ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Εξάλειψη της προκατάληψης σε μοντέλα Τεχνητής Νοημοσύνης

Παναγιώτης Βλαχαΐτης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κωνσταντίνος Κολομβάτσος
Αναπληρωτής καθηγητής

Λαμία 4 Σεπτεμβρίου έτος 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Εξάλειψη της προκατάληψης σε μοντέλα Τεχνητής Νοημοσύνης

Παναγιώτης Βλαχαΐτης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κωνσταντίνος Κολομβάτσος
Αναπληρωτής καθηγητής

Λαμία 4 Σεπτεμβρίου έτος 2025

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Mitigation of bias in artificial intelligence models

Panagiotis Vlachaitis

FINAL THESIS

ADVISOR

Kolomvatsos Konstantinos
Associate Professor

Lamia September 4th year 2025

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί ***χωρίς να τα περικλείω σε εισαγωγικά*** *και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

2. Δέχομαι ότι η αυτολεξεί ***παράθεση χωρίς εισαγωγικά****, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 04/09/2025

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ABSTRACT

This thesis tackles the challenge of mitigating skin tone bias in deep learning models for skin cancer classification through the application and extension of Latent Adversarial Debiasing (LAD). While deep neural networks have demonstrated high performance in various tasks, they often adopt present biases inside the data unintentionally. To address this, we adapt LAD, an adversarial based technique for mitigating biases and evaluate its effectiveness across multiple architectures, including ResNet, DenseNet, and EfficientNet, under fully biased and semi-biased conditions. The study compares the models on a combination of the ISIC2020 and the PAD-UEFS, in addition to evaluating fairness metrics such as Equal Opportunity (EO), Disparate Impact (DI), and Accuracy Parity (AP) alongside traditional performance measures. Results show that LAD-enhanced models achieve significant improvements in fairness, with DenseNet121 and EfficientNetB0 in particular demonstrating robust performance across both fairness and accuracy. Moreover, the analysis highlights the trade-off between fairness and predictive accuracy, indicating that the model is sensitive to hyperparameters and extensive searches should be considered when considering implementing it. This work contributes insights into fairness-aware model design and provides a concrete foundation for deploying debiased deep learning models in meaningful real-world applications such as medical imaging. The implementation code is publicly available on the following github repository https://github.com/Panaghs01/Thesis.

Table of Contents

Table of Figures

1 Introduction

Artificial Intelligence (AI) has seen significant advancements in recent years with the introduction of Machine Learning (ML) and later, Deep Learning (DL). These approaches have demonstrated exceptional performance in various complex tasks such as image recognition, decision making and natural language processing. These technologies are applied in various fields of science in order to aid other researchers in their work by providing powerful computing structures and problem-solving methodologies. Healthcare in particular, has been frequently utilizing AI to assist clinicians with their work, providing second opinions, improving diagnosis outcomes and planning treatments just to name a few. One of the most popular challenges in healthcare ML is oncology, and specifically cancer diagnosis and treatment. Cancer has been the leading cause of death worldwide, accumulating nearly 10 million deaths in 2020, and based on 2019-2023 statistics there are approximately 145.4 deaths per 100.000 cancer cases [1], [2]. Melanoma, a form of skin cancer, is one of the most common types of cancer and it represents a major global health concern. When diagnosed in early stages, melanoma has a five-year survival rate of more than 99%, falling to 75% when the cancer reaches the lymph nodes and then even further at 35% when the cancer reaches distant organs [3].

Despite recent successes, AI systems are not immune to the biases present in the data on which they are trained. Systemic underperformance and unequal outcomes on certain minority class data can often go unnoticed if not taken fully into account. In medical contexts, such biases can have serious consequences, potentially leading to unequal performance across demographic groups and reinforcing existing healthcare disparities. For example, in dermatology, skin lesion classification models often underperform for patients with underrepresented skin tones, due to imbalanced datasets and confounding correlations in the training data. This raises critical concerns about fairness, and the ethical deployment of AI in clinical practice.

Recent research has highlighted the need for fairness-aware AI systems that not only achieve high accuracy but also maintain equitable performance across diverse populations. Bias in AI can arise from multiple sources, including data collection processes, annotation practices, and model architectures that inadvertently exploit spurious correlations. Recent research in 2024 compared 54 publicly available skin cancer datasets and concluded that only 3 of them included a skin tone attribute inside the dataset [4]. In the context of skin lesion detection, collider bias where non-causal features can become predictive due to dataset composition can cause models to rely on irrelevant cues like skin tone rather than the lesion itself.

This study addresses the challenge of bias mitigation in deep learning models for skin lesion detection. Building upon the Latent Adversarial Debiasing (LAD) framework, we aim to deploy the LAD model on a real world application and evaluate its effectiveness to separate causal signals from confounding factors on skin lesion image data. This approach is suited to medical imaging scenarios where perfectly balanced datasets are impractical to obtain. By integrating adversarial perturbations in the latent space, the method aims to produce

debiased training data that improves fairness metrics such as Equal Opportunity, Disparate Impact, and Accuracy Parity, without significantly compromising classification accuracy. To evaluate the proposed system, we combine two publicly available datasets, the ISIC2020 and PAD-UFES, creating a more demographically diverse benchmark for skin lesion classification. We conduct experiments across multiple CNN architectures, including ResNet, DenseNet, and EfficientNet, under both fully biased and semi-biased conditions.

This study, situated at the core of AI fairness, medical imaging, and bias mitigation methodologies, contributes to the expanding field of equitable healthcare AI. The outcomes emphasize the necessity of embedding fairness considerations into every stage of clinical AI development, from design and training to evaluation, making efforts to support the creation of diagnostic systems that are both dependable and fair in advancing cancer care.

In the following chapters we discuss the following matters: Chapter 2 sets up basic Artificial Intelligence, Machine Learning and Deep Learning principles and structures, describes model architecture and training procedure in detail and listing some of the most significant architectures proposed in the recent years that were also important for the scopes of this study, as well as challenges and limitations of AI. In chapter 3 we explore how AI has reshaped healthcare, listing improvements applied in medical imaging and tasks that these models address, in addition to detailing challenges like biases especially in skin lesion detection and way to detect them using evaluation metrics. In chapter 4, we present how the Latent Adversarial Debiasing mechanism works, describing its components, the structure of the datasets and the evaluation metrics that were used. In chapter 5 we set up the experimental environment and list all its details, including dataset pre-processing and code implementation key settings that were used. Additionally, we test the three pre-trained architectures on different hyperparameter settings of LAD and then proceed to compare them, first each model with itself then all models together. In chapter 6 we conclude on the experimental results, discussing future work and limitations.

2 Artificial Intelligence (AI)

Artificial Intelligence is a field in computer science where, via algorithms and mathematics, the computer tries to imitate human intelligence [5]. It involves methods that enable the computer to learn through past experiences, adapt to new data, and perceive its environment. In general, AI is a broader definition that encompasses any artificially created computational system designed to mimic intelligent and complex behavior, like decision making or understanding a language [6]. In Section 2.1, we will dive deep into machine learning and deep learning, as well as neural networks, how they work, development frameworks, and the challenges they have to face. In Section 2.2, core architectures like Convolutional Neural Networks (CNN), autoencoders and variational autoencoders, Generative Adversarial Networks (GAN), and pre-trained models will be introduced, as well as data preprocessing and augmentation.

2.1 Machine Learning and deep learning

Research in Artificial Intelligence traces back to 1943 when Turing and Champernowne found the first chess-playing computer program. Later [7] developed the first perceptron (Fig. 8) model, a hypothetical nervous system that could make predictions and classifications [8]. In this Section, we introduce concepts such as machine learning, deep learning, model architectures, and training methodologies that play a pivotal role in the present study.

Machine Learning (ML) is a part of Artificial Intelligence designed to imitate organic intelligence and encourage the computer to learn from its surroundings. The focus of ML is creating algorithms and statistical models that enable computers to learn independently from data. ML encompasses several learning paradigms, most notable are supervised, unsupervised, and reinforcement learning, each suited to different problem settings. Models are trained through examples, and their goal is to find the underlying patterns and capture them, so that the learned generalization can be applied to new, unseen data [9].

Deep Learning (DL) is a subcategory of machine learning that uses the multilayered structure of Artificial Neural Networks (ANN), imitating the architecture of the biological neural network to process more than one problem at a time. Specifically, DL focuses on deep neural networks (DNN) that are often denser and more complex models than traditional ANNs, enabling DNNs to learn hierarchical feature representations [10]. In general, DL excels when processing large and high-dimensional data like images, videos, speech, etc., without requiring manual feature engineering [11], [12].

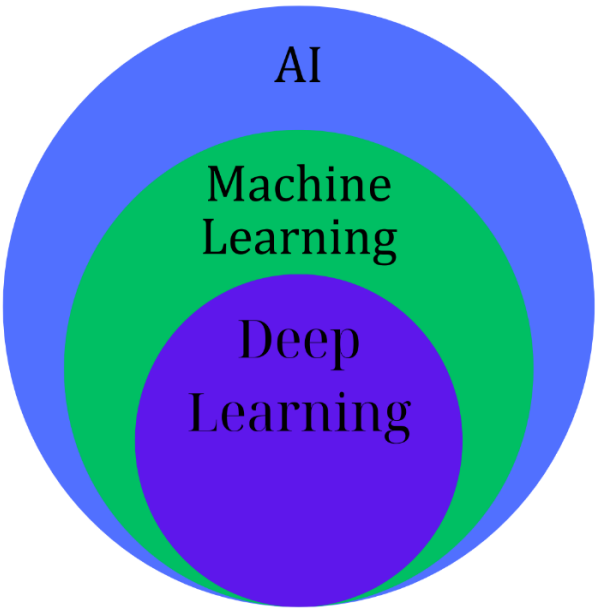

Fig. 1 Artificial Intelligence and its subsets.

