which was due to the prevalence of image borders and the presence of aliasing patterns in generated faces. StyleGAN3 introduced sufficient zero–padding around the image to reduce the border effect, reformulated the operations to work on continuous image representations to avoid the introduction of aliasing artefacts, simplified the network structure with respect to StyleGAN2, and removed some regularization that was previously introduced. Furthermore, translational and rotational invariance is achieved by substituting the learned constant introduced in StyleGAN2

with information–wise equivalent Fourier features and by substituting 3×3 with 1×1 convolutions. Finally, low–pass filters were applied to upsample and downsample operations to further reduce the generation of artefacts [51].

Applications and risks of GANs

GANs have impacted multiple fields thanks to their ability to synthesize high–quality and realistic data. As already mentioned, image synthesis is one of the most common ones, with results such as photorealistic subjects of human faces or landscapes, but also artworks and virtual environments. Similarly, GANs can also be used for domain adaptation, where they map data from one domain to another, such as translating satellite images to maps or converting sketches to detailed artworks [52]. They can generate not only static images, but they can also be used for video generation and editing, with animations and deepfake content.

Another fundamental application of GANs and other common generative models is data augmentation, consisting in the expansion of datasets so to improve performance of machine learning models, especially in scenarios with limited training data [53], such as medical imaging [54] or data regarding rare diseases, where diagnostic algorithms often do not have enough data to work with.

Finally, GANs are increasingly used in natural language processing (NLP) for text–to–image generation [55] and synthetic speech creation, facilitating applications in virtual assistants, video dubbing, and audiobook narration.

This potential comes with the risk of the generation of content with malicious intent, with highly realistic fake data proliferating on the internet, creating important challenges in misinformation, fraud and identity theft. Deepfake technology has been exploited to produce fake news, political propaganda, and even forged evidence in legal contexts [18, 19]. In addition to ethical concerns, GANs introduce security vulnerabilities, with adversarial attacks exploiting GAN–generated data to deceive machine learning models, thus with repercussions on critical applications such as autonomous vehicle driving, financial fraud detection, and biometric authentication [15]. Privacy concerns also arise, as GANs can be exploited to reconstruct synthetic data resembling sensitive information from training datasets, potentially violating data privacy regulations such as GDPR. This issue is particularly critical in scenarios involving sensitive personal or medical data, where adversarial models can inadvertently leak private information. For instance, the application of Privacy–Preserving GANs (PPGANs) has been explored to mitigate this risk, but significant challenges remain in balancing utility and privacy [56, 57]. Additionally, there is also a matter of transparency regarding the origin of training data, since datasets used to train GANs are often collected from the internet or other large–scale repositories without clear documentation of their authorship [58, 59]. For instance, personal images scraped from social media or proprietary datasets could be used during the training phase, potentially violating laws such as GDPR. Moreover, the lack of accountability in data sourcing can result in biases within the training data, leading to harmful outputs and difficulties in the attribution of responsibility in cases where GAN–generated content is misused [60]. Last, but not least, another aspect to consider is the environmental impact of training GANs. Large–scale models require substantial computational resources, leading to significant energy consumption and carbon emissions [61].

2.3 Transfer learning paradigms

Transfer learning is a machine learning technique that allows knowledge acquired from performing a certain task to be applied to a different but related one. The idea is that certain features learned while performing the former task, especially in the first layers of deep neural networks, are generic enough to be used also for other domains. This approach is generally useful in scenarios where obtaining labelled data for the target task is challenging or resource–intensive, since leveraging pre–trained models significantly reduces the computational cost and the amount of data required, and also allows a better performance, thus making this strategy beneficial for multiple kinds of tasks.

Transfer learning techniques can be categorized based on the way knowledge is transferred. This section provides an overview of the most common paradigms, as described in [62–64].

Feature extraction – One of the most common applications for transfer learning is feature extraction, where the pre–trained model is leveraged just to extract useful features that are used for the task of interest. This works under the assumption that the features captured by the first layers of a neural network are generic enough to be leveraged for multiple tasks. A simple way to do this is by freezing the first layers of the model that is being used as a support and just retrain its final layers specifically for the new task.

Fine–tuning – This approach is similar to the previous one, with the difference that the weights of the pre–trained model are all updated during the new training, since they are used as an initial configuration of the target model, with all the network or some chosen layers being retrained. Fine–tuning is more beneficial when the source and target domain are similar, as it helps the model adapt its features to particular details of the target task.

Domain adaptation – Domain adaptation aims to transfer knowledge between a source domain and a target domain with different data distributions, typically by minimizing the distribution discrepancy between the two, using metrics such as Maximum Mean Discrepancy (MMD) or adversarial learning frameworks. Domain adaptation is usually useful in scenarios where labelled data is scarce in the target domain but abundant in the source domain.

MultiTask Learning (MTL) – This is an extension of the previous method, with the difference that MTL optimizes multiple tasks at the same time within a shared model, rather than adapting knowledge between distinct domains, enhancing the performance of the model in all of the tasks.

Zero–Shot (ZSL) and Few–Shot Learning (FSL) – ZSL and FSL share the same goal, that is, generalizing to novel tasks or categories with little to no labelled data in the target domain, often relying on embedding spaces or auxiliary information such as semantic attributes to infer from known to unknown classes.

In this thesis, weights pre–trained on a face segmentation task are utilized as an initialization point for training a fake detection network. This strategy comes from the inherent correlation between the tasks, since the knowledge needed to parse the information contained within the

face is beneficial for both of them. In [65], one of the two key works on which this thesis is built, two approaches are explored: retraining only the final layers of the network while freezing the backbone (feature extraction) and retraining the entire network (fine–tuning). As the results show, both strategies significantly reduce the required training time and sample size while improving the classifier's performance. More details about this procedure are presented in Parts II and III

2.4 Explainability and interpretability

In this context, the term explainability refers to the clarification of the decision mechanisms of a model, in order to make its functioning more transparent and understandable to humans. Since the adoption of deep and complex models like large neural networks, often operating as "black boxes" [66], this concept has become a critical concern, especially in ethically delicate domains, such as healthcare, finance, and autonomous system, where trusting the reasoning behind a model decision is crucial for its usability.

A concept closely related to, but distinct from explainability is interpretability, often used —improperly — as a synonym. However, while explainability usually involves generating external explanations for models that are black boxes by design, interpretability focuses more on the intrinsic transparency of a model. A comprehensive taxonomy of methods for both interpretability and explainability is outlined in the literature [67–72], where these methods are often categorized based on several aspects, listed below.

- **Purposes of explainability and interpretability:** some methods can be designed to create intrinsically interpretable models or to explain black box models post–hoc, but also to enhance the fairness of predictions and testing the sensitivity of model outputs.

- **Model–specific vs. model–agnostic:** some methods are model–specific, meaning they apply only to particular types of models, while others are model–agnostic and can be used for any ML model.

- **Local vs. global methods:** local methods explain individual predictions, while global methods provide an overview of the entire model behavior.

- **Data type:** techniques vary depending on the type of data they are applied to, such as tabular data, images, text, or graphs.

For instance, global interpretability techniques aim to explain the overall behavior of the model, while local methods, such as LIME [73], focus on explaining a single prediction. As highlighted in [66], intrinsic methods focus on designing interpretable models, such as decision trees or linear models, while post–hoc methods explain pre–trained black box models using tools like feature importance measures, surrogate models, or visualization techniques. In the context of this thesis, the focus will be on a post–hoc, model–agnostic method giving model–specific explanations, which will be discussed in detail in the following sections.

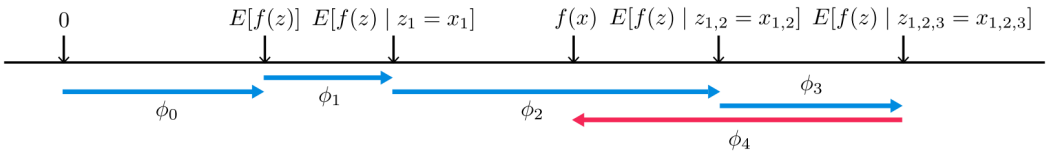

Figure 2.5: The expected base value $E[f(z)]$ is the predicted value of the model without any known features and $f(x)$ is the current output of the model given the input x. The diagram shows how SHAP values attributed to each feature change the expected model prediction when conditioning on that feature.

2.4.1 Explainability via Shapley Value analysis

SHapley Additive exPlanations (SHAP) is a powerful explainability method based on the Shapley value, introduced by Lloyd Shapley in 1953 to address the problem of fairly distributing the total contribution generated by a group of agents among individuals in a cooperative game [74]. In order to properly introduce the Shapley value, it is necessary to mention some Game Theory concepts [75].

A *game* is a set of circumstances whereby two or more players contribute to an outcome. Let N be a set of players, $S \subseteq N$ is a coalition and N is the grand coalition. The characteristic function v assigns to each coalition S the best results that the players in S may obtain, i.e., the total payoff, independently of the choices of the other players.

Definition 2.4. Given a game with N players and characteristic function v, the **Shapley value** is the vector $\phi(v)$ whose component $\phi_i(v)$ is the average marginal contribution of player $i \in N$ with respect to all the permutations of the players [76].

The definition guarantees the existence and uniqueness of the Shapley value, which is one of the most commonly used point solutions and, according to the main equation that can be found in the original paper, represents the average expected marginal contribution of one player after all possible combinations have been considered, thus rewarding each agent for its individual contribution, taking into account cooperation with others [77].

The SHapley Additive exPlanations (SHAP) [78] method, that is exploited in this thesis, is based on the calculation of the Shapley value of the conditional expectation function of a model, in order to study its explainability. In the case of simple models, the best explanation method consists of the model itself. For more complex architectures, such as estimator ensembles and deep neural networks, however, it is necessary to introduce simpler explanation models that can be defined as interpretable approximations of the original one [78]. Local methods, in particular, explain a prediction $f(x)$, with f being the estimation model and x its input, using a simplified version of the input x' that maps to the original x via the mapping function $x = h_x(x')$. Given a local explanation model g, $g(x') \approx f(h_x(z'))$ must be ensured whenever $z' \approx x'$.