2.1.1 Artificial Neural Network Foundations

In an attempt to recreate the information processing structure of a human brain, Neural Networks have been developed, composed of multiple nodes called neurons connected to each other [13]. Each neuron produces an output with the help of the activation function. Connections between two neurons denote the weight of a signal passing through them, called a weight, which can be interpreted as the network's ability to memorize[14]. The final output of the network may vary depending on the input, the corresponding weight values, and the activation function that was used. Generally, these networks are created to implement either some natural phenomena or a logical strategy [15]. ANNs can be computed in parallel, making good use of modern-day Graphics Processing Units (GPUs) and distributed architectures [16].

The neurons of an ANN can represent different concepts like letters, features, patterns, or more abstract meanings. The network can be divided into three layers: the input layer, the hidden layer and the output layer [17]. The input layer receives the real-world data that we want our model to be trained with. The output layer yields the results of the model, i.e., a final prediction or a classification. The hidden layer cannot be observed outside the network and serves as the connection between the input and the output layers [18]. These layers are connected through weights, and each neuron of the previous layer is connected to every neuron of the next layer. The computation of the weights and the connection between neurons translate to the representation and processing of the data inside the network. Additionally, to help the network cover a wider range of values in the model output a constant bias factor is introduced in the calculation of a neuron. The computation of a neuron output can be mathematically interpreted as follows:

$$\sum_i w_i \times x_i + b$$

Where *i* is the index of the neuron's input, w is the weight, x is the actual input, and b is a constant bias factor [7].

The activation function is applied after the calculation and introduces non-linearity to the model. This component of the ANN architecture is crucial for the model's learning process, absence of a non-linear activation function would effectively collapse the model into one singular linear transformation [19]. Some popular and effective activation functions are the Rectified Linear Unit (ReLU), Leak ReLU, Exponential Linear Unit (ELU), Parametric ReLU (PReLU), Sigmoid, Hyperbolic tangent (tanh), and Softmax [20]. Each of these functions has its application. For example, Sigmoid is used mainly in Binary Classification tasks because it outputs values in a range of [0,1], and Softmax is used in multi-class classification tasks as it outputs probabilities. The activation function illustrations and their formulas are shown below.

1. ReLU: $\begin{cases} x & if\ x > 0 \\ 0 & if\ x \leq 0 \end{cases}$
2. Leaky ReLU: $\begin{cases} x & if\ x > 0 \\ 0.01x & if\ x \leq 0 \end{cases}$
3. ELU: $\begin{cases} x & if\ x > 0 \\ a(e^x - 1) & if\ x \leq 0 \end{cases}$
4. PReLU: $\begin{cases} x_i & if\ x_i \geq 0 \\ a_i x_i & if\ x_i < 0 \end{cases}$ (where a_i is a learnable parameter)
5. Sigmoid: $\frac{1}{1+e^{-x}}$
6. Tahn: $\frac{e^x - e^{-x}}{e^x + e^{-x}}$
7. Softmax: $\frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$ [20]

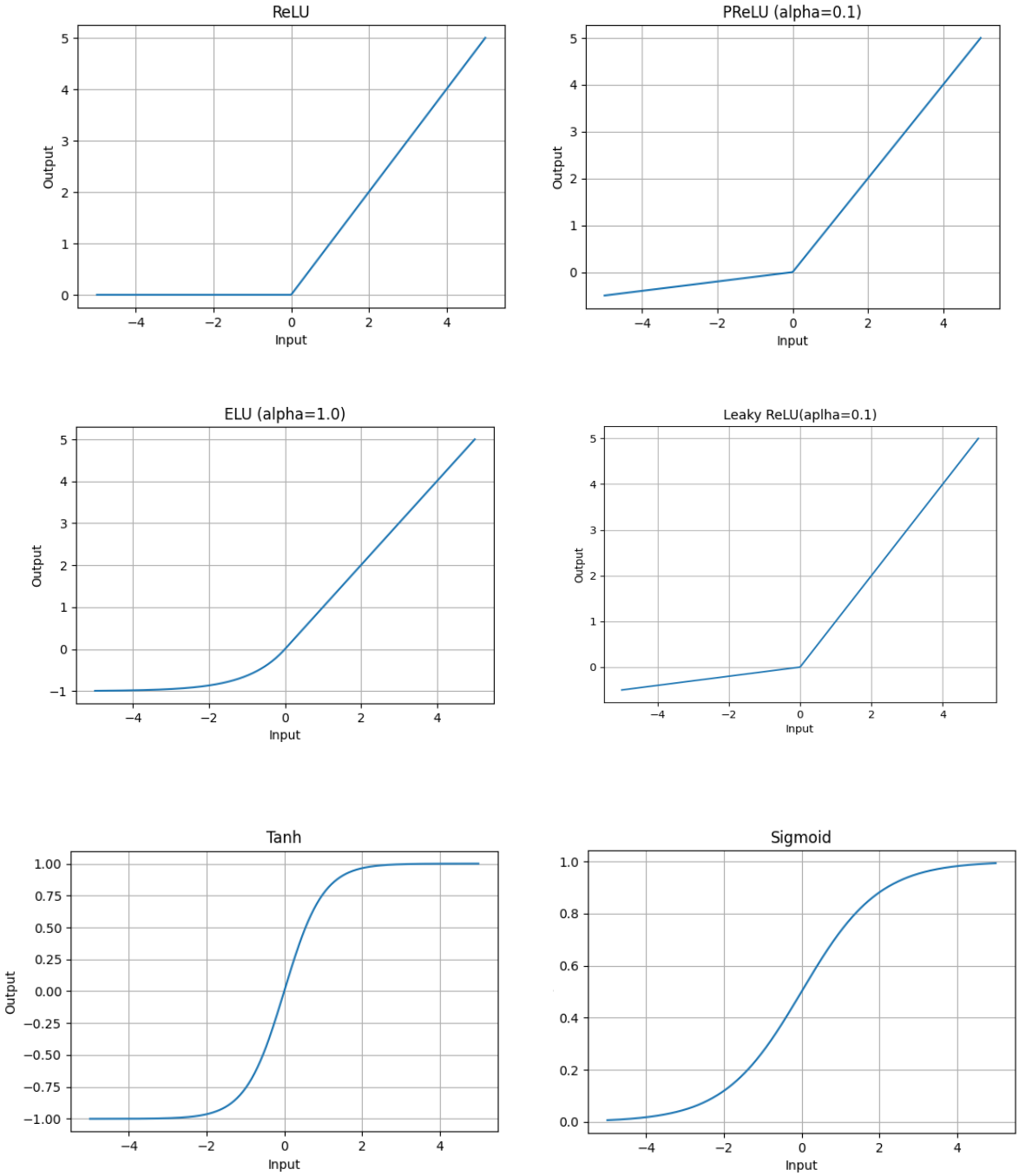

Fig. 2 Activation functions

The perceptron, first introduced in [7], is considered the foundation of AI. It introduced the concept of the neuron and served as a breakthrough for AI research. Later, the multilayer perceptron (Fig. 3) architecture was established in [21] marking the start of Deep feedforward networks trained through back-propagation.

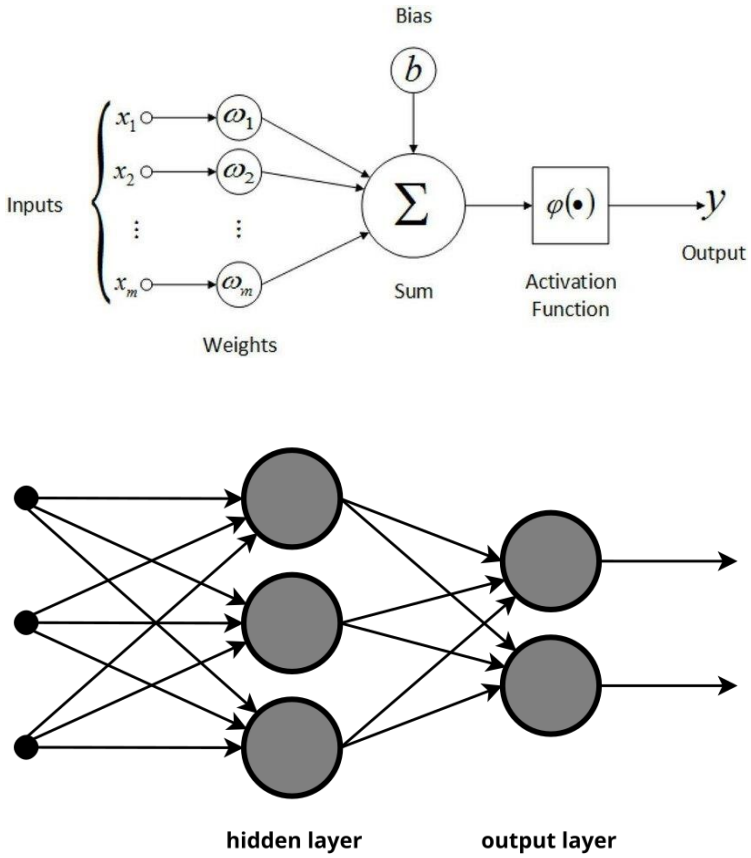

Fig. 3 Neuron – Multiple neurons, Multilayer perceptron [129]

Feedforward networks typically mean that at the start of training, the data course through the entire network from the input layer to the output layer. Most modern DL models have adopted this approach.

2.1.2 Training Procedure of a Neural Network

The training of a Deep Learning model can be partitioned into two different stages. The first stage is the forward stage, where the data passes through our model in small batches and produces an outcome. After the outcome is produced, it is evaluated by the chosen loss function. Loss functions play a central role in model training as they calculate how far off the model's predictions are from the desired output. [22]. Common loss functions are the cross-entropy loss function [23] typically used in classification problems, and Mean Square Error (MSE) [24] applied in prediction models like regression [25]. After the Error has been computed, the loss function guides the model's weight updates through gradient descent [26]. Gradient descent is an

optimization algorithm that uses backpropagation to compute the gradients of the loss function with respect to the model parameters. The gradient shows the direction of the biggest increase in loss and takes a step in the opposite direction. This step is controlled by a hyper-parameter called Learning rate. Hyper-parameters are tunable model parameters that need to be evaluated outside of model inference. Small learning rates cause the model to stagnate at local minimums while large learning rates cause the model to make big changes and never find the optimal weights for the model. The gradient descent is mathematically interpreted as follows:

$$p^{(t+1)} = p^{(t)} - l\frac{dL}{dp^{(t)}}$$

Where $p^{(t)}$ are the model's parameters at iteration t, l is the learning rate and L is the calculated loss function. Gradient descent and its variants, Adam[27], RMSprop [28] and Stochastic Gradient Descent [29] are some of the more commonly used and well-known available optimizers. To dynamically change the learning rate of the model, learning rate schedulers have been introduced to adjust the parameter with either static or dynamic strategies [30]. Regularization methods such as batch normalization, layer normalization, and dropout are widely used in model training. Batch normalization smooths the optimization landscape by using batch statistics, enabling faster and more accurate training [31]. Layer normalization performs normalization on layer neurons and does not require batch statistics [32]. Dropout simply zeros out some of the neurons randomly to prevent model overfitting, while also reducing computational cost, as there will not be as many neurons to compute as before [33]. When all the above techniques have been implemented and the model does not seem to improve, we use Early Stopping based on a stopping criterion to inspect and tune the model settings and hyper-parameters[34]. After all the above functions have been executed, the model will then pass through the data multiple times (epoch number), performing the same operations and gradually improving the model.

Another essential part of the training procedure is how we handle data before they are processed. Data are present in datasets where the structure may vary, but in general, datasets include a metadata file that describes the data and a partition for train and test sets. Datasets, more often than not, are not fully ready for processing from the start. Data mining introduces data preparation techniques that are key to training efficient and accurate models. For the model to generalize, we need to partition our dataset, if it is not already, rather than simply feeding all the data in training, this would cause the model to overfit, which will be discussed in Section 2.3 [35]. Data cleaning is also important for model training as it removes unusable data such as null (empty) values, outliers that attempt to throw off the model, and fix inconsistencies [35], [36]. Another important aspect of data preprocessing is data integration [35], [37]. Data from multiple sources needs to be evaluated and integrated into a single dataset that will be used in training. Finally, data augmentation is important to model learning, as our dataset usually will not contain all the data we would like it to have. In

cases where there is insufficient data, class imbalance occurs in classification problems, or there is limited variance in our dataset, data augmentation is a good way to address these issues. Augmentations are applied based on a given probability on the whole dataset and are either saved in the dataset or created for model training only [38]. Some common augmentations include:

Random horizontal flip mirrors the image along its vertical axis, helping the model generalize to left–right variations. Random vertical flips work similarly but flips along the horizontal axis, which can be useful for the same reasons as horizontal flip. Random rotation rotates the image by a small angle within a defined range, allowing the model to become robust to orientation changes. Random crop extracts a smaller portion of the image, often followed by resizing back to the original dimensions, encouraging the model to focus on different regions of the input. Random erasing removes a rectangular patch from the image and fills it with random values or a constant color, simulating occlusion and forcing the model to learn from incomplete information. Gaussian blur smooths the image by averaging pixel values using a Gaussian kernel, mimicking out-of-focus effects or sensor noise. Color jitter randomly alters brightness, contrast, saturation, and hue, teaching the model to ignore lighting variations. Random perspective warps the image by slightly shifting the positions of its corners, simulating changes in camera angle and viewpoint. Finally, random affine transformations combine scaling, translation, rotation, and shearing in a linear fashion, providing the model with invariance to geometric distortions. (Fig. 4).

To practically implement all the above, several Python frameworks exist for DNN development. The most well-known framework for research is PyTorch, a library that provides a vast selection of implementations and high modularity in a MATLAB-like environment. Lua was primarily chosen for PyTorch's development and backend, along with C and C++, which significantly accelerate the execution speed due to low-level code execution. PyTorch uses a new data type called Tensor to efficiently calculate the necessary gradient computations on a GPU if possible. PyTorch can run on Linux, Windows, Android, and iOS operating systems and supports distributed code execution on multiple CPUs and GPUs, further decreasing the execution time. Other frameworks like TensorFlow, Caffe and Theano also exist that offer similar capabilities [39]. The implementation of this study is coded in PyTorch and can be found in [40].

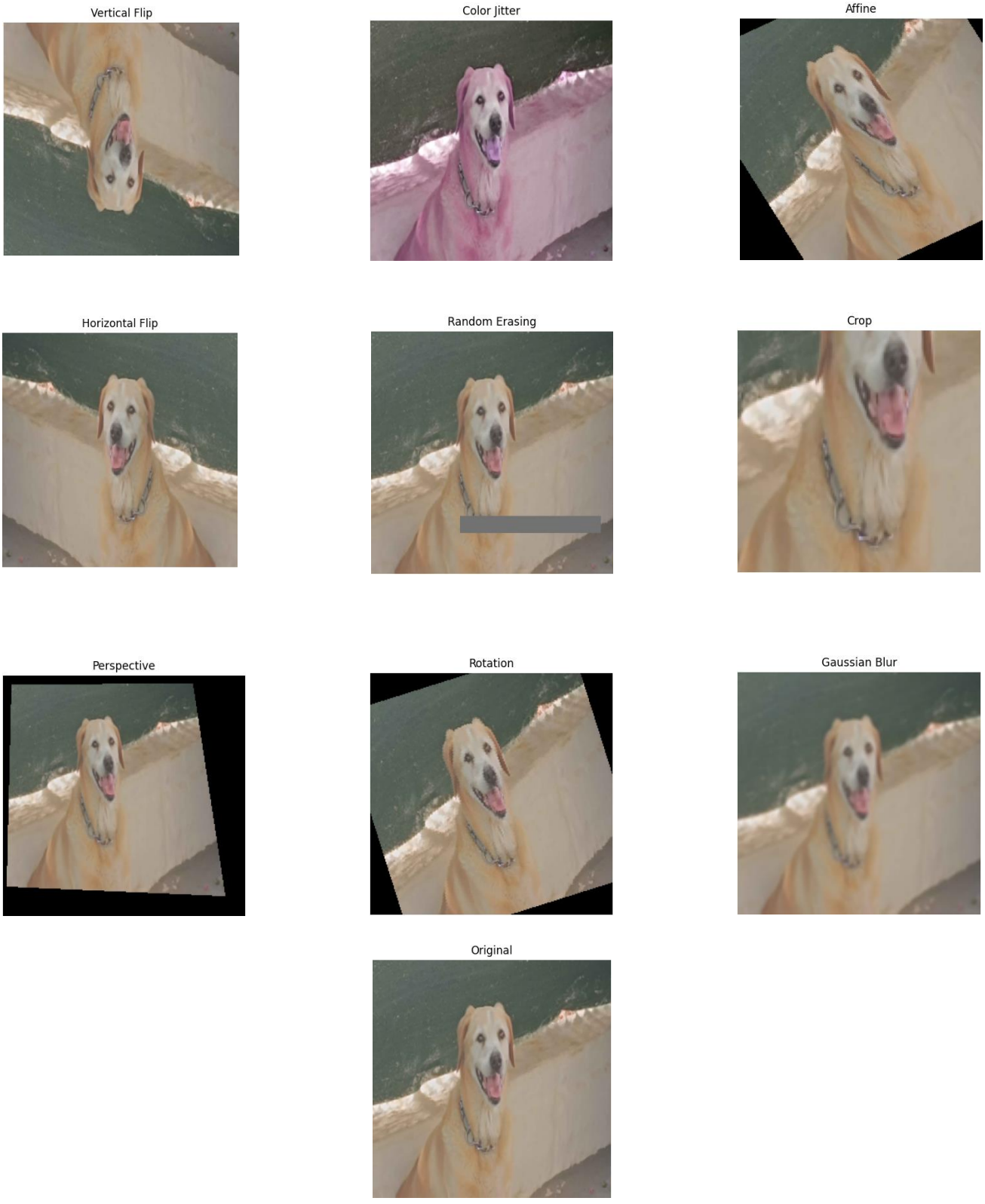

Fig. 4 Data augmentation Examples

2.1.3 Overfitting, Underfitting, and Bias

While artificial intelligence models have shown remarkable performance in their respective applications, common pitfalls emerge from poor data management quality or architectural design. Overfitting occurs when a model demonstrates exceptional performance on training data but fails to generalize to new unseen data, from the same distribution as the original data [41]. Several factors can be attributed to model overfitting, such as the model learning the exact representation of all data points, including "noisy" data, and the model being too complex for our data and problem at hand [41], [42]. Underfitting is the opposite of overfitting. Sometimes, due to insufficient training time, too simplistic model architecture, or lack of data, our model will not be able to generalize well, and its performance on both the train and test set will be sub-optimal as it fails to capture the needed underlying patterns [41], [42]. Bias refers to systematic errors that lead to the model favoring a specific outcome or group over another, often due to imbalances or gaps in the training data [43]. As highlighted in recent research, bias can stem from unrepresentative datasets that fail to capture the full diversity of the target population, leading to disparities in model performance across demographic groups [44]. For example, in medical imaging, if the training data underrepresents certain skin tones or age groups, the model may perform worse for those populations, reinforcing existing healthcare inequalities.