Additive feature attribution methods are those for which the explanation model is a linear function of two binary variables

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i, \tag{2.2}$$

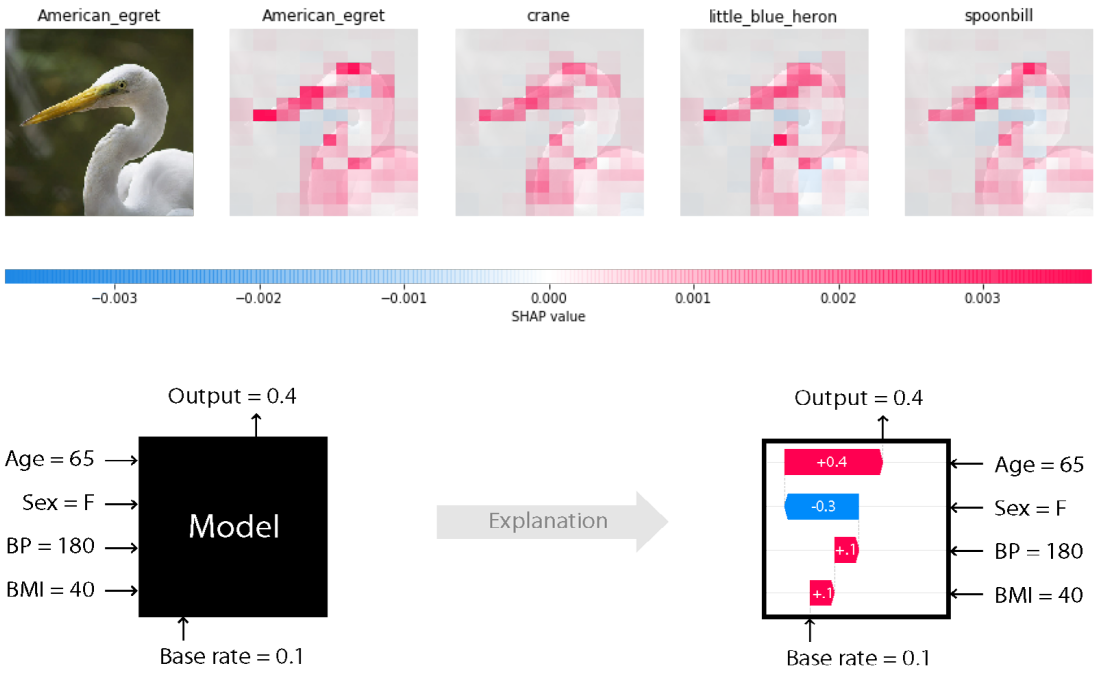

Figure 2.6: Application of SHAP to two different types of data. In the first image, the method is used to highlight areas that were important for an image classification task using MobileNetV2 [79], while the second example shows how the method works in a generic example with a different type of data. Both of the images are sourced from the official SHAP documentation page.

where $z' \in \{0,1\}^M$, M is the number of simplified input features and $\phi_i \in \mathbb{R}$ is the effect of feature i on the explanation model g. Methods which have explanation models that match this definition attribute an effect to each feature and the sum of their attribution effects approximates the output of the original model. Alternatively, the vector z can be interpreted as a coalition vector, with M being the maximum size of the coalition.

Several popular explanation methods for deep learning, such as LIME [80], DeepLIFT [81] and Layer–Wise Relevance Propagation [82], all satisfy Equation (2.2), as well as three explanation models based on the Shapley value, Shapley regression values [83], Shapley sampling values [84] and quantitative input influence [85].

Since the Shapley value is the only vector of values that relates to the properties of local accuracy, missingness and consistency defined in the original paper, the only additive feature assignment method that satisfies the statement must be based on the Shapley value, and methods not based on the Shapley value violate local accuracy and/or consistency, thus making SHAP, built on the Shapley value of a conditional expectation function of the original model, the only method adhering to the three desired properties and using conditional expectations to define simplified inputs. The definition of the method aligns perfectly with Shapley regression, Shapley sampling, and quantitative input influence, while allowing connections to LIME, DeepLIFT, and layer–wise relevance propagation feature attribution techniques. An illustration for the

construction of Shapley values is shown in Figure 2.5, while Figure 2.6 illustrates two examples of the application of the method.

2.5 Fake detection state–of–the–art review

Synthetic face detection approaches can be classified in three major branches [86]: physical–based methods, physiological–based methods, and DL–based methods.

- **Physical–based methods** focus on the identification of real world–related artifacts and inconsistencies, such as incongruous image illumination and reflections. Works belonging to this category leverage known problems of synthetic faces: eye pupil illumination, morphology and eyes symmetry are of particular interest, since current GAN generators struggle with this aspect.
- **Physiological–based methods** instead rely on the semantic aspect of human faces, searching for irregularities in face symmetry, pupil shape, iris colour and texture, and other salient characteristics. DL–based methods use deep neural networks trained to effectively and automatically extract salient features from the images to detect synthetic faces. Usually, large networks pretrained on other image classification tasks are used and fine–tuned, leveraging their already effective feature extraction capabilities. Both physical and physiological–based methods achieve higher–than–human performance (which ranges between 26% and 80% accuracy) and provide built–in interpretability. Their strength is however limited by the strong environmental constraints used to define these methods, such as frontal portrait pose and limitations with regard to face occlusion.
- **DL–based methods** use deep neural networks trained to automatically extract features from the images to detect synthetic faces. These models, typically fine–tuned from large pretrained networks, can achieve significantly higher performance than physical and physiological methods; however, they completely lack interpretability and operate like black box systems.

Recently, numerous efforts have been made in the field of DL–based methods for fake face detection, founded on both specialized datasets and tools, which were created specifically for this purpose. For instance, the OpenForensics dataset, introduced in [87], provides detailed face–wise annotations to aid in training models for multi–face forgery detection and segmentation in uncontrolled settings, enhancing research into deepfake prevention and face detection. Early works on DL–based synthetic face detection used what was at the time the state–of-the–art pretrained CNN architectures, such as the VGG–Net used in [88]. Other approaches include the a combination of both fine–grained frequency components and RGB colour values of the image [89, 90]. Additionally, other datasets and methods that incorporate facial landmark detection and segmentation are introduced in [91], paving the way for a more interpretable and granular solution to the face verification problem. In a similar way, ref. [92] tackles fake detection in a fine–grained classification approach, analysing facial features to improve detection over multiple datasets. In [93], a novel architecture called Gram–Net is introduced, which bases its detection power on global image texture features, highlighting texture as an important indicator of image authenticity. Other works, such as [94], focus on visual features that make the detection robust

to post–processing procedures by leveraging luminance, chrominance components, and colour space characteristics. In [95], DETER, a method that analyses neuron activation patterns across layers to identify synthetic images, is proposed, demonstrating its robustness to various GAN–generated images. Finally, in [96], a depth map–guided triplet network for effective fake detection is exploited, using depth information to enhance the detection accuracy by distinguishing real from fake faces based on discontinuity, inconsistent illumination, and blurring.

Despite these advancements, challenges remain. Both physical and physiological methods achieve performance higher than human accuracy (26%–80%) but are limited by environmental constraints. Conversely, DL–based methods surpass these approaches in terms of raw performance but lack transparency and interpretability. This gap has motivated recent efforts to explore techniques that provide a more explainable detection process.

Part II

Methodology and implementation

Chapter 3

Proposed methodology

This chapter describes the entire pipeline designed to detect generated faces, from data collection and feature extraction to semantic segmentation, classification and explainability of the results. The chapter concludes with some implementation details, fundamental for the reproducibility of the procedure.

3.1 Datasets

The complete version of the Mut1ny [97] dataset was employed to train the background removal module. The dataset, comprising 70,621 images, has already been used in the literature [98–100] and features a wide range of subjects of different ethnicities, ages, genders, facial poses and camera angles, with pixel–level labels hand–created by the Mut1ny team and other volunteers, resulting in fourteen different classes, as shown in Figure 3.1. A free 'community edition' of the dataset is also available [101], containing slightly less than a quarter of the images, fewer classes (no left/right differentiation), and not being updated.

For the fake detection task, the data were extracted from ArtiFact [102] (Artificial and Factual), a large collection of different datasets of real and synthetic images, including multiple categories such as people, animals, vehicles, and more. The dataset comprises 964,989 real images drawn from eight sources, to ensure diversity, and 1,531,749 synthetic images, obtained with twenty–five methods, specifically thirteen GANs [103] (such as StyleGAN, StyleGAN2 [16], StyleGAN3 [51], BigGAN [45], and CycleGAN [48]), seven diffusion models (such as Stable Diffusion and Latent Diffusion [104]), and five miscellaneous generators (such as Taming Transformer [105]), for a grand total of 2,496,738 different samples.

Since most of the real images contained in the dataset depict real–life objects, environments, and animals, a preliminary selection had to be made in order to extract a satisfactory amount of human subjects, specifically close–up images of faces, possibly including shoulders and part of the background. In particular, CelebA–HQ [106] and FFHQ [16] were used for the *Real* class (see Figure 3.2). The former is a high–quality version of the CelebA [107] dataset, comprising detailed images of celebrity faces, while the latter is a high–quality dataset of human faces taken from Flickr, thus offering a wide range of diverse samples. Synthetic images are generated within the same categories as the real images to maintain consistency in the dataset. Text–to–image and inpainting generators utilize captions and image masks from the COCO [108] dataset, while noise–to–image generators use normally distributed noise with different random seeds.

As for the *Fake* image class, the images employed in the analyses were generated using StyleGAN2 and StyleGAN3. These two models have been chosen due to their popularity and prevalence in related research works and since they are capable of generating high–quality syn-

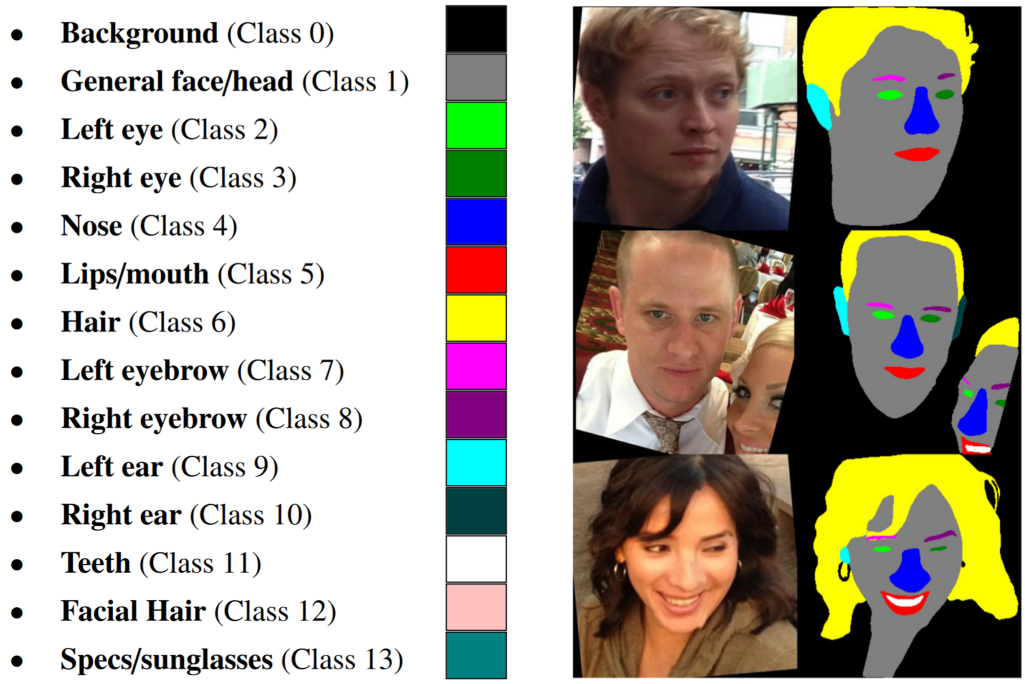

Figure 3.1: Examples of segmentation masks in Mut1ny dataset. In the experiments, the segmentation module was only used to finely remove the background in the images. To ensure a good variety of training samples, the facial images are drawn from different ethnicities, ages and genders, with a wide facial poses (angle range from –90 to 90 degrees), and are randomly rotated.

thetic facial images.

To accurately reflect real–world conditions, both real and synthetic images in the ArtiFact dataset undergo various impairments. These include random cropping with a ratio of $r = 5/8$ and crop sizes ranging from a minimum of 160 to a maximum of 2048 pixels, resizing to 200 $\times$ 200 pixels, and JPEG compression with quality levels between 65 and 100.

3.2 Feature extraction

The architecture used in this work is mainly based on DeepLabV3+, which is a model belonging to the DeepLab [109] widely used family of semantic segmentation models developed by Google Research. In the presented version of the model, a MobileNetV3 Large [110] is used as a feature extractor, providing a lightweight alternative to the original Xception [111] backbone, which, although offering strong segmentation and classification performance, requires significantly more computational resources. MobileNet still achieves competitive performance with fewer parameters by leveraging depthwise separable convolutions and efficient block designs. A ResNet50 [7] backbone was also tested to explore a middle ground between MobileNetV3 and

(a) Example data taken from the Tensorflow CelebA-HQ dataset catalog page, as of 07/01/2025.

(b) Teaser figure for the Flickr-Faces-HQ Dataset. Taken from the official NVlabs GitHub repository, accessed on 07/01/2025.

*Figure 3.2: The datasets used for the Real class samples. In (**a**), some examples from CelebA–HQ dataset are collected, which provides high–quality images of celebrity faces. Images are retrieved from the official TensorFlow dataset documentation page. In (**b**), there is a teaser figure for the Flickr–Faces–HQ dataset, which offers a diverse set of human face images sourced from Flickr. Images are retrieved from the NVlabs ffhq–dataset GitHub repository.*

Xception, needing more parameters with respect to the former, but significantly less than the latter.

The key innovation introduced in DeepLabV3+, with respect to its predecessor, is the integration of Atrous Spatial Pyramid Pooling (ASPP), allowing the model to precisely capture multi–scale information, crucial for refining segmentation boundaries. More precisely, this technique combines atrous convolution and spatial pyramid pooling (SPP [112]) operations to allow the detection of significant information independently of the scale, thus making it particularly indicated for semantic segmentation tasks. During the atrous convolution operation, gaps are introduced inside the kernels; specifically, if a rate r is specified, $r-1$ zeroes are introduced between consecutive filter values, effectively enlarging a kernel of size $k \times k$ to size $k_e = k + (k-1)(r-1)$. This way, the receptive fields of the neurons are expanded — allowing them to capture features across various scales — without increasing the number of parameters of the model. On the other hand, SPP is a particular pooling technique that enables the creation of outputs of uniform length, independently of the input images size, overcoming the fixed input size limitations of current CNN models. These architectures commonly present a first feature extraction section composed of convolutional layers followed by a series of fully connected layers. Convolutional layers operate using a sliding–window approach, thus, they do not require a fixed input size since they can produce feature maps of arbitrary size. On the contrary, the fully connected layers require a fixed input size. This fixed size requirement is normally satisfied either by cropping or resizing the input images, which could result in a loss of information. In SPP, a custom layer on top of the convolutional part of the CNN is inserted, where the input feature maps are segmented into multiple bins at varying scales, pooling each one separately for feature extraction, and then concatenating these features to combine them. This operation results in k M–dimensional vectors where M is the number of bins and k is the number of filters in the last convolutional layer, resulting in a fixed–size output vector that can be processed by the following section of

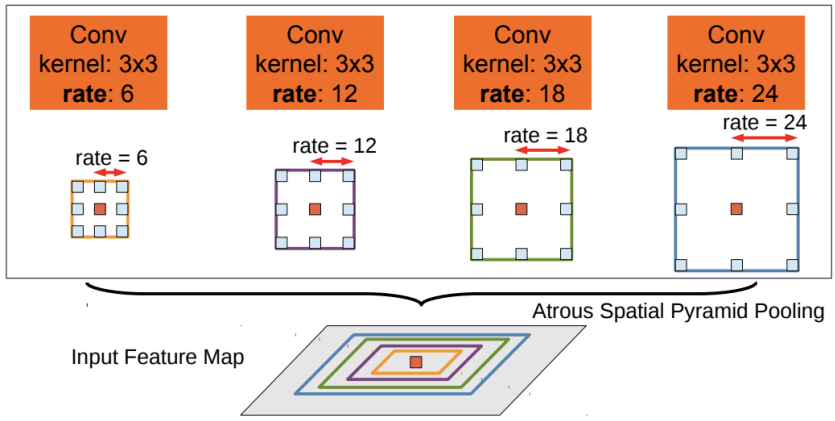

Figure 3.3: The ASPP module as proposed in [109].

the network.

A visual representation of the ASPP module can be seen in Figure 3.3.

3.3 Semantic segmentation and background removal

For the background removal phase, DeepLabV3+ was trained on the Mut1ny dataset to obtain the semantic segmentation of images — Figure 3.4 shows the proposed architecture. Augmentation was applied on the training set: specifically, random rotations within a range of ±15 degrees to emulate different head poses, horizontal and vertical translations, shifting images up to 10% of their original dimensions to adjust for alignment discrepancies, shear transformation of up to 0.2 radians to modify the image geometry, imitating a shift in perspective, random zoom by up to 20% to replicate variations in distance from the camera, and horizontal flips to represent different orientations. For the new pixels generated by rotations or shifts in width and height, a "nearest" filling method was applied to maintain local pixel similarity. All augmentation parameters were deliberately kept within moderate ranges, ensuring that the transformations (±15 degrees rotations, 0.2 radians shear) introduced variability without significantly altering the realism of the images. The ASPP module incorporated dilation rates of 1, 4, 8, and 16 to achieve a balance between large contextual information and finer details. Furthermore, the kernels of the ASPP layer have been chosen in {3, 5, 7, 11} to optimize the model ability to capture contextual details.

The previously trained segmentation module is applied to both real and generated images, and the class *Background* is isolated from the others, which are collapsed into one to obtain a fine foreground/background separation and provide the second version of the same datasets needed for the subsequent experiments. Then, both the original and the segmented images are fed separately to the model for the classification phase.

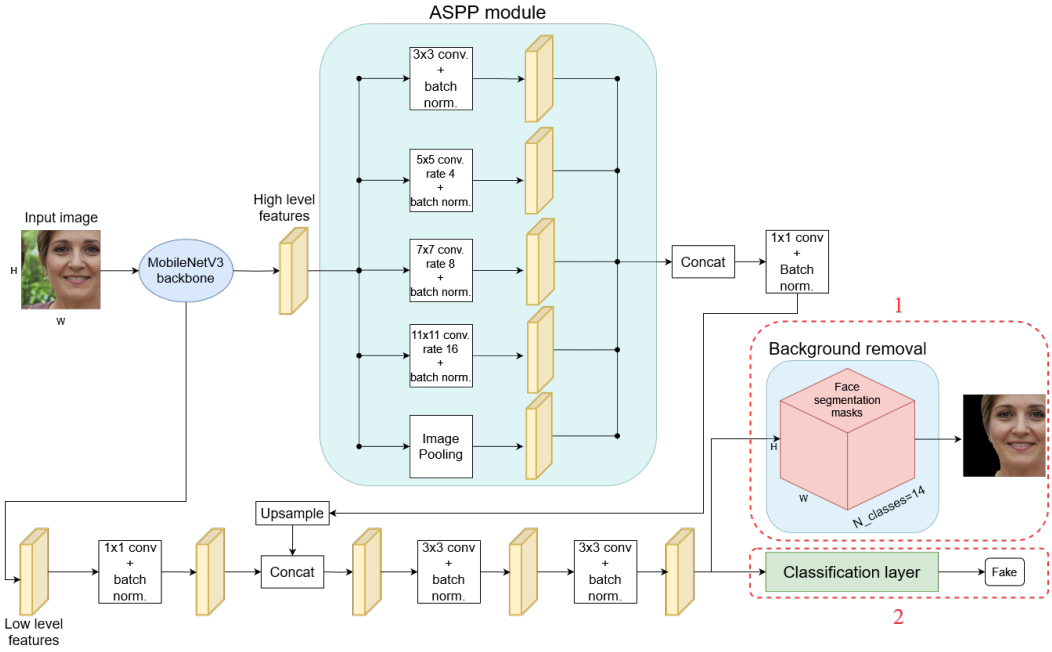

Figure 3.4: The proposed network architecture. The model serves for both face segmentation and real/fake classification tasks. The images are first processed by the background removal head, to obtain their segmented version; then, the actual fake classification procedure is run on both the original and the cropped images separately. The main core of the model is the same for both tasks, with only the head determining its final result. In a first phase the model is trained to only perform an image segmentation task — depicted by the flow which ends at the red dotted rectangle labeled '1' —, obtaining a mask representing the fourteen different segmentation classes. After this procedure, the 0 class still represents the background, while all the classes ranging from 1 to 13 are collapsed into one single foreground class. The model trained this way is used first to infer on the ArtiFact dataset, obtaining its finely cropped version. In a second, separate and successive, phase — represented by the flow ending at the red dotted rectangle labeled '2' — the model performs the classification task on both versions of the dataset.

3.4 Detection of GAN–generated faces

The classification task focuses on distinguishing real facial images from those generated by StyleGAN2 and StyleGAN3 and relies on the modified version of the DeepLabV3+ architecture also used for the segmentation process. For the detection of GAN–generated faces, the model utilizes a binary classification head that operates on both the original and the segmented images, the latter obtained after the procedure detailed in Section 3.3, which provides a new input for the classifier. Once the datasets for the original and segmented images are prepared, the model processes both inputs independently through the selected feature extractor (backbone), and the resulting extracted features are passed through the ASPP module. The resulting feature maps are upsampled dynamically to ensure alignment with the original input dimensions, and the

output is fed into a final convolutional layer, needed to obtain a binary classification of the input.

3.5 Transfer learning for classification

Transfer learning was implemented to enhance the detection capabilities of the classification models by leveraging the knowledge learned in the segmentation task. Specifically, the weights obtained from the segmentation task were saved and used as a pre–trained base for the subsequent deepfake classification problem.

Two transfer learning approaches were considered: first, a partial transfer learning setting was employed, in which only the head layers of the pre–trained model were retrained for the classification task, while keeping the backbone weights frozen. According to the categorization given in Section 2.3, this approach falls under the class of feature extraction techniques, leveraging the fact that the initial layers of the network can extract meaningful and task–agnostic features, leaving the final layers to adapt to a specific task. On the other hand, a full transfer learning strategy was also adopted, where all the layers of the pre–trained model were set to be retrainable on the classification task. Using the same categorization as before, this approach falls within the fine–tuning techniques, and it allows the model to adjust not only the final layers but also the underlying feature extraction layers, to better capture domain–specific details.

The rationale behind both transfer learning strategies here adopted lies in the correlation between the segmentation and classification tasks, since the network is able to learn latent representations that can highlight inconsistencies in facial geometry or artifacts in synthetic images, both common in GAN–generated images, making this approach beneficial for the task not only in terms of classification performance and model robustness, but also optimizing computational resources, thus reducing training times.

3.6 Explainability through SHAP analysis

In this work, SHAP was integrated into the methodology to evaluate the predictions of the binary classification model discussed in Section 3.4 and to assess whether the model's decisions aligned with meaningful facial features or relied on less relevant cues, such as background inconsistencies or artifacts, providing insights into which features are most influential in distinguishing real images from synthetic ones.

Given SHAP's high computational cost, due to the fact that it requires evaluating all possible feature combinations, the analysis pipeline was designed to operate on a subset of the test dataset, including both real and GAN–generated images. For each image, the model's prediction was decomposed into contributions from individual input features. Pixel coalitions were treated as features that were analyzed both qualitatively and quantitatively to determine the model's focus areas, and SHAP values were computed to quantify the impact of each coalition on the classification outcome. In particular, the SHAP framework highlighted regions of high importance contributing positively to the prediction and other detracting from it.