In summary, overfitting, underfitting, and bias are fundamental challenges that can significantly impact the performance and fairness of AI models. Effectively addressing these issues requires careful model design, attentive data management, and continuous evaluation to ensure that AI systems are both accurate and equitable across diverse populations.

2.2 Common architectures

Having explored the essentials of training neural networks, it is now crucial to understand the architectures that learn from data. Deep learning encompasses a variety of specialized models tailored to different tasks and data modalities. This section explores some of the core principles of the architectures used in the present study.

2.2.1 CNNs

The idea of Convolutional Neural Networks is to mimic human vision and perception, and was first introduced in [45] and practically implemented in [46]. Since then, they have dominated the field of computer vision and pattern recognition on images and videos [47]. The basic CNN can be divided into 3 key components.

The first is the Convolutional layer, which focuses on finding the best kernels for the current task. After an image input, the image itself is represented by a matrix, with the pixel values being the values inside the

matrix. Kernels are smaller filters that slide across the input image, performing element-wise multiplication and summing them to produce feature maps that can highlight specific parts of an image, like edges, shapes, or textures [47], [48]. Convolutional layers can be tuned by 3 hyperparameters: depth, stride, and padding. Depth controls how many filters are applied in a single convolution to output the same number of feature maps. Generally, larger depths lead to more diverse and complex features at the cost of training time. The stride parameter controls the step that the kernel executes at every application, skipping multiple pixels if adjusted to reduce training time at the expense of output quality. The padding parameter simply adds a blank pixel border around the image to further control the image dimensions [49]. After the convolutions are done, the image's dimensions will be greatly altered. To calculate the dimensions, we use the following formula:

$$\frac{(I - K) + 2P}{S + 1}$$

Where I represents the input image channels (3 if RGB, 1 if grayscale, or 13 if a Satellite image), K is the kernel size, P is the padding size and S is the stride.

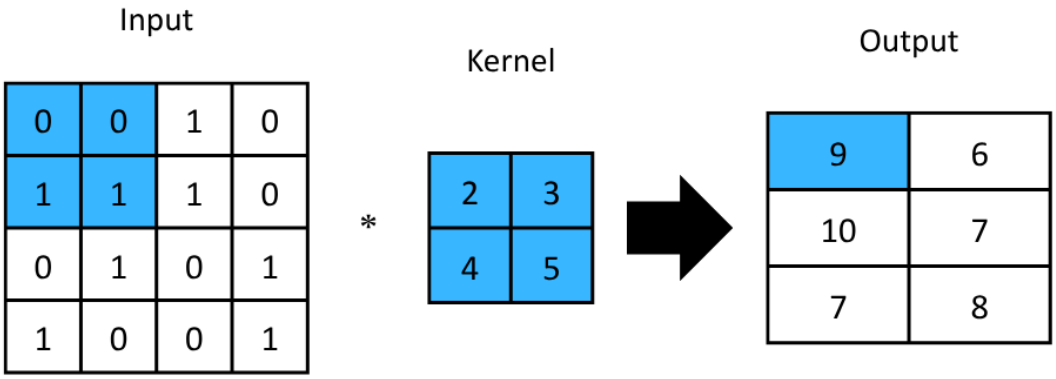

Fig. 5 Kernel application on the input image.

The second component is the pooling layer, which performs simple down-sampling of the input, reducing the number of parameters. The pooling layer is applied over all extracted feature maps and operates based on the set configuration, most common being the max pooling configuration (Fig. 7). In essence, max pooling summarizes the area it is applied to; for example, with a 2×2 filter, the max function selects the largest value inside the kernel. This helps reduce the spatial dimensions of the image at the cost of quality [49], [50].

The final component is the fully connected layer, which flattens all the spatial dimensions to one, then calculates the probabilities for the output classes, like a traditional ANN [48].

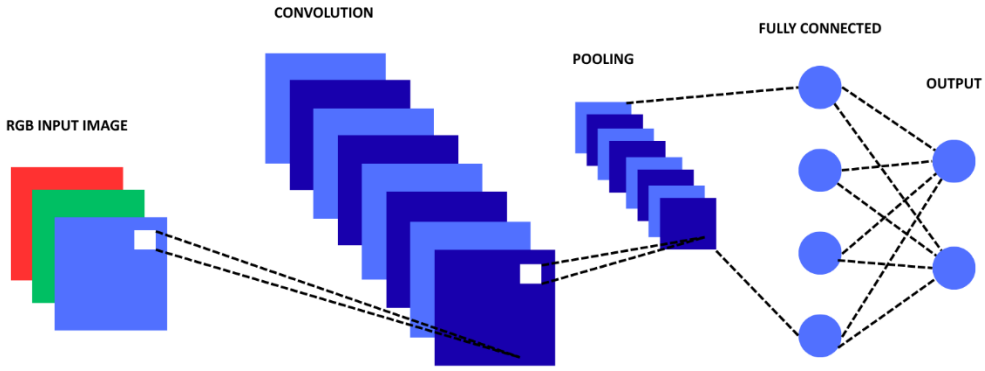

Fig. 6 Convolutional Neural Network illustration

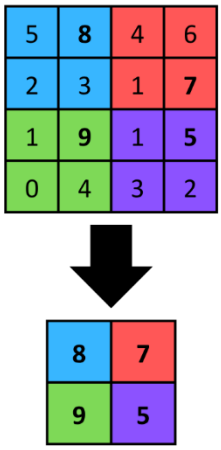

Fig. 7 Max pooling example

2.2.2 Autoencoders

Autoencoders (AE) are a type of neural network that encodes the given input into a meaningful, compressed representation and then proceeds to decode it, effectively learning an optimal compression method to deconstruct and reconstruct data [51]. First introduced as a generalization of Principal Component Analysis (PCA) [52], a data mining dimensionality reduction technique [53], then implemented in [54] . The autoencoder model has been used in anomaly detection [55], image processing using convolutions (Fig. 8) [56], compression, and many other fields.

An autoencoder is composed of two parts: an encoder and a decoder. The encoder maps data to a lower-dimensional latent space, far smaller than the original, imposing a bottleneck in the network, significantly

reducing the input's dimension. The decoder part does the opposite of the encoder, decoding the latent representation to reconstruct the input image. Combining all the above with a loss function to tune the parts results in the encoder learning an optimal compression structure and the decoder learning to reconstruct it back to the original input. Since an autoencoder performs a regression-like operation, the most common function is the MSE loss to measure the reconstruction error [57] or Binary Cross Entropy loss for binary images [58]. Building on autoencoders, convolutional autoencoders can compress and reconstruct images using deconvolution layers that find a set of kernels and feature maps to rebuild the original image, and are also able to magnify them with super resolution [59], [60].

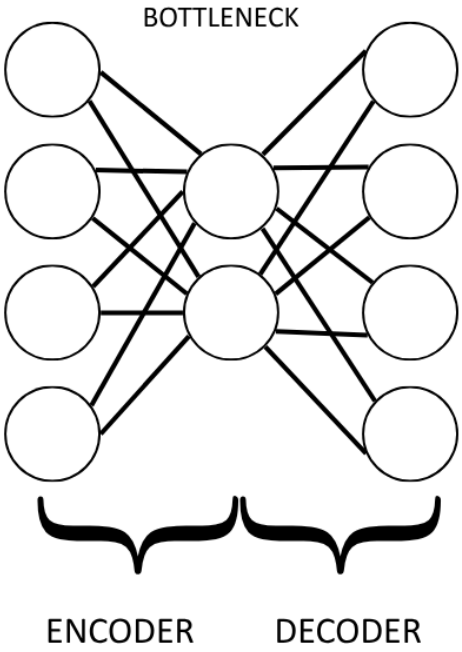

Fig. 8 Basic autoencoder architecture.

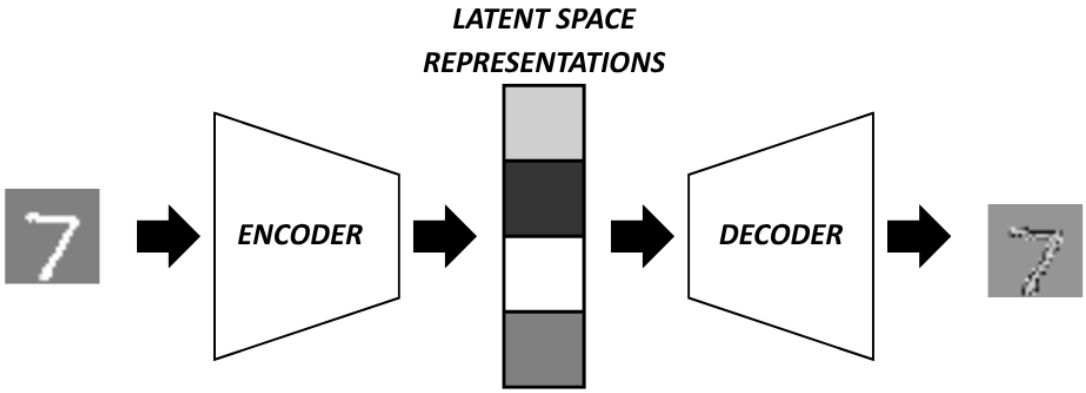

Fig. 9 Autoencoder with convolutions example showing image reconstruction of MNIST handwritten number dataset. The autoencoder was trained for 2 epochs for demonstration purposes.