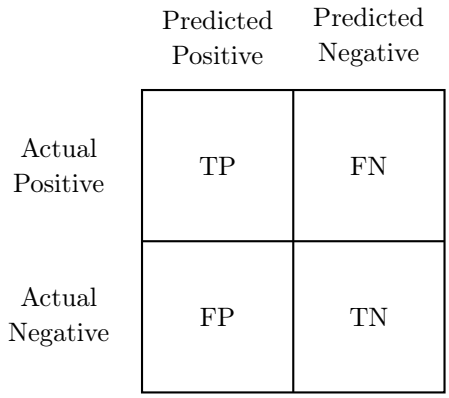

Figure 3.5: Generic confusion matrix for a binary classification problem. In this work, the positive (True) class represents generated samples.

3.7 Evaluation metrics

Since this study involves both segmentation and classification tasks, different metrics were used to evaluate the performance of the models. For the semantic segmentation task, pixel–wise accuracy was employed, which measures the ratio of correctly classified pixels with respect to the total number of pixels in the dataset, providing an overall measure of how well the segmentation model identifies each pixel class. As for the actual fake detection, since this was a binary classification problem, the metrics chosen to evaluate the performance of the model were those usually employed for this kind of tasks, specifically accuracy, precision, recall and F1–score, as defined in Equations (3.1) to (3.4).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

$$\text{F1–score} = \frac{2TP}{2TP + FP + FN} \tag{3.4}$$

Observe that the F1–score can be interpreted as a weighted average of the precision and recall to provide a more balanced measure in case of imbalanced datasets.

Confusion matrices for the classification task have also been plotted to visualize a detailed breakdown of the model's performance in distinguishing between real and fake images, plotting the numbers of true positives, true negatives, false positives, and false negatives obtained in the classification. In general, a confusion matrix for a binary classification problem has the structure shown in Figure 3.5.

Additionally, the Matthews Correlation Coefficient (MCC) was considered for evaluation. The MCC takes into account all four quadrants of the confusion matrix (TP, TN, FP, FN) and provides a balanced measure even if the classes are of significantly different sizes. Unlike the

F1–score, which primarily focuses on the positive class and balances precision and recall, the MCC provides a comprehensive evaluation of the classifier performance across both classes.

The MCC is defined as

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.5}$$

The MCC ranges from -1 to $+1$, where $+1$ indicates a perfect prediction, 0 indicates random prediction, and -1 indicates no correct predictions.

3.8 Implementation details

All the models are implemented using TensorFlow and Keras, and the backbone architectures are initialized with pre–trained weights from ImageNet, with support for both ResNet50 and MobileNetV3 Large. The various tasks and approaches can be reproduced in a modular and efficient way by changing certain variables that act as hyperparameters, to adapt both to segmentation and classification tasks.

A `DynamicUpsample` custom layer is implemented to dynamically resize feature maps to match the spatial dimensions of reference tensors using bilinear interpolation. Feature extraction backbones, which can be set as trainable or frozen depending on the transfer learning policy, are configured with specific layers for high–level and low–level feature extraction (high: `conv4_block6_2_relu` for ResNet50, `expanded_conv_12/project` for MobileNetV3, and `block14_sepconv2_act` for Xception; low: `conv2_block3_2_relu`, `expanded_conv_3/project`, and `block4_sepconv2_act`), respectively. ASPP is implemented with dilation rates of $[1, 4, 8, 16]$. Kernel sizes increase proportionally with dilation rates, following the formula:

$$\text{kernel size} = 3 + \text{int}(\text{rate} \times 0.5)$$

For example:

- with a dilation rate of 1, the kernel size remains 3;
- with a dilation rate of 4, the kernel size becomes 5;
- with a dilation rate of 8, the kernel size becomes 7;
- with a dilation rate of 16, the kernel size becomes 11.

This ensures that larger dilation rates correspond to larger kernel sizes, enabling the model to capture broader contextual information.

For segmentation tasks, a 1×1 convolution produces pixel–wise classification outputs, while, for classification tasks, a dense layer with a sigmoid activation produces binary outputs.

Images are resized to 224×224 pixels and normalized using backbone–specific preprocessing functions (`resnet50.preprocess_input` or `mobilenet_v3.preprocess_input`). As for the augmentation techniques, their usage is set as a parameter, so that their employment can be decided for each experiment. These methods, including random rotations ($\pm 15°$), translations

(±10%), shearing (0.2 radians), zooming (20%), and horizontal flips, are all applied using TensorFlow `ImageDataGenerator` or equivalent functions. A custom function `image_dataset_from_directory` is defined for the loading of the images, with batch sizes being configured dynamically based on the dataset split (e.g., 60% training, 20% validation, 20% testing). The transfer learning policy is also parametrized through a specific string variable, which can be set to `none`, `last`, or `all`, depending on the experiment being conducted.

Due to the imbalanced nature of for the two classes, class weights are calculated as:

$$\text{weight}_{\text{class}} = \frac{\text{total samples}}{2 \times \text{class samples}},$$

to ensure balanced contributions from real and fake classes during training.

The Adam optimizer, implemented using the `keras.optimizers.Adam` class from the TensorFlow library, is used with a learning rate of 0.01. For the loss functions, `keras.losses.SparseCategoricalCro` and `keras.losses.BinaryCrossentropy`, both part of the `tensorflow.keras` package are employed for the segmentation and classification tasks, respectively. Early stopping is applied with a patience of 10 epochs, restoring the best model weights when validation loss stabilizes.

Training metrics (loss, accuracy, etc.) are logged and visualized using Matplotlib and confusion matrices and performance metrics are saved for post–training analysis. Trained models are saved in HDF5 format with custom objects (`DynamicUpsample`) included for reproducibility. Results are organized into dedicated directories for models, performance logs, and plots. All the trainings were conducted on two NVIDIA RTX 4090 GPU with 24 GB VRAM, installed in two different machines. The `os.environ["CUDA_VISIBLE_DEVICES"]` variable is used to configure GPU settings dynamically based on the computer.

Part III

Experimental results and analysis

Chapter 4

Results and discussion

In this chapter, experimental results derived from the application of the methods presented in Part II are presented, together with their discussion. The results here analysed concern all the experiments conducted during the study, that is, the evaluation of the proposed methods in terms of segmentation performance and computational cost, impact of the background removal on the final classification, relevance of the transfer learning strategies and explainability analysis with SHAP.

4.1 Facial segmentation results

Some qualitative results of the application of the trained face segmentation module applied to both real and generated images are depicted in Figure 4.1. It is worth noting that, although the segmentation model was trained on the Mut1ny dataset, which includes both real images and 3D–rendered synthetic faces, some imperfections are still visible when segmenting AI–generated images. For example, in the central image of the third row, a green outline—belonging to the same class as the left eye—can be seen around the nose, which should instead fall under the general face region. Similarly, in the adjacent image to the right, a portion of the right earlobe is misclassified as a left ear, as evidenced by the light blue color. Similar misclassifications also appear in multiple other generated samples, while these phenomena seems to happen less frequently in genuine images, suggesting that, while the model generalizes reasonably well to GAN–generated content, small domain gaps still remain. After the segmentation model was trained, an inference run was performed on all the datasets, in order to obtain a processed version of the dataset with no background, to assess its influence on the classification performance. Classes ranging from 1 to 13 were collapsed into a single *foreground* class, while class 0 remained designated as the *background* class, effectively isolating the main subjects in the images and ensuring that the classifier performance could be evaluated without any bias introduced by the background information. Some of the results of the application of the background removal procedure to GAN–generated images are shown in Figure 4.2.

To evaluate the best segmentation result, both the model accuracy and its number of parameters were considered. Since both versions with the ResNet50 and MobileNetV3 Large backbones delivered comparable accuracy (0.9539 and 0.9484, respectively), the latter was preferred due to its much lower number of parameters (15,411,966 and 5,815,646, respectively). This preference becomes even more evident when considering the Xception backbone, which, despite achieving the best segmentation performance, requires a significantly higher number of parameters (74,803,174). The lightweight nature of MobileNetV3 makes it particularly suitable for deployment in resource–constrained environments, such as mobile devices or embedded systems,

Table 4.1: DeepLabV3+ performance for the three different segmentation models. The feature extraction backbone architecture used is reported together with the total number of trainable parameters and the pixel–pixel accuracy of the trained model on the test section of the complete Mut1ny dataset.

Backbone	Number of Parameters	Accuracy
Xception	74,803,174	**0.9601**
ResNet50	15,411,966	0.9539
MobileNetV3 Large	**5,815,646**	0.9484

Table 4.2: Fake detection performance metrics for different generators, obtained with a MobileNetV3 Large backbone. Notice how the background plays a significant role in StyleGAN2–generated images, aiding, with its presence, the decision process. This is not true for StyleGAN3.

Generator	Background	Accuracy	Precision	Recall	F1-Score	MCC
StyleGAN2	Yes	**0.9495**	**0.9562**	**0.9687**	**0.9624**	**0.8860**
	No	0.9022	0.9232	0.9307	0.9270	0.7790
StyleGAN3	Yes	0.8373	0.8314	0.6327	0.7186	0.6118
	No	**0.8499**	**0.8448**	**0.6648**	**0.7441**	**0.6492**

without significantly compromising segmentation accuracy. On the other hand, Xception higher parameter count can be beneficial for applications requiring high–accuracy, but limits its usability in real–time or battery–sensitive scenarios. Table 4.1 displays the segmentation performance metrics.

4.2 Fake detection and transfer learning methods

While the current experiments were designed to isolate the influence of the background by removing it, an inverse setting — in which the foreground is masked out and only the background is preserved — could represent a compelling future direction. Such an approach would allow for a more precise quantification of how much discriminative power is retained in background regions alone, and to what extent inconsistencies in the surrounding environment contribute to fake detection, independently of facial features.

Table 4.2 displays the performance metrics for the classification tasks. These results suggest that, for the detection of StyleGAN2–generated images, the background plays a crucial role, as its removal leads to a significant reduction in performance, confirming hypothesis H1. However, this is not true with StyleGAN3, since, as Table 4.2 shows, the background seems to be a distracting factor for the model, weakening its ability to detect images obtained with this generator, and this is also highlighted by the drastic reduction in performance across all the metrics considered with respect to the first two experiments (this is likely also due to the much smaller number of generated samples gathered for StyleGAN3).