The Variational autoencoders (VAE) are a probabilistic class of autoencoders that have achieved significant improvements in the latent space representation operations. Although they share the encoder-decoder architecture, VAEs introduce a probabilistic framework, modeling the encoder as an approximate posterior distribution over latent variables and the decoder as a generative model that samples from this distribution to reconstruct or synthesize new data [51], [57].

2.2.3 GANs

Generative Adversarial Networks (GANs) are revolutionary models in the field of generative AI. They were first introduced in [61] and have been widely used for data and image generation, upscaling image resolution, face aging, and many more applications. They involve training a pair of networks that are in competition with each other, the generator and the discriminator. The generator model does not have access to real images; instead, it uses random noise input to try and map it into a synthetic data sample so that the discriminator thinks it is a real, non-synthetic image. The discriminator uses inputs from both real images from the training dataset and synthetic images from the generator and tries to distinguish which is real and which is synthetic. The error of the discriminator measures how well it separates real from synthetic images, while the generator uses that same error to see if the generated image managed to fool the discriminator [62]. The GAN's objective function is to match the original image data distribution with the generator's mapping of noise. This expands to the discriminator aiming to minimize the binary classification error and the generator aiming to maximize the probability of its generated images being real [63].

Some of the most used GAN architectures include the Deep Convolutional GAN (DCGAN) [64] which uses convolutions instead of simple linear layers and Conditional GANs (cGAN) [65] which conditions both the discriminator and generator to extra information about the data in a separate input layer. These architectures have been instrumental in advancing the quality of the generated outputs and have enabled applications in photorealistic synthetic images and domain-specific image generation.

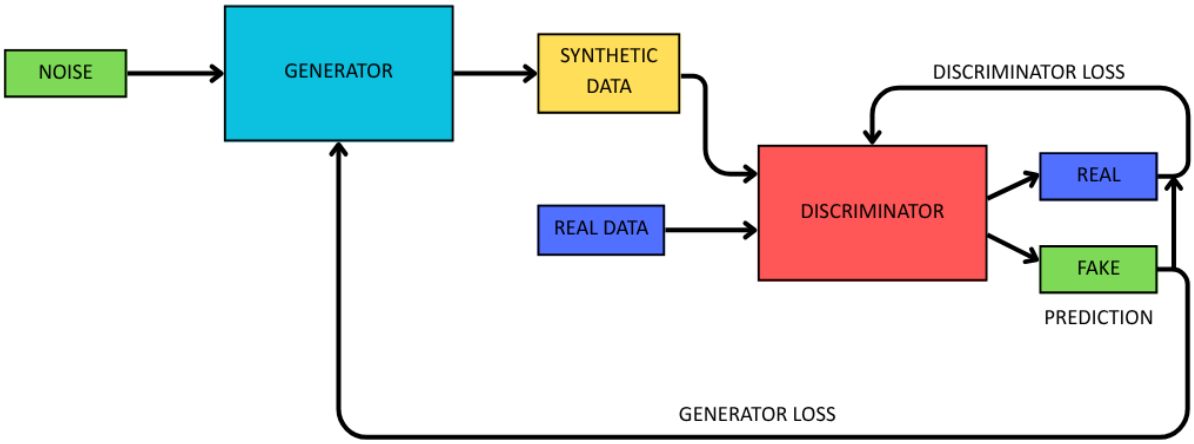

Fig. 10 GAN network architecture

2.2.4 Pre-trained models and transfer learning

Transfer learning is a type of learning procedure that utilizes already existing models to solve current challenges. These models are already trained on a huge dataset, most commonly the ImageNet dataset, consisting of 15 million labeled images [66]. They are often large, making them easily reusable in a lot of scenarios with a small amount of fine-tuning [67]. Some popular pre-trained models include ResNet, DenseNet, and EfficientNet.

ResNet introduced in [68], uses residual learning (Fig. 11) that learns a residual function to propagate the input forward, reducing the difficulty of optimizing deeper CNNs (150+ layers). Since then, skip connections have become a standard method in DL and have inspired many other great advancements like DenseNet, Transformer models, and Unets. There are numerous models of ResNet, each with higher levels of layers: ResNet18, ResNet34 (Fig. 12), ResNet50, ResNet101, and ResNet152, with the number referring to the number of layers inside the network.

DenseNet introduced in [69], builds on ResNet by connecting each layer to all its subsequent layers. This increases feature reusage, has fewer parameters, and helps the optimizer be more efficient. DenseNet models include DenseNet-121, DenseNet-161, DenseNet-169, and DenseNet-201.

EfficientNet proposed in [70] uses mobile convolutions for efficiency and introduces compound model scaling which scales model depth, width and resolution via a controlled parameter φ for better resource management. EfficientNet models are based on the value of φ : EfficientNetB0, EfficientNetB1 …, EfficientNetB7.

These pretrained architectures offer a wide spectrum of accuracy-efficiency trade-offs, making them powerful starting points for transfer learning across domains.

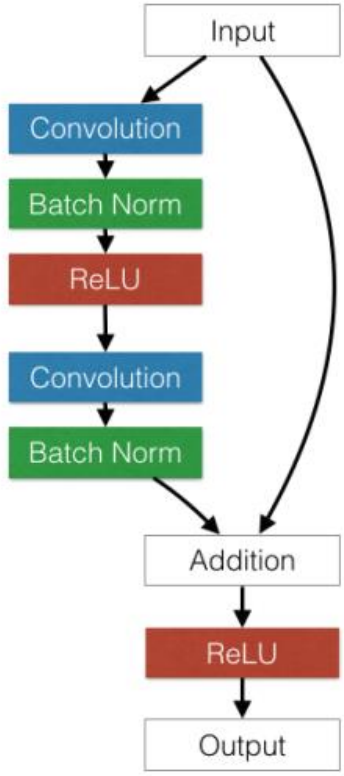

Fig. 11 ResNet residual block [130]

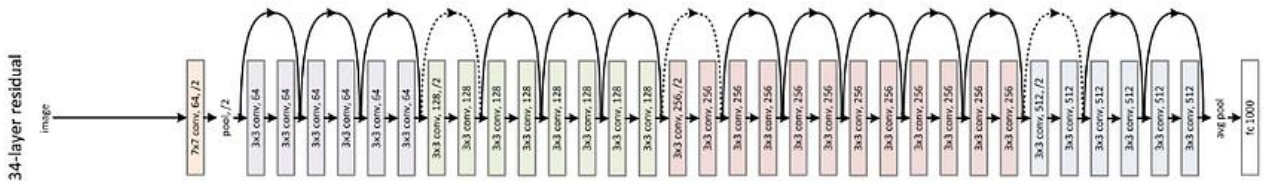

Fig. 12 ResNet34 architecture [130]

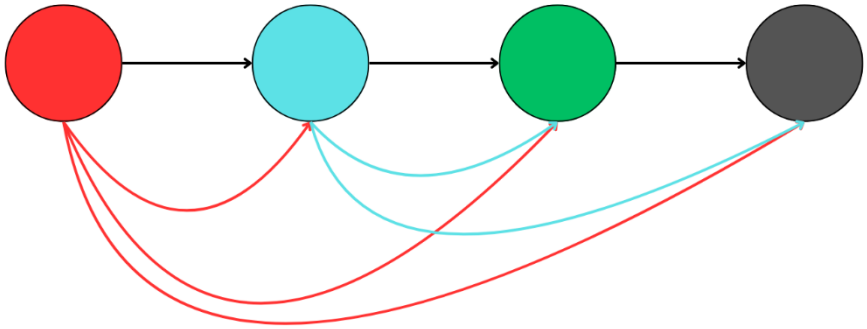

Fig. 13 DenseNet Dense connectivity illustration

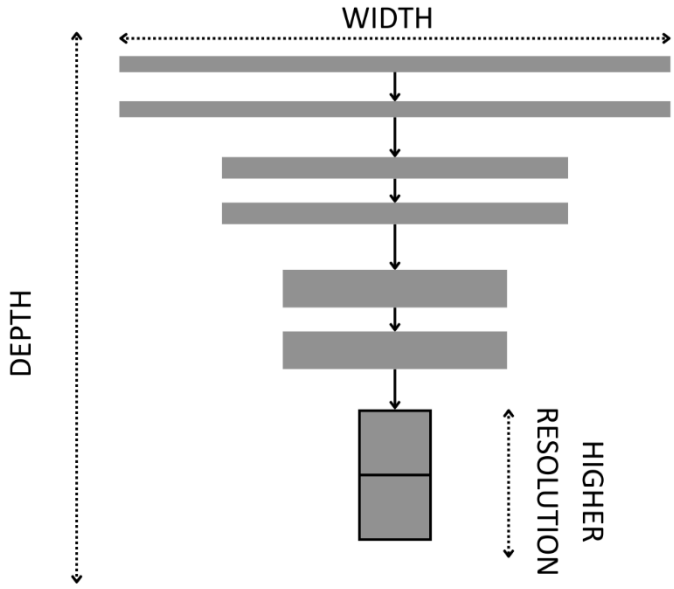

Fig. 14 EfficientNet dynamic scaling technique scaling the three dimensions simultaneously.

3 Deep Learning in Medicine

With the rapid transformation of healthcare data to electronic documents, ML and DL have emerged as pivotal technologies in healthcare, enabling a wide range of applications such as computer-aided diagnosis, medical image registration, annotation, segmentation, multimodal image fusion, image-guided interventions, and medical image database retrieval [71]. These capabilities are essential in addressing diagnostic deficiencies that, in some cases, may have irreversible consequences. In the following sections, prior work in the medical field will be discussed (3.1), proposed methods in medical imaging (3.2), and finally, we will focus on fairness and bias in medicine and how they can be fairly mitigated (3.3).

3.1 Overview

Deep learning has become integral to many medical imaging tasks, from anatomical modeling (e.g., segmentation of organs and structures) to tumor detection, disease classification, and surgical planning. Beyond image analysis, DL techniques have also enhanced computer-aided diagnosis (CAD) systems and decision support tools, enabling more precise, consistent, and scalable interpretation of complex medical data. The first application of CAD systems dates to 1998 when it was approved by the Food and Drug Administration (FDA) to be used as a second opinion in screening mammography [72], [73]. Since then, a plethora of DL models have been used to aid physicians, such as autoencoders diagnosing Alzheimer's disease early from brain Magnetic Resonance Imaging (MRIs) [74] and CNNs classifying skin cancer [75]. Beyond imaging, DL has made significant contributions to analyzing Electronic Health Records (EHR) and genomics. For instance, using Long Short-Term Memory models (LSTMs) to predict medicine based on a patient's history [76], autoencoders to predict future diseases and clinical events [77], [78], [79], gene protein backbone predictions, and cancer prediction from gene expression profiles [80], [81]. While DL's influence spans diverse areas of healthcare, its most profound advancements and clinical integrations have emerged in medical imaging, which we explore in greater depth in the following section.

3.2 Medical Imaging

With recent advancements in CNNs and related deep learning architectures, computer vision has achieved significant advances in medical imaging. These techniques now enable solving a variety of tasks, including image classification, object detection, image segmentation, and image enhancement, each playing a vital role in clinical decision-making. From classifying medical images into diagnostic categories [80], [82], to detecting and localizing abnormalities such as tumors or lesions [75], [80], [83], to segmenting organs and pathological

regions at the pixel level [84], these methods have significantly improved diagnostic accuracy and consistency. Furthermore, image enhancement techniques such as super-resolution reconstruction, denoising, and contrast adjustment have enabled higher-quality visualization of medical data, even from low-dose or degraded scans [60], [85]. The aforementioned capabilities of CNNs not only reduce interpretation time for physicians but also facilitate early disease detection, more precise treatment planning, and improved patient outcomes.

3.2.1 Image Classification

Image classification refers to the process of categorizing an image by associating it with one or more labels, which is one of the fundamental tasks in computer vision. Traditionally, the procedure of classifying images was to use a low-end feature-extracting technique and then propagate the outcome to a machine learning classifier like Support Vector Machines (SVMs) [86]. DL approaches have recently been proven more powerful than traditional methods, making the use of CNNs in image classification the new standard due to their ability to automatically learn hierarchical feature representations and achieve state-of-the-art performance [66], [87]. DL architectures have been used to classify different categories of interstitial lung disease patterns in high-resolution computed tomography [87], detect focal liver lesions on multiphase computed tomography images [66], classify breast cancer from mammography MRI images [88], and classify histopathology images automatically [89] in addition to numerous other medical imaging tasks.

3.2.2 Object detection

Object detection goes beyond image classification by not only recognizing the presence of multiple objects within an image but also localizing them using bounding boxes and confidence scores. This task is fundamental in medical imaging, where precise localization of abnormalities such as tumors, lesions, or nodules is crucial for diagnosis and treatment planning. Traditional methods for object detection relied on hand-crafted features combined with sliding window techniques, which were computationally expensive and often inaccurate [90]. Several models have been proposed to tackle object detection, like YOLO (You Only Look Once)[91], Single Shot MultiBox Detector(SSD) [92], R-CNN, Fast and Faster R-CNN [93] showing promising performance even on medical data on which they were not explicitly trained. Abnormality detection in X-ray imaging, cellular structure or tissue anomaly identification, endoscopic and micro-endoscopy location, and classification of lesions during procedures [94] are a few of the documented applications of these models.

3.2.3 Image Segmentation

Image segmentation refers to the process of partitioning an image into multiple meaningful regions or structures, allowing detailed analysis at the pixel or voxel level. In medical imaging, segmentation is essential for identifying and delineating anatomical structures, lesions, and other regions of interest across modalities such as CT, MRI, PET, and ultrasound. Traditional segmentation methods relied on thresholding, region

growing, or edge detection, but these approaches often struggled with complex medical images [95]. Architectures such as Fully Convolutional Networks (FCNs) and U-Net have become the standard, demonstrating robust performance in extracting features from multimodal medical images. Applications of segmentation include tumor and lesion detection [75], organ and tissue delineation, treatment planning, disease progression monitoring, and cellular analysis in microscopy images [95], [96], [97], [98]. By enabling precise localization and quantification, deep learning-based segmentation plays a critical role in advancing CAD and supporting personalized medicine.

3.2.4 Image enhancement

Image enhancement refers to the process of improving the visual quality of medical images by increasing contrast, reducing noise, and highlighting relevant anatomical details. Unlike higher-level tasks such as classification or detection, enhancement operates as a crucial preprocessing step that directly impacts the accuracy of subsequent analysis. Traditional approaches include spatial domain methods, such as histogram equalization, gamma correction, and adaptive contrast enhancement, as well as frequency domain techniques like Fourier and wavelet transforms. Recent deep learning-based approaches have further advanced this area by learning data-driven mappings for denoising, super-resolution, and contrast adjustment [60], [78], [88]. In medical imaging, enhancement has been applied across multiple modalities: improving mammography for early breast cancer detection, refining cardiac MRI and echocardiography for more accurate diagnosis of cardiovascular disease, enhancing MRI brain scans to visualize tumors and lesions, and improving histopathology images for cellular-level analysis [60]. By producing clearer and more informative images, enhancement techniques facilitate both human interpretation and the performance of downstream AI models, contributing to more precise and reliable clinical decision-making [99].

3.3 Fairness and bias mitigation

While preceding applications in medical imaging have made significant contributions in the field of medicine, difficulties arise when sensitive attributes like sex, age, and ethnicity are present in the dataset. These attributes can unintentionally act as discriminators, leading the model to pick up counterfeit correlations instead of true disease-related signals.

3.3.1 Collider bias

A particular concern is collider bias, which occurs when the data collection or sampling process entangles non-causal factors with the target outcome [100]. For example, if images of certain demographic groups are more likely to be acquired with specific scanners, under different conditions or the demographic groups have significant differences, models may rely on these background signals rather than the true clinical features of interest. Swayamdipta S. et al mention that DNNs are prone to latching onto “easy-to-learn” features inside the data, like background [101], [102]. Darlow L. et al reinforced this claim by introducing the one-pixel problem, a toy scenario in the MNIST dataset, where a classifier would typically train in the dataset, but the images would change a pixel based on the image’s label [103]. The classifier would then learn the change that corresponds to the pixel change and not the actual number. Such biases not only reduce generalization performance but also risk propagating health disparities. Addressing them requires fairness-aware strategies such as balanced data collection, adversarial debiasing, and subgroup evaluation to ensure that models rely on causal signals rather than confounding ones.

3.3.2 Fairness metrics

To identify if a DL model discriminates against a specific demographic group, fairness metrics have been introduced to evaluate model fairness and performance. Three main metrics are used in the present study: Equal Opportunity (EO), Impact Disparate (DI), and Accuracy Parity (AP) [100], [104].

To make sure our algorithm does not favor specific groups, Equal Opportunity measures the parity of true positive rates between two groups and can be interpreted as follows:

$$EO = \frac{P(True\ Positive\ of\ Protected\ Group)}{P(True\ Positive\ of\ Non\ Protected\ Group)}$$

Where $P(True\ Positive\ of\ Group)$ denotes the likelihood of accurately recognizing positive cases for the said group. An $EO \approx 1$ means that both groups have equal true positive rates, meaning that our algorithm is fair.

To evaluate the decision-making process of a classifier, DI measures the successful outcome of the classification between groups. It can be interpreted as follows:

$$DI = \frac{P(Positive\ Prediction\ of\ Protected\ Group)}{P(Positive\ Prediction\ of\ Non\ Protected\ Group)}$$

Where $P(Positive\ Prediction\ of\ Group)$ indicates the likelihood of the classifier making a positive prediction, i.e., classifying the case are 1 for binary classification, A $DI \approx 1$ means that both groups are being equally classified as positive.

AP compares the accuracy of classifications between groups. It can be formulated as follows:

$$AP = \frac{Accuracy\ of\ protected\ group}{Accuracy\ of\ non\ protected\ group}$$

Where accuracy means correctly predicting a case for the group.

These fairness metrics provide a complementary perspective on model equity. While Equal Opportunity focuses on balancing error rates, Disparate Impact evaluates parity in decision outcomes, and Accuracy Parity ensures overall predictive performance remains consistent across groups. By jointly applying these measures, we obtain a more holistic assessment of fairness, reducing the risk that the model inadvertently favors or disadvantages any demographic subgroup.

3.3.3 Mitigation techniques

Mitigating biases has been an active research topic in ML and DL. A number of methods have been proposed to combat bias, including pre-processing, in-processing, and post-processing. Pre-processing methods focus on preparing the data to be free of bias. Methods include bias suppression, where the sensitive attribute is completely removed from the training data, oversampling and undersampling imbalanced classes, random perturbations like introducing noise to the data, re-weighting training instances without changing labels or input data, and fair representation learning [105], [106], [107]. In-processing techniques refer to actively incorporating fairness into a model or an algorithm. Such techniques include adversarial de-biasing, where two competing models are trained to minimize bias, the use of a prejudice remover regularizer that integrates fairness directly into the objective function, exponentiated gradient reduction, which introduces fairness constraints through cost-sensitive classifiers, and fair model training with encrypted attributes using multi-party computation[105], [108], [109]. Post-processing methods, on the other hand, operate on model outputs to adjust predictions to improve fairness. Examples include gradient feature auditing, which checks whether

sensitive attributes influence classifier decisions, calibrated equalized odds that balance false positive and false negative rates across subgroups, and randomized threshold optimization that satisfies fairness criteria while maintaining accuracy and minimizing disparities [105], [107]. While these methods have been widely applied in medicine, skin lesion detection has not received as much attention as it should until recent studies spanning the last 3 years [110], [111].