Regarding the transfer learning strategies, the results highlight the strong correlation between

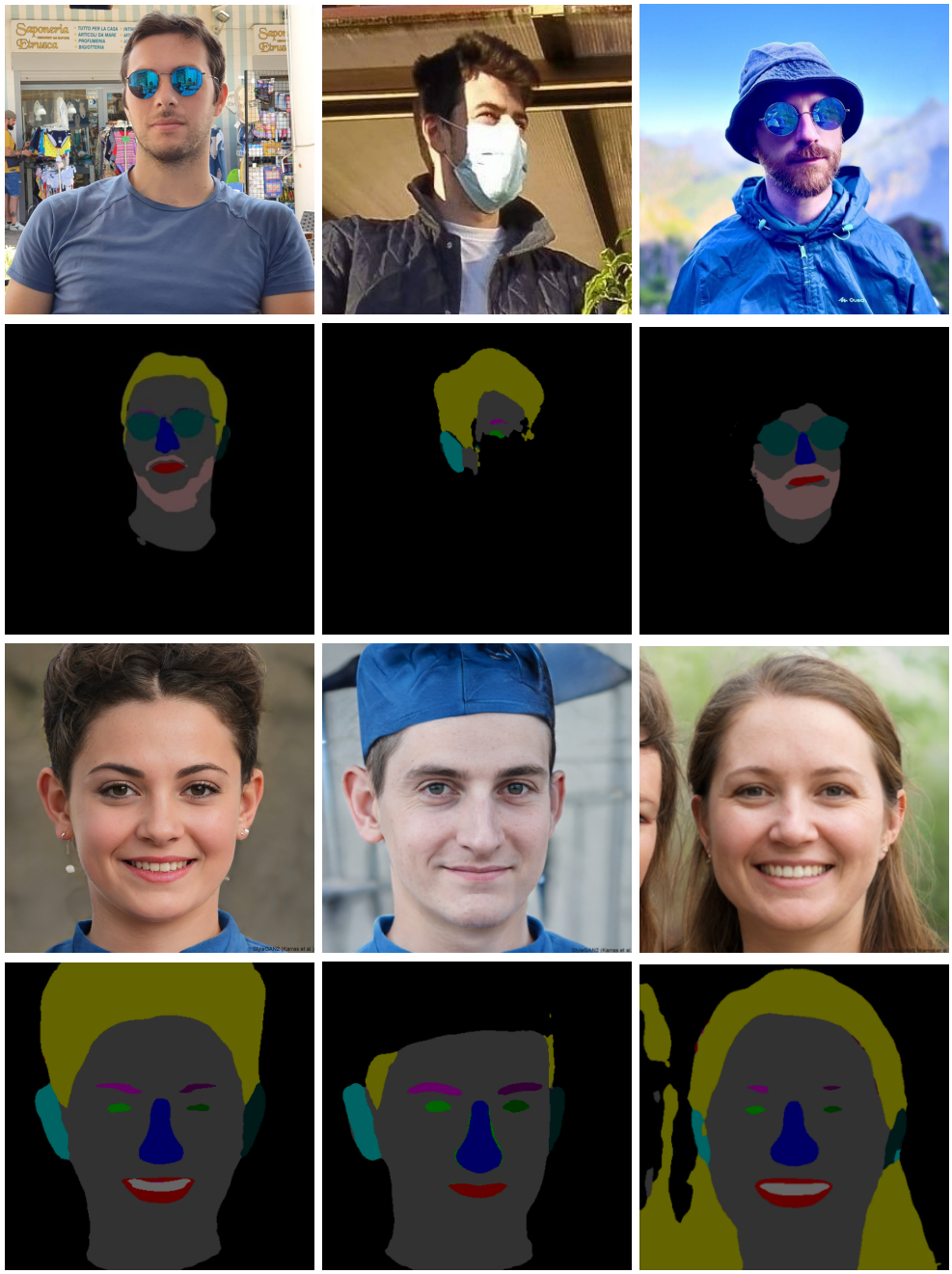

Figure 4.1: Comparison of original and segmented images, with segmentation masks coloured as per Mut1ny colormap, shown in Figure 3.1. The first two rows collect real images — of three of the authors in [65] and the result of their segmentations —, while the last two illustrate StyleGAN2–generated images and their segmentations.

Figure 4.2: Application of the background removal procedure to some StyleGAN2–generated images. The segmentation process provides finely cropped facial images, isolating the main subjects and blackening the background. The same approach is followed for both the generated and the real samples of the dataset, providing alternative and separate versions of the training samples to ensure that the model is only focusing on faces without the help of the background.

the segmentation pre–task and the final classification task. Indeed, Table 4.3 confirms the significant advantages they offered in improving the performance of the classifier for the fake detection task, verifying that pre–training the network on a segmentation task allowed the model to leverage meaningful spatial representations of facial features, enhancing its ability to distinguish between real and StyleGAN2–generated images.

As shown in Table 4.3, the *All* strategy — i.e. the one where all parameters of the model are set as retrainable, using the pre–trained weights as initialization — consistently outperformed the *Last* — i.e. the strategy where only the final layers were retrained — across all metrics, except for the precision in the case of MobileNetV3 backbone. Moreover, both of the strategies improved the results obtained with no transfer learning applied, demonstrating the importance of fine–tuning the entire network to adapt the pre–trained features to the specific task, and proving hypothesis H2. In general, the MobileNetV3 Large backbone showed superior performance compared to alternative architectures such as ResNet50, especially when transfer learning was applied. Despite being apparently surprising, this outcome could be attributed to better model regularization, more efficient feature extraction, and initial feature representation more aligned with the dataset. Additionally, the smaller number of parameters in MobileNetV3 might have reduced the risk of overfitting compared to ResNet50, which, despite being a more powerful

model, may have learned less generalizable patterns.

In particular, the transfer learning strategy had a remarkable effect for both the backbone models over the non–transfer learning cases, with an improvement in performance of about 8% for the MobileNet submodule and of about 7% in the ResNet submodule, when the pre–trained models are used only as parameters initializers. That is, starting from a favorable situation in the parameters initialization, the models can adapt the hidden representation of the samples to the task they are trained for. This result is in accordance with known literature [6], that highlights how large models, as the ones used in the study as segmentation submodule, benefit from transfer learning. With respect to the final classification performance, the results are in line with the literature regarding StyleGAN2–generated images detection, as reported in [113], even considering the heterogeneity of approaches and evaluation methods there reported. However, it is worth noting that, to the author's knowledge, using transfer learning from segmentation models to the detection of generated images is a novel approach, which could possibly improve current methodologies, so as to achieve increasingly high detection. Additionally, the original Xception backbone featured in the DeepLabV3+ architecture has been tested. Although this provided slightly better segmentation performance, this improvement came at the cost of about 12 times the number of parameters (∼75M *vs.* ∼6M with the MobileNet backbone), thus greatly extending training time and reducing practical feasibility for many applications.

As Table 4.4 shows, another important aspect is that all the transfer learning approaches also led to faster convergence during training, reducing the training time with *Last* and *All* to approximately 80% and 50% of the time needed for an epoch in the no transfer learning scenario, respectively, as pre–trained weights provided the model with a strong starting point. A possible explanation is that with the pre–training procedure, the model learned deep representations of facial features such as eyes, nose, and mouth, which are critical for detecting inconsistencies introduced by GANs.

4.3 SHAP analysis

Finally, SHAP was employed to enhance the interpretability of the model decisions by providing a quantitative breakdown of each feature contribution to the prediction. SHAP treats coalitions of pixels as features, and the higher the modulus of the Shapley value assigned to each coalition, the more impactful that coalition was for the final decision, while its sign indicates the direction in which the feature guided the outcome.

As illustrated in Figure 4.3, SHAP visually highlights the role of the different areas of the images taken as examples, with areas providing higher Shapley values (red) being the pixel coalitions which confirmed the decision made by the model; the more intense the color (i.e., the less transparent it appears), the higher the Shapley value associated with that coalition and its absolute contribution. Conversely, lower Shapley values (blue) indicated that the corresponding areas were leading the model towards the choice opposite to the final decision taken for that image. As its high Shapley values indicates, the most important area for the first picture in Figure 4.3 turned out to be the one at the bottom left, which includes a large portion of the background, further validating H1. On the other hand, in the last picture of Figure 4.3 the model focused much more in the central portion of the face, specifically on the mouth–nose–eyes region. To better assess the relevance of the background region in the images, SHAP analysis

Table 4.3: Fake detection performance on StyleGAN2–generated images in the test set, obtained with the MobileNetV3 Large backbone. The column TL refers to the transfer learning strategy used in the learning procedure: "—" means that no transfer learning is applied; Last means that only the backbone feature extractor and final layers are trainable, while the rest of the sub–model is set as non–trainable; finally, All means that all the model parameters are re–trainable, using the pre–trained parameters only for the initialization of the whole model. Both transfer learning strategies improve the performance of the classifier, for both the feature extractor backbones, showing the feasibility of the approach.

Backbone	TL	Accuracy	Precision	Recall	F1
Xception	—	0.9508	0.9748	0.9508	0.9627
	Last	0.9690	**0.9807**	0.9726	0.9766
	All	**0.9716**	0.9756	**0.9819**	**0.9787**
ResNet50	—	0.8704	0.8751	0.9397	0.9062
	Last	0.9176	0.9376	0.9387	0.9382
	All	**0.9342**	**0.9509**	**0.9504**	**0.9507**
MobileNetV3 Large	—	0.8690	0.9073	0.8950	0.9011
	Last	0.8840	**0.9842**	0.8395	0.9061
	All	**0.9504**	0.9623	**0.9634**	**0.9628**

Table 4.4: Training time and convergence epoch for different backbones and transfer learning strategies. For each entry, the total training time and the best epoch (convergence) are reported.

Backbone	No TL	Last	All
Xception	40h 34m (25)	30h 47m (17)	22h 03m (10)
ResNet50	2h 53m (15)	2h 17 (13)	1h 48 (12)
MobileNetV3 Large	**2h 31m (15)**	**1h 59m (11)**	**1h 15m (11)**

has also been executed on the two different versions of some of the images obtained before and after the background removal process. Some of the comparisons are shown in Figure 4.4 In many of the cases under analysis, SHAP highlighted the critical role of background elements in determining whether some of the images were classified as real or generated; thus, hypothesis H3 can be considered verified.

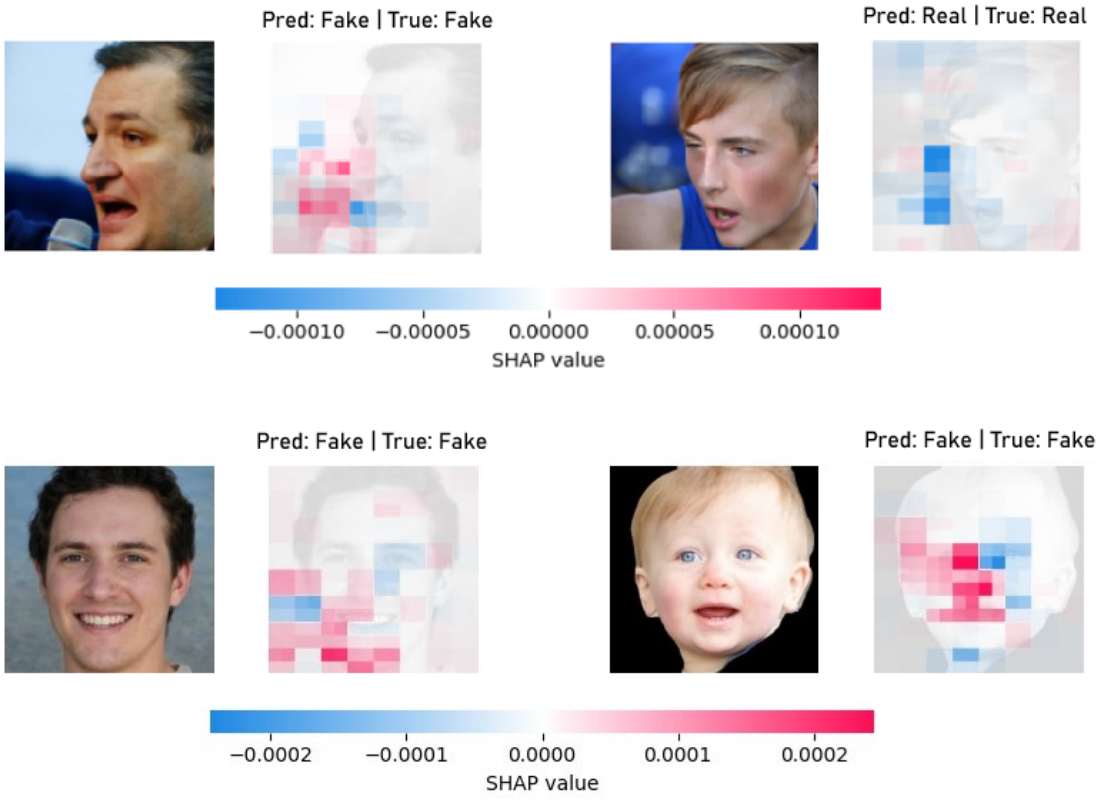

Figure 4.3: Application of SHAP to four test images. For each sample, the input image and the corresponding SHAP values for the coalitions of pixels are reported. Positive SHAP values (red coalitions) indicate that the group of pixels contributes positively toward the model prediction, while negative SHAP values (blue coalitions), indicate a negative contribution. In many images, the background plays a crucial role in the decision, both in negative and positive ways (e.g., in the top right and bottom left figures, respectively), thus validating the initial claim and the classification results.

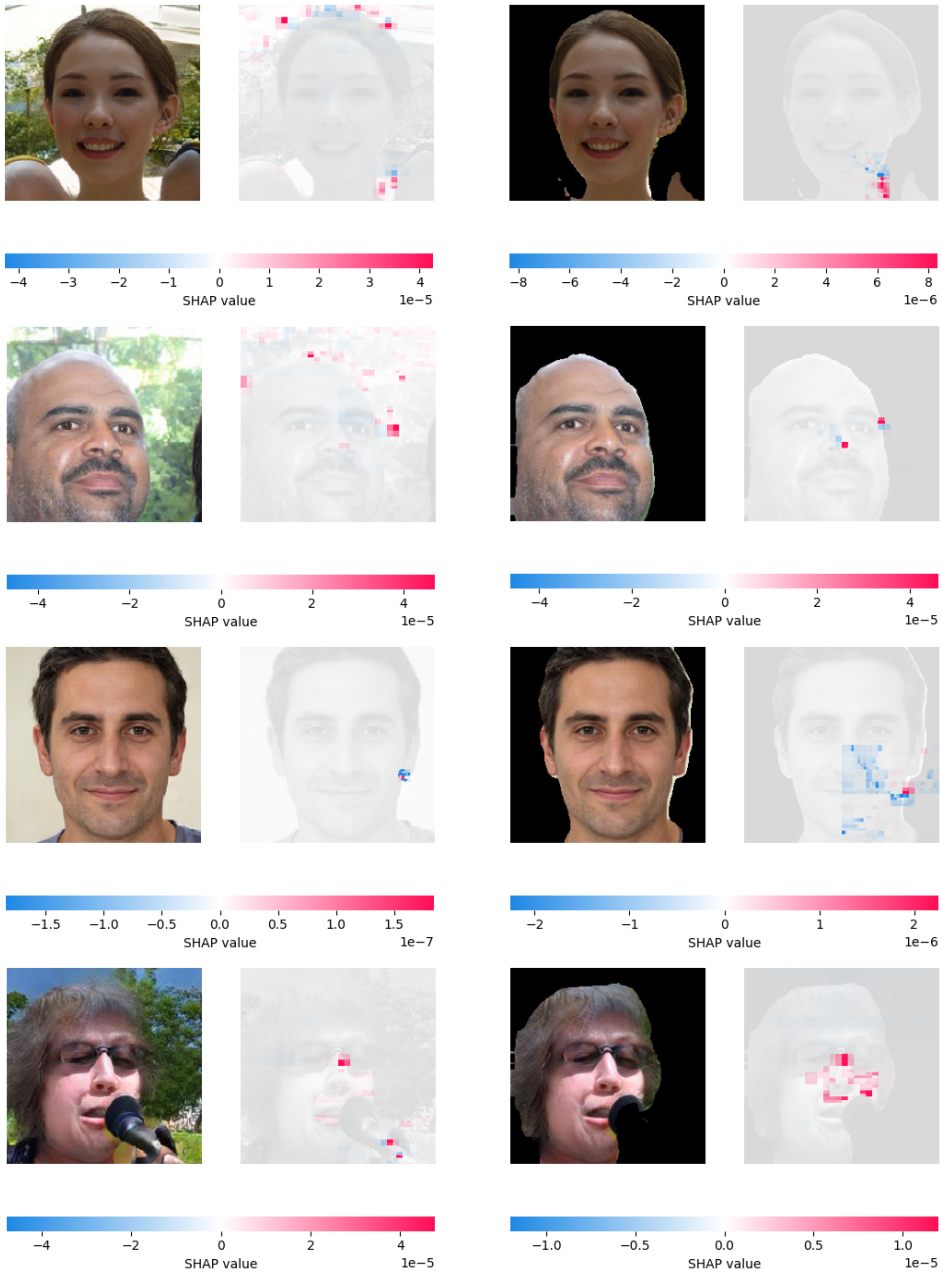

Figure 4.4: SHAP analysis results before (on the left) and after (on the right) background removal. Notice how, in the first two examples, the model strongly relies on the background for the classification, while the facial features are almost irrelevant to the decision. In the third example, as the background in the original image is already uniform and monochromatic, the model naturally focuses on the facial (ear) region. However, background removal further aids the decision mechanism by making the model concentrate on additional facial features. Finally, in the last example, a noticeable visual artifact is present in the lower right corner of the image, which the model also focuses on. Removing this artifact compels the model to analyze facial features more central to the face.

Part IV

Conclusion

Chapter 5

Conclusions and future work

5.1 Summary of findings

This thesis presents a novel approach for detecting AI–generated human faces, with particular focus on the importance of the background for the detection. The architecture is mainly based on a state–of–the–art semantic segmentation model (DeepLabV3+), modified with the introduction of a MobileNetV3 Large or ResNet50 backbone in place of the original Xception network for efficient feature extraction. A preliminary procedure is performed, by training the model on the Mut1ny facial segmentation dataset, to differentiate the various areas of the images.

After the first training is concluded, the background is removed from the images contained in another dataset, used for the fake detection task and consisting of both real and generated images of human subjects. This dataset is processed to isolate the areas of the images containing the faces from their background, providing an alternative version of it without the background. The detection procedure is then performed on both the versions of the dataset and the performance is compared, to evaluate the importance of the background for the classification.

A transfer learning approach is also introduced to enhance model performance. Specifically, two different transfer learning policies are explored, both leveraging the pre–training phase of the model on the facial segmentation task. The first policy involves fine–tuning only the final head of the pre–trained model, keeping the majority of the network frozen, to retain the general features learned during the segmentation phase, while the second one retrains the entire network, allowing all layers to adapt to the new classification task and using the weights of the segmentation task as its starting point.

Finally, an explainability tool called SHAP was also integrated into the framework to provide a better understanding of the model decision–making process. Specifically, the method highlights the areas of the image most relevant for the final classification, both for confirming the decision, and for deviating from it.

Experiments revealed that:

- The background plays a crucial role in improving the accuracy of fake image detection, particularly for StyleGAN2–generated images, where its removal leads to a significant drop in performance. Conversely, in StyleGAN3–generated images, the background appears to act as a distracting factor, weakening the model detection capabilities.

- Transfer learning, whether partial or full, significantly improved fake image detection accuracy compared to training from scratch. The full transfer learning approach, in particular, demonstrated better adaptability and performance by allowing the entire network to fine–tune itself for the classification task.

- The integration of SHAP provided interpretable insights into the classifier decisions, validating the model reliance on critical features both in the background and in the foreground and showing that the former was crucial for the decision since, when removed, the model changed its area of focus.

5.2 Contributions to the field

This research contributes to the field of AI–generated image detection by:

- introducing a hybrid model that combines semantic segmentation and classification for robust and interpretable generated content detection;
- demonstrating the effectiveness of transfer learning strategies in leveraging pre–trained segmentation models for classification tasks;
- providing empirical evidence on the role of background information in classification, particularly for the StyleGAN model family;
- offering insights into the interpretability of classification decisions using tools like SHAP, which can improve deep models' trustability.

5.3 Limitations of the study and prospects for future research

The study has certain limitations to take into account:

- the reliance on StyleGAN2 and StyleGAN3 for synthetic datasets limits the generalizability of the findings to other generative models;
- although the model used in this study demonstrated reasonable performance in fake detection, it was not originally designed for this specific purpose and was developed by the authors of [114] and [65] by modifying a semantic segmentation model; other state–of–the–art models, specifically tailored for fake detection, might exhibit different results compared to those observed here;
- a significant portion of the real dataset (FFHQ) overlaps with the training datasets of some generative models, introducing potential biases;
- the limited application of the interpretability tool, due to high computational costs, restricted the depth of insights into model decision–making.

The results and limitations of this study suggest several directions for future research.

- Expanding datasets to include a wider variety of AI–generated and real images from diverse generative models and environments, together with the enhancement of the generalizability of the proposed approach across different conditions, by exploring domain adaptation techniques.

- Comparing the performance of the proposed model with state–of–the–art architectures specifically designed for fake detection to evaluate differences both in performance and in the impact of the background removal procedure.
- Investigating the implementation of the most lightweight architecture here presented, i.e. the one with the MobileNetV3 backbone, in real–time applications on mobile or other resource–limited devices.
- Applying other explainabilty or interpretability tools, such as Grad–CAM, to further understand model decisions and improve training strategies, even by integrating these models into the training pipeline to guide it.
- While the current experiments were designed to isolate the influence of the background by removing it, an inverse setting—in which the foreground is masked out and only the background is preserved—could represent an interesting future direction, to allow for a more precise quantification of how much discriminative power derives from background regions alone, and to what extent inconsistencies in the surrounding environment contribute to fake detection, independently of facial features.
- Evaluating the impact of synthetic images with modified backgrounds, such as AI–generated faces superimposed on real backgrounds, to study their effects on detection accuracy.

Chapter 6

Other works

This chapter briefly presents other research contributions obtained during the doctoral period that fall outside the primary scope of the present work. These projects showcase the application of machine learning techniques in different domains, including the medical field—ranging from medical imaging to genetic studies on COVID-19 susceptibility—and industrial fault diagnosis through lightweight neural network models.

6.1 Hybrid deep learning model for liver tumor segmentation

The paper, entitled *A Hybrid Deep Learning Approach for Liver Tumor Segmentation Using DeepLabV3+ and Hidden Markov Models* [115], focuses on the semantic segmentation of liver tumors in contrast–enhanced Computed Tomography (CT) scans, with the objective of distinguishing between healthy liver tissues and tumour lesion areas, separating both from the background, in order to support diagnosis and treatment planning.

The proposed method also builds upon DeepLabV3+ with a MobileNetV3 Large backbone—the same segmentation architecture used for the main works that constitute this thesis—which was adapted and trained on the liver tumor task from the Medical Segmentation Decathlon (MSD). The dataset provided for this task consists of 3D CT volumes; therefore, the data are first converted by extracting 2D slices from each of them, while class imbalance is handled through frequency–based weighting.

To overcome limitations related to slice–by–slice prediction—such as spatial incoherence and the misclassification of small or low–contrast lesions—and to leverage the information conveyed by the spatial configuration of consecutive slices, a post–processing step based on Hidden Markov Models (HMMs) was introduced. Each pixel location across slices was treated as a temporal sequence of labels: a three–state ergodic HMM (background, liver, tumor) was trained using the Baum–Welch algorithm, and the most probable label sequences were inferred using the Viterbi algorithm, filtering out low–confidence predictions. Finally, a 3D median filter was applied to smooth the segmentation volume.

The effectiveness of this hybrid approach was validated by measuring the Intersection over Union (IoU) score before and after the refinement procedure. The results show a consistent improvement in segmentation performance, particularly in the tumor class, confirming the positive impact of the HMM–based approach and highlighting the potential of combining deep learning with probabilistic modeling in the field of medical image segmentation.