3.3.4 Bias in skin lesion detection

Skin melanoma classification has been a popular benchmark for a lot of studies and pre-trained deep learning models [110]. Unfortunately, the majority of these models do not account for skin tone variety that may exist inside the dataset or apply in real-world scenarios, making DL models biased unintentionally [110], [112]. Recent efforts to mitigate skin tone bias include variational autoencoder feature extraction and resampling, pre-processing practices along with pre-trained classifiers using transfer learning, augmented datasets and using style transfer on synthetic data to better train a classifier, training a swin transformer model along with a CNN classifier and using diffusion models [113], [114], [115], [116]. While these strategies highlight the progress in DL fairness, they also underline that no single solution can fully eliminate bias in medical imaging. Effective bias mitigation often requires a multi-faceted approach, combining data-level interventions, algorithmic fairness constraints, and careful evaluation with fairness metrics tailored to clinical contexts. As datasets become more diverse and models more advanced, the integration of fairness-aware design from the ground up will be essential to ensure that automated melanoma classification systems are not only accurate but also trustworthy and clinically applicable across diverse patient populations.

4 Proposed System

We acknowledge that real-world datasets often lack balanced demographic group representation. To address this, the present work adapts the Latent Adversarial De-biasing network (LAD) [103] for application in skin lesion detection. A key strength of LAD is that it does not require bias-free data during training, making it suitable for medical imaging contexts where collecting perfectly balanced datasets is impractical. By learning fair latent representations through adversarial training, LAD can be integrated into lesion classification tasks to enhance both accuracy and fairness across diverse patient populations. The following subsections detail the architecture of the proposed LAD-based framework (4.1) and its dataset and evaluation setup (4.2).

4.1 Latent Adversarial Debiasing

LAD's main objective is to counter collider bias mentioned in section 3.3.1 by augmenting the training data to separate biased information and keep the causal signals needed for the classification process. LAD consists of three main components: A Vector-Quantized Variational AutoEncoder (VQ-VAE) that produces a latent space that can approximately separate causal and confounding signals, a simple classifier trained on the latent space produced by the VQ-VAE, and a method to remove the confounding signals that the biased classifier was trained on [103].

4.1.1 VQ-VAE

The VQ-VAE is effectively an autoencoder architecture with the key difference that it produces a quantized latent space. This means that the encoder outputs a discrete representation instead of a continuous. To achieve this, VQ-VAE uses a fixed, trainable codebook, where there is a static number of vectors that can be assigned to values. When the encoder produces a representation, it is then matched to the closest value of the codebook using L2 norm nearest neighbor look-up. This process is called the vector quantization process. Since the quantization is non-differentiable gradients of the decoder are copied to the encoder. After the quantization, the decoder learns to reconstruct the image from the discrete representations. While the codebook size is static, its embeddings are trainable; each training iteration moves the embeddings closer to the encoder output to increase codebook utilization (perplexity) and decrease reconstruction loss, helping the codebook adapt to the data distribution. The connection in the backpropagation [117]. The loss function can be interpreted as follows:

$$L = \log p(x|z_q(x)) + \|sg[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - sg[e]\|_2^2$$

Where sg stands for stop gradient, and it is used to constrain the operand to be a non-updatable constant, e stands for the codebook embedding, x is the input, z_e is the output of the encoder, z_q is the closest encoding vector to the encoder's output, β is the commitment loss term that tunes the level of codebook commitment and $||\cdot||_2^2$ is the operator denoting the square Euclidean distance between the two encodings. The first term is optimized by both encoder and decoder, the second term optimizes the embeddings, and the last term is optimized by the encoder only [117].

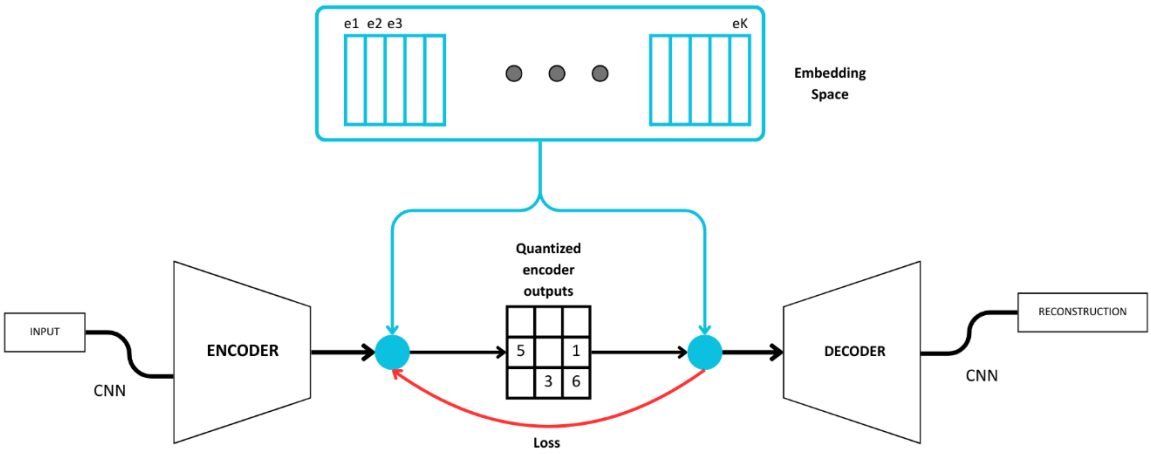

Fig. 15 VQ-VAE architecture illustration

4.1.2 Adversarial walk

The classifier is attached to the encoder's latent space. While the latent space contains both confounding and causal signals, the classifier is prone to latch onto the easy-to-learn features and enable LAD to work. While conventional adversarial attacks operate in pixel space by perturbing inputs to maximize classification loss, LAD performs the walk directly in the latent space of the encoder with the objective of maximizing the entropy of the simple classifier.

$$max_{\delta} H(f(quantise(h+\delta, G)))$$

Where H is the entropy of the classifier's prediction probabilities, $f(\cdot)$ is the simple classifier, h is the input's latent representation and $quantise(\cdot)$ refers to VQ-VAE's (G) quantization mechanism. The perturbation strength is influenced by δ:

$$\delta = a \cdot \frac{\frac{\partial H}{\partial h} - mean\left(\frac{\partial H}{\partial h}\right)}{std\left(\frac{\partial H}{\partial h}\right)}$$

ALGORITHM 1: QUANTIZATION-CONSTRAINED ENTROPY TARGETED ADVERSARIAL WALK.

INPUT : $f(\cdot), y, h, a, steps, G() \cdot$

OUTPUT: $Adjusted\ latent\ representation, h_\delta$

$h_{delta} \leftarrow h$ $\quad$ $\#\ initialize\ h_\delta\ to\ h$

FOR $i \leftarrow 1$ **TO** $steps$ **DO**

$\quad \hat{y} \leftarrow f(h_{delta});$ $\quad \#\ prediction$

$\quad H \leftarrow entropy(\hat{y});$

$\quad backpropagation\ to\ maximise\ H;$

$\quad compute\ \delta;$

$\quad h_{delta} \leftarrow h_{delta} + a \cdot \delta$

$\quad h_{delta} \leftarrow quantise(G, h_{delta})$

END

Where mean and std refer to mean and standard deviation computed over the gradient and a is a hyper-parameter controlling the step size of the adversarial walk.

This approach preserves perceptual consistency while systematically probing which features the classifier relies on. After the walk has been executed, the latent representation should be adjusted such that the classifier outputs high entropy probabilities, and the VQ-VAE decoder can now reconstruct the new perturbed image. The aim is for the new image to contain as little of the confounding signal as possible, providing a way to augment images to remove confounders intentionally. After the augmentation is done, we can train a new, final classifier on the augmented data and produce a debiased classifier.

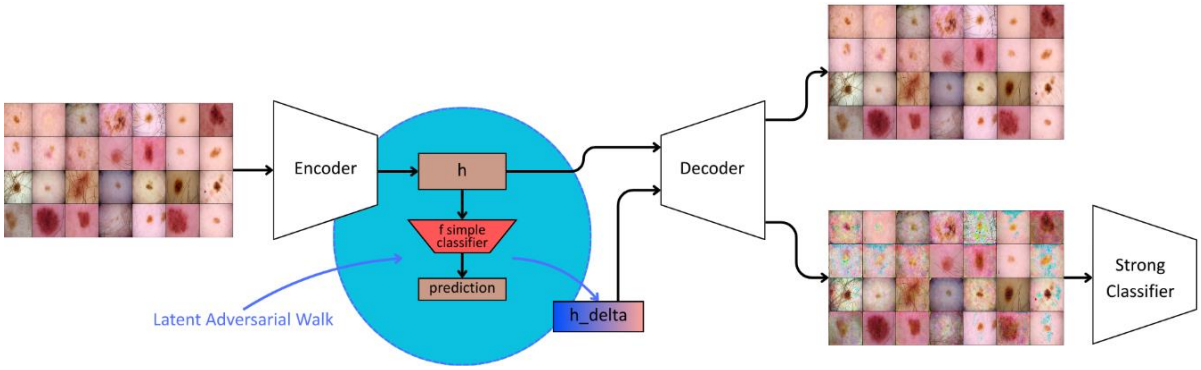

Fig. 16 The LAD architecture illustrated with real world skin cancer examples from the following datasets. Alpha was set to 0.1 denoting heavy perturbations.