6.2 Fault diagnosis in roller bearings using MobileNet

In the paper titled *A MobileNet Neural Network Model for Fault Diagnosis in Roller Bearings* [116], the use of deep learning for predictive maintenance in an embedded system context is explored. In particular, a MobileNetV3 Small architecture was designed and deployed, to classify vibration signals from roller bearings affected by different types of mechanical defects, with the aim of implementing the resulting model on low–power microcontrollers.

The data used for training are obtained by collecting signals from three different accelerometers with diverse metrological properties. These signals are then emulated by a custom–built bearing fault emulator capable of reproducing vibration patterns typical of three common fault types: outer race, inner race, and ball defects. The raw time–domain signals were converted into 64×64 grayscale images through a lightweight preprocessing strategy, and the final result of this procedure composes the novel dataset used for training the model.

A transfer learning approach is applied by using the version of the DL model pretrained on ImageNet. Initially, only the final classification layers are retrained, followed by a fine–tuning phase, in which the entire network is unfrozen. The final model achieves an accuracy of 99.41% on the test set, which includes data collected at machine resonances different from those used during training, confirming the network generalization capabilities, at a remarkably low computational cost, making it suitable for deployment on low–power devices.

6.3 Genetic studies on host susceptibility to COVID–19

During the COVID–19 pandemic, a substantial research effort was dedicated to uncovering host genetic factors associated with SARS–CoV–2 infection and disease severity. Although this research activity began prior to the doctoral program, some of its outcomes were published in peer–reviewed medical journals during the PhD years. For this reason, and despite the shift in research focus, these contributions are briefly reported here.

The first of these, titled *SARS–CoV–2 susceptibility and COVID–19 disease severity are associated with genetic variants affecting gene expression in a variety of tissues* [117], explored associations between host genotypes and COVID–19 outcomes through transcriptome–wide association studies, identifying variants influencing gene expression in tissues such as lung and blood.

The second, *Mapping the human genetic architecture of COVID–19* [118], was a large–scale international effort within the COVID–19 Host Genetics Initiative. It aggregated data from dozens of cohorts worldwide to identify genomic loci associated with both susceptibility to infection and disease progression. The study revealed 13 genome–wide significant loci, highlighting mechanisms linked to immune response, lung function, and inflammatory pathways.

The third study, *Common, low–frequency, rare, and ultra–rare coding variants contribute to COVID–19 severity* [119], proposed an integrated polygenic modeling approach combining variants across the frequency spectrum to predict COVID–19 severity and demonstrating that severity can be partly explained by a broad spectrum of genetic variation, offering interpretable predictive models based on whole–exome data.

Finally, *Whole–genome sequencing reveals host factors underlying critical COVID–19* [120] employed whole–genome sequencing of critically ill patients to uncover 23 significant genetic

associations with severe COVID–19. The findings included novel loci related to interferon signaling, immune cell differentiation, and coagulation pathways.

Appendices

Appendix A

Code snippets

This appendix shows some of the code used to run the experiment here presented.

A.1 DeepLabV3+ function declaration

```
def DeeplabV3Plus_mobilenetv3(num_classes,
                                    filters_conv1=24, filters_conv2=24,
                                    filters_spp=128, filters_final=128,
                                    dilated_conv_rates=[1, 4, 8, 16],
                                    trainable_backbone=True,
                                    path_model_trained: str = None,
                                    transfer_learning: str = "last"):
    assert transfer_learning in ["last", "all"], "Transfer learning policy not
    supported"
    model_input = keras.Input(shape=(None, None, 3))
    preprocessed = keras.applications.mobilenet_v3.preprocess_input(model_input)
    mobilenetv3 = keras.applications.MobileNetV3Large(weights="imagenet",
    include_top=True, input_tensor=preprocessed)
    mobilenetv3.trainable = trainable_backbone

    x = mobilenetv3.get_layer("expanded_conv_12/project").output
    input_b = mobilenetv3.get_layer("expanded_conv_3/project").output

    x1 = layers.GlobalAveragePooling2D()(x)
    x1 = layers.Reshape((1, 1, x.shape[-1]))(x1)
    x1 = layers.Conv2D(filters=filters_conv1, kernel_size=1, padding="same")(x1)
    x1 = layers.BatchNormalization()(x1)
    x1 = DynamicUpsample()(x1, x)

    pyramids = []
    for rate in dilated_conv_rates:
        if rate == 1:
            pyramid = layers.Conv2D(filters=filters_spp, kernel_size=3,
    dilation_rate=rate, padding="same")(x)
            pyramid = layers.BatchNormalization()(pyramid)
            pyramids.append(pyramid)
        else:
            pyramid = layers.Conv2D(filters=filters_spp, kernel_size=3 + int(rate
     * (1 / 3)), dilation_rate=rate,
                                    padding="same")(x)
            pyramid = layers.BatchNormalization()(pyramid)
            pyramids.append(pyramid)

    x = layers.Concatenate(axis=-1)([x1] + pyramids)
```

```
    x = layers.Conv2D(filters=filters_spp, kernel_size=1, padding="same")(x)
    x = layers.BatchNormalization()(x)

    input_b = layers.Conv2D(filters=filters_conv2, kernel_size=1, padding="same")
    (input_b)
    input_b = layers.BatchNormalization()(input_b)

    input_a = DynamicUpsample()(x, input_b)

    x = layers.Concatenate(axis=-1)([input_a, input_b])
    x = layers.Conv2D(filters=filters_final, kernel_size=3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Conv2D(filters=filters_final, kernel_size=3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = DynamicUpsample()(x, model_input)

    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dense(200, activation="selu")(x)
    model_output = layers.Dense(num_classes, activation="sigmoid")(x)
    model = keras.Model(inputs=model_input, outputs=model_output)

    if path_model_trained is not None:
        model_from = tf.keras.models.load_model(path_model_trained,
                                                custom_objects={'DynamicUpsample'
    : DynamicUpsample})
        for l, l_old in zip(model.layers[:-3], model_from.layers):
            try:
                l.set_weights(l_old.get_weights())
            except Exception as e:
                print("Error in Layer: ", l.name, l.trainable, "new", [i.shape
    for i in l.get_weights()], "old",
                      [i.shape for i in l_old.get_weights()], e)
            if transfer_learning == "last": l.trainable = False
    mobilenetv3.trainable = trainable_backbone

    return model
```

Listing A.1: Declaration of the DeepLabV3+ model function with MobileNetV3 backbone, supporting adjustable parameters for transfer learning, feature extraction, and dilated convolutions.

A.2 Dynamic Upsample Class

```
class DynamicUpsample(tf.keras.layers.Layer):
    def __init__(self, method='bilinear', **kwargs):
        super().__init__(**kwargs)
        self.method = method

    def call(self, inputs, ref_tensor):
        return tf.image.resize(inputs, (tf.shape(ref_tensor)[1], tf.shape(
    ref_tensor)[2]), method=self.method)
```

Listing A.2: Implementation of the DynamicUpsample class, a custom TensorFlow layer designed to dynamically resize tensors to match the spatial dimensions of a reference tensor using the specified interpolation method.

Bibliography

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[2] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988. [Online]. Available: https://doi.org/10.1007/BF00133570

[4] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980. [Online]. Available: https://doi.org/10.1007/BF00344251

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: https://ieeexplore.ieee.org/document/726791

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[7] K. He, X. Zhang, S. Ren, and J. Sun, " Deep Residual Learning for Image Recognition ," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2016, pp. 770–778. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.90

[8] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013. [Online]. Available: http://arxiv.org/abs/1311.2901

[9] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *CoRR*, vol. abs/1506.06579, 2015. [Online]. Available: http://arxiv.org/abs/1506.06579

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," 2015. [Online]. Available: https://arxiv.org/abs/1412.0767

[11] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *CoRR*, vol. abs/1412.2306, 2014. [Online]. Available: http://arxiv.org/abs/1412.2306

[12] B. Zhao, J. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[13] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[14] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014, pp. 2672–2680.

[16] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410.

[17] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253520303110

[18] B. Chesney and D. Citron, "Deep fakes: A looming challenge for privacy, democracy, and national security," *California Law Review*, vol. 107, no. 6, pp. pp. 1753–1820, 2019. [Online]. Available: https://www.jstor.org/stable/26891938

[19] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," 2018. [Online]. Available: https://arxiv.org/abs/1812.08685

[20] J. Wang, B. Tondi, and M. Barni, "An eyes-based siamese neural network for the detection of gan-generated face images," *Frontiers in Signal Processing*, vol. 2, 2022. [Online]. Available: https://www.frontiersin.org/journals/signal-processing/articles/10.3389/frsip.2022.918725

[21] T. Fu, M. Xia, and G. Yang, "Detecting gan-generated face images via hybrid texture and sensor noise based features," *Multimedia Tools and Applications*, vol. 81, no. 18, pp. 26 345–26 359, 2022. [Online]. Available: https://doi.org/10.1007/s11042-022-12661-1

[22] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 833–851.

[23] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[24] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological Cybernetics*, vol. 20, no. 3, pp. 121–136, 1975. [Online]. Available: https://doi.org/10.1007/BF00342633

[25] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[26] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *CoRR*, vol. abs/1710.05941, 2017. [Online]. Available: http://arxiv.org/abs/1710.05941

[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, no. 3, 2013. [Online]. Available: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, " You Only Look Once: Unified, Real-Time Object Detection ," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2016, pp. 779–788. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[31] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[32] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[33] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 194–281. [Online]. Available: https://direct.mit.edu/books/monograph/4424/chapter/189414/Information-Processing-in-Dynamical-Systems

[34] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, D. van Dyk and M. Welling, Eds., vol. 5. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 16–18 Apr 2009, pp. 448–455. [Online]. Available: https://proceedings.mlr.press/v5/salakhutdinov09a.html

[35] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations (ICLR)*, 2014. [Online]. Available: https://arxiv.org/abs/1312.6114

[36] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2256–2265. [Online]. Available: https://proceedings.mlr.press/v37/sohl-dickstein15.html

[37] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[38] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on*

Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 297–304. [Online]. Available: https://proceedings.mlr.press/v9/gutmann10a.html

[39] G. Iglesias, E. Talavera, and A. Díaz-Álvarez, "A survey on gans for computer vision: Recent research, analysis and taxonomy," *Computer Science Review*, vol. 48, p. 100553, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013723000205

[40] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 214–223. [Online]. Available: https://proceedings.mlr.press/v70/arjovsky17a.html

[41] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:11758569

[42] D. Bang and H. Shim, "Mggan: Solving mode collapse using manifold-guided training," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 2347–2356.

[43] W. Li and Y. Tang, "Soft generative adversarial network: Combating mode collapse in generative adversarial network training via dynamic borderline softening mechanism," *Applied Sciences*, vol. 14, no. 2, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/2/579

[44] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.

[45] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," 2019. [Online]. Available: https://arxiv.org/abs/1809.11096

[46] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7354–7363. [Online]. Available: https://proceedings.mlr.press/v97/zhang19d.html

[47] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6629–6640.

[48] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251.

[49] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, " StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation ," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 8789–8797. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00916

[50] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[51] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 852–863. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/076ccd93ad68be51f23707988e934906-Paper.pdf

[52] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[53] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2018. [Online]. Available: https://arxiv.org/abs/1711.04340

[54] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218310749

[55] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1060–1069. [Online]. Available: https://proceedings.mlr.press/v48/reed16.html

[56] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: A survey toward private and secure applications," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3459992

[57] I. K. Dutta, B. Ghosh, A. Carlson, M. Totaro, and M. Bayoumi, "Generative adversarial networks in security: A survey," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2020, pp. 0399–0405.