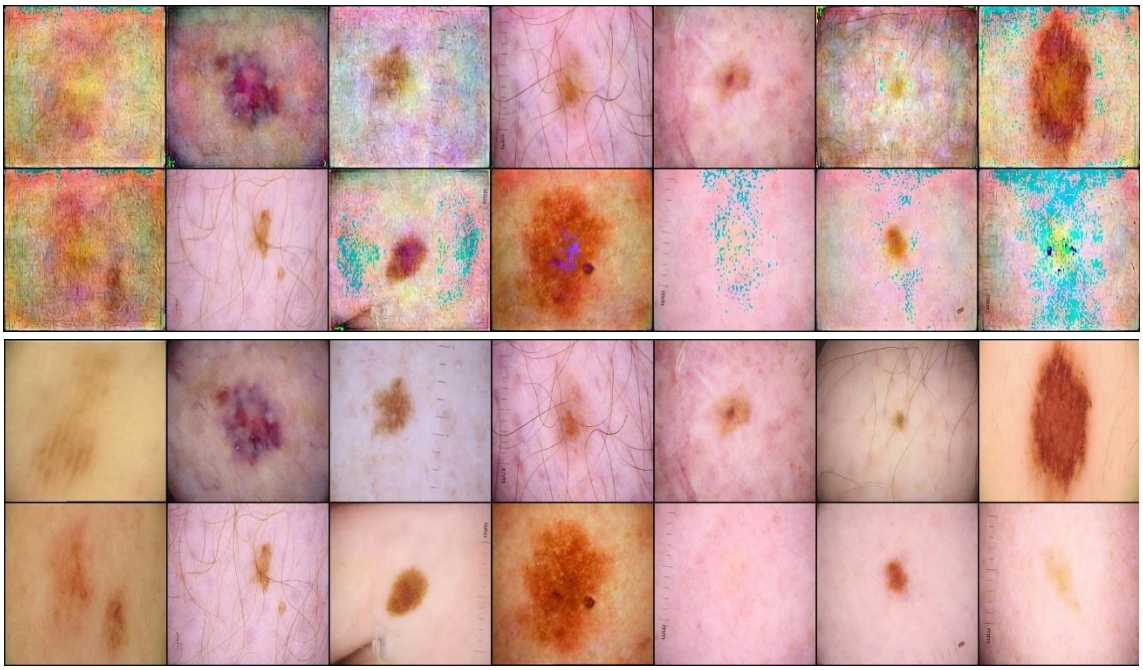

Fig. 17 Post-adversarial-walk images with alpha = 0.1 (heavy perturbations). Bottom images were taken before the walk, and the upper images were taken after the walk.

4.2 Dataset and evaluation

For our train dataset, we wanted to collect data that included skin tone labels along with diagnosis labels (benign-malignant). Due to limitations in the availability of such datasets, we decided to fuse two datasets, ISIC2020 [118], [119] and PAD-UEFS [120] into one whole dataset. While ISIC2020 covers the majority of our dataset, containing more than 40.000 images, it lacks skin tone labeling. PAD-UEFS is used to compensate for this unavailability by providing instances with skin tone labels.

4.2.1 ISIC2020 Dataset

The International Skin Imaging Collaboration (ISIC) is a joint effort between academia and industry that leverages digital skin imaging to advance early detection and ultimately reduce mortality from skin cancer. The ISIC archive comprises a range of datasets suitable for skin lesion classification. The ISIC2020 dataset was created for the 2020 skin lesion classification Challenge and is a publicly accessible dataset, consisting of 33.126 training images and 10.982 test images. The samples were extracted from 2.056 unique patients that are identified by a unique patient identifier. Images consist of contributions made from Memorial Sloan Kettering Cancer Center (11,108), The University of Queensland Diamantina Institute, The University of Queensland, Dermatology Research Centre (8,449), Department of Dermatology, Hospital Clínic de Barcelona (7,311), ViDIR Group, Department of Dermatology, Medical University of Vienna (4,374), Sydney Melanoma Diagnostic Center at Royal Prince Alfred Hospital and Pascale Guitera (1,884) [121]. All malignant diagnoses were confirmed via histopathology, and benign diagnoses were confirmed with expert agreement, longitudinal follow-up, or histopathology. Metadata for each image describes patient age at time of image capture, biological sex, general anatomic site of the lesion, anonymous patient identification number, benign or malignant category, and the specific diagnosis if one was available based on an acceptable ground truth confirmation method. The dataset contains 584 confirmed melanoma instances [121]. While there is no label for skin tone inside the metadata, a skin tone labelling algorithm using the Fitzpatrick skin type metric [122], estimates that $\sim 90\%$ of the dataset belongs to skin type I-IV.

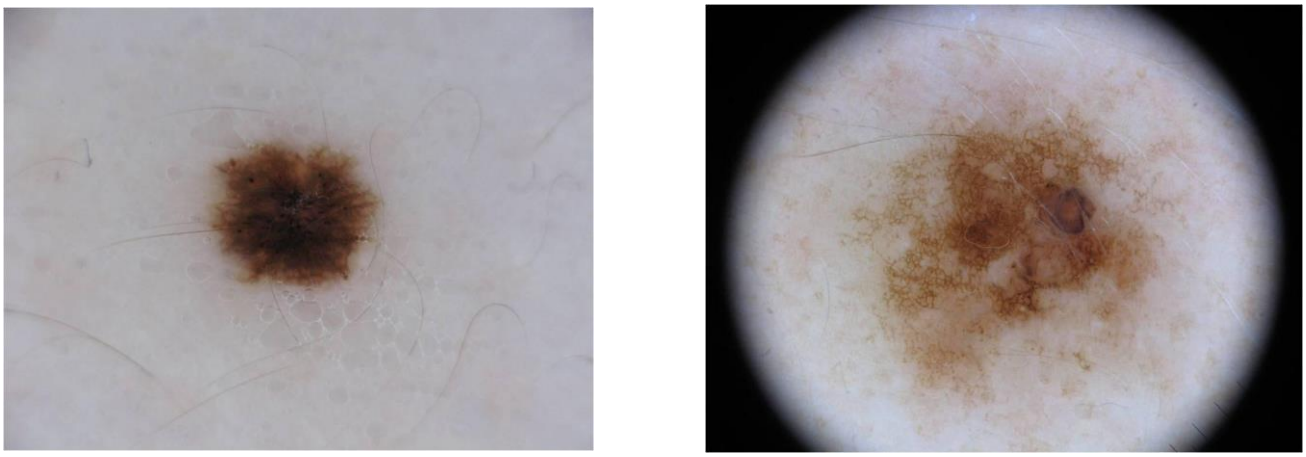

Fig. 18 Two Images from the ISIC2020 dataset.

4.2.2 PAD-UEFS Dataset

The PAD-UFES dataset was created by the Federal University of Espírito Santo (UFES) in Brazil as an openly accessible resource to support research in computer-aided skin cancer diagnosis. It was designed to provide clinical and dermoscopic images collected in real-world conditions using smartphone cameras, to facilitate robust skin lesion classification methods. The dataset includes 2,298 skin lesion samples from 1,373 unique patients, contributed by two Brazilian cancer care institutions: Santa Casa de Misericórdia de Vitória and the Federal University of Espírito Santo Hospital. A total of 658 malignant lesions were confirmed through histopathological examination, while benign cases were validated by dermatologist consensus. Metadata for each sample includes patient age, sex, lesion site, diagnostic category, parents' background, fitzpatrick skin type and several fields describing the lesion like itch (True/False), bleed etc. [123]. In this study, the dataset was used to incorporate more diverse skin tones into the ISIC2020 dataset.

Fig. 19 PAD-EUFS dataset images

4.2.3 Evaluation metrics

Evaluation metrics are essential for assessing the performance of our model, as they provide computable measures to compare predictions against ground truth labels. The most commonly used metrics include accuracy, which measures the proportion of correctly classified instances and is useful when classes are balanced; confusion matrix, which is a simple way of 2x2 array of ground truth labels and predictions; precision, which indicates the ratio of correctly predicted positive cases to all predicted positives and is crucial when false positives carry high costs; recall, which measures the proportion of actual positives correctly identified and is important when missing positive cases is critical; and the F1-score, which is the harmonic mean of precision and recall, offering a balanced metric when both false positives and false negatives need to be considered[124], [125].

These metrics can be computed with the following formulas:

If we assume that TP is a positive prediction that was correct, TN is a negative prediction was correct, FP is a positive prediction that was incorrect, and FN is a negative prediction that was incorrect:

Table 1 Confusion Matrix

	Positive (1)	*Negative (0)*
Positive (1)	True Positive (TP)	False Positive (FP)
Negative (0)	False Negative (FN)	True Negative (TN)

$$ACCURACY = \frac{TP + TN}{TP + FP + TN + FN}$$

$$PRECISION = \frac{TP}{TP + FP}$$

$$RECALL = \frac{TP}{TP + FN}$$

$$F1\ SCORE = \frac{2(PRECISION \cdot RECALL)}{PRECISION + RECALL}$$

On top of these vital metrics, we also add fairness metrics discussed in chapter 3.3.2. These metrics are crucial for model inference because they highlight different aspects of model behavior under various conditions, ensuring that evaluation is not biased toward a single perspective such as overall accuracy, receiving a comprehensive understanding of model reliability, robustness, and suitability for real-world applications.

5 Experiments

Our experiments are performed on the aforementioned dataset fusion context. We define the minority group as the protected group in the experiments. Two settings were tested on models: one with a few samples of protected group lesions and one with a more balanced dataset. We call the first setting “Fully biased” and the second “semi-biased”. The reason we test our model in these two different settings is that, like Darlaw L. et al [103], we