[58] A. Birhane and V. U. Prabhu, " Large image datasets: A pyrrhic win for computer vision? ," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Los Alamitos, CA, USA: IEEE Computer Society, Jan. 2021, pp. 1536–1546. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/WACV48630.2021.00158

[59] A. Paullada, I. D. Raji, E. M. Bender, E. Denton, and A. Hanna, "Data and its (dis)contents: A survey of dataset development and use in machine learning research," *Patterns*, vol. 2, no. 11, p. 100336, Nov 12 2021.

[60] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 610–623. [Online]. Available: https://doi.org/10.1145/3442188.3445922

[61] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 09, pp. 13 693–13 696, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7123

[62] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[63] A. Farahani, B. Pourshojae, K. Rasheed, and H. R. Arabnia, "A concise review of transfer learning," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2020, pp. 344–351.

[64] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[65] M. Tanfoni, E. G. Ceroni, N. Pancino, M. Bianchini, and M. Maggini, "Facial segmentation in deepfake classification: a transfer learning approach," *Procedia Computer Science*, vol. 246, pp. 4160–4168, 2024, 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050924022749

[66] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/1/18

[67] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, Aug. 2018. [Online]. Available: https://doi.org/10.1145/3236009

[68] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[69] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 2018, pp. 80–89.

[70] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Inf. Fusion*, vol. 58, pp. 82–115, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:204824113

[71] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: Interpreting, explaining and visualizing deep learning.* Springer Nature, 2019, vol. 11700.

[72] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/8/832

[73] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[74] L. S. Shapley, "17. a value for n-person games," in *Contributions to the Theory of Games (AM-28), Volume II*, H. W. Kuhn and A. W. Tucker, Eds. Princeton: Princeton University Press, 1953, pp. 307–318. [Online]. Available: https://doi.org/10.1515/9781400881970-018

[75] D. Schmeidler, "The nucleolus of a characteristic function game," *SIAM Journal on Applied Mathematics*, vol. 17, no. 6, pp. 1163–1170, 1969. [Online]. Available: http://www.jstor.org/stable/2099196

[76] L. S. Shapley and M. Shubik, "A method for evaluating the distribution of power in a committee system," *American Political Science Review*, vol. 48, no. 3, p. 787–792, 1954.

[77] A. E. Roth, "Introduction to the shapley value," in *The Shapley Value: Essays in Honor of Lloyd S. Shapley*, A. E. Roth, Ed. Cambridge University Press, 1988, p. 1–28.

[78] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[79] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, " MobileNetV2: Inverted Residuals and Linear Bottlenecks ," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 4510–4520. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474

[80] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[81] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[82] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[83] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001.

[84] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, pp. 647–665, 2014.

[85] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 598–617.

[86] X. Wang, H. Guo, S. Hu, M.-C. Chang, and S. Lyu, "Gan-generated faces detection: A survey and new perspectives," *ECAI 2023*, pp. 2533–2542, 2023.

[87] T.-N. Le, H. H. Nguyen, J. Yamagishi, and I. Echizen, "Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 097–10 107.

[88] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015, pp. 1–14.

[89] Q. Gu, S. Chen, T. Yao, Y. Chen, S. Ding, and R. Yi, "Exploiting fine-grained face forgery clues via progressive enhancement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 735–743.

[90] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *European conference on computer vision*. Springer, 2020, pp. 86–103.

[91] K. Songsri-in and S. Zafeiriou, "Complement face forensic detection and localization with facial-landmarks," 2019.

[92] A. V. Nadimpalli and A. Rattani, "Facial forgery-based deepfake detection using fine-grained features," in *2023 International Conference on Machine Learning and Applications (ICMLA)*, 2023, pp. 2174–2181.

[93] Z. Liu, X. Qi, and P. H. Torr, "Global texture enhancement for fake face detection in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8060–8069.

[94] B. Chen, X. Liu, Y. Zheng, G. Zhao, and Y.-Q. Shi, "A robust gan-generated face detection method based on dual-color spaces and an improved xception," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, pp. 1–1, 09 2021.

[95] R. Wang, F. Juefei-Xu, L. Ma, X. Xie, Y. Huang, J. Wang, and Y. Liu, "Fakespotter: a simple yet robust baseline for spotting ai-synthesized fake faces," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, ser. IJCAI'20, 2021.

[96] B. Liang, Z. Wang, B. Huang, Q. Zou, Q. Wang, and J. Liang, "Depth map guided triplet network for deepfake face detection," *Neural Networks*, vol. 159, pp. 34–42, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608022004725

[97] Mut1ny, "Face/head segmentation dataset commercial purpose edition," 3 2024. [Online]. Available: https://store.mut1ny.com/product/face-head-segmentation-dataset-pro?v=cd32106bcb6d

[98] A. Hassani, Z. E. Shair, R. Ud Duala Refat, and H. Malik, "Distilling facial knowledge with teacher-tasks: Semantic-segmentation-features for pose-invariant face-recognition," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 741–745.

[99] M. Reimann, M. Klingbeil, S. Pasewaldt, A. Semmo, M. Trapp, and J. Döllner, "Locally controllable neural style transfer on mobile devices," *The Visual Computer*, vol. 35, no. 11, pp. 1531–1547, 2019. [Online]. Available: https://doi.org/10.1007/s00371-019-01654-1

[100] E. Khoshnevisan, H. Hassanpour, and M. M. AlyanNezhadi, "Face recognition based on general structure and angular face elements," *Multimedia Tools and Applications*, 2024. [Online]. Available: https://doi.org/10.1007/s11042-024-18897-3

[101] Mut1ny, "Face/head segmentation dataset community edition," 3 2024. [Online]. Available: https://store.mut1ny.com/product/face-head-segmentation-dataset-community-edition?v=cd32106bcb6d

[102] M. A. Rahman, B. Paul, N. H. Sarker, Z. I. A. Hakim, and S. A. Fattah, "Artifact: A large-scale dataset with artificial and factual images for generalizable and robust synthetic image detection," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 2200–2204.

[103] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-gan: Training gans with diffusion," 2023.

[104] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[105] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," 2020.

[106] W. Xia, Y. Yang, J.-H. Xue, and B. Wu, "Tedigan: Text-guided diverse face image generation and manipulation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2256–2265.

[107] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[108] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[109] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[110] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for mobilenetv3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[111] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[112] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[113] X. Wang, H. Guo, S. Hu, M.-C. Chang, and S. Lyu, "Gan-generated faces detection: A survey and new perspectives," *ECAI 2023*, pp. 2533–2542, 2023.

[114] M. Tanfoni, E. G. Ceroni, S. Marziali, N. Pancino, M. Maggini, and M. Bianchini, "Generated or not generated (gng): The importance of background in the detection of fake images," *Electronics*, vol. 13, no. 16, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/16/3161

[115] M. Tanfoni, E. G. Ceroni, M. Maggini, N. Pancino, and M. Bianchini, "A hybrid deep learning approach for liver tumor segmentation using deeplabv3+ and hidden markov models," in *2024 IEEE International Symposium on Systems Engineering (ISSE)*, 2024, pp. 1–5.

[116] E. Landi, F. Spinelli, M. Intravaia, M. Mugnaini, A. Fort, M. Bianchini, B. T. Corradini, F. Scarselli, and M. Tanfoni, "A mobilenet neural network model for fault diagnosis in roller bearings," in *2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2023, pp. 01–06.

[117] M. D'Antonio, J. P. Nguyen, T. D. Arthur, H. Matsui, A. D'Antonio-Chronowska, K. A. Frazer, and C.-. H. G. Initiative, "Sars-cov-2 susceptibility and covid-19 disease severity are associated with genetic variants affecting gene expression in a variety of tissues," *Cell Reports*, vol. 37, no. 7, p. 110020, 2021, epub 2021 Nov 3.

[118] M. E. K. Niemi, J. Karjalainen, R. G. Liao, B. M. Neale, M. Daly, A. Ganna, G. A. Pathak, S. J. Andrews, M. Kanai, K. Veerapen, I. Fernandez-Cadenas, E. C. Schulte, P. Striano, M. Marttila, C. Minica, E. Marouli, M. A. Karim, F. R. Wendt, J. Savage, L. Sloofman, G. Butler-Laporte, H.-N. Kim, S. Kanoni, Y. Okada, J. Byun, Y. Han, M. J. Uddin, G. D. Smith, C. J. Willer, J. D. Buxbaum, J. Mehtonen, H. Finucane, M. Cordioli, A. R. Martin, W. Zhou, B. Pasaniuc, H. Julienne, H. Aschard, H. Shi, L. Yengo, R. Polimanti, M. Ghoussaini, J. Schwartzentruber, I. Dunham, and C.-. H. G. Initiative, "Mapping the human genetic architecture of covid-19," *Nature*, vol. 600, no. 7889, pp. 472–477, 2021. [Online]. Available: https://doi.org/10.1038/s41586-021-03767-x

[119] C. Fallerini, N. Picchiotti, M. Baldassarri, K. Zguro, S. Daga, F. Fava, E. Benetti, S. Amitrano, M. Bruttini, M. Palmieri, S. Croci, M. Lista, G. Beligni, F. Valentino, I. Meloni, M. Tanfoni, F. Minnai, F. Colombo, E. Cabri, M. Fratelli, C. Gabbi, S. Mantovani, E. Frullanti, M. Gori, F. P. Crawley, G. Butler-Laporte, B. Richards, H. Zeberg, M. Lipcsey, M. Hultström, K. U. Ludwig, E. C. Schulte, E. Pairo-Castineira, J. K. Baillie, A. Schmidt, R. Frithiof, S. Furini, F. Montagnani, M. Tumbarello *et al.*, "Common, low-frequency, rare, and ultra-rare coding variants contribute to covid-19 severity," *Human Genetics*, vol. 141, no. 1, pp. 147–173, 2022. [Online]. Available: https://doi.org/10.1007/s00439-021-02397-7

[120] A. Kousathanas, E. Pairo-Castineira, K. Rawlik, A. Stuckey, C. A. Odhams, S. Walker, C. D. Russell, T. Malinauskas, Y. Wu, J. Millar, X. Shen, K. S. Elliott, F. Griffiths, W. Oosthuyzen, K. Morrice, S. Keating, B. Wang, D. Rhodes, L. Klaric, M. Zechner, N. Parkinson, A. Siddiq, P. Goddard, S. Donovan, D. Maslove, A. Nichol, M. G. Semple, T. Zainy, F. Maleady-Crowe, L. Todd, S. Salehi, J. Knight, G. Elgar, G. Chan, P. Arumugam, C. Patch, A. Rendon, D. Bentley, C. Kingsley, J. A. Kosmicki, J. E. Horowitz, A. Baras, G. R. Abecasis, M. A. R. Ferreira, A. Justice, T. Mirshahi, M. Oetjens, D. J. Rader, M. D.

Ritchie, A. Verma, T. A. Fowler, M. Shankar-Hari, C. Summers, C. Hinds, P. Horby, L. Ling, D. McAuley, H. Montgomery, P. J. M. Openshaw, P. Elliott, T. Walsh, A. Tenesa, J. K. Baillie, G. investigators, and C.-. H. G. Initiative, "Whole-genome sequencing reveals host factors underlying critical covid-19," *Nature*, vol. 607, no. 7917, pp. 97–103, 2022. [Online]. Available: https://doi.org/10.1038/s41586-022-04576-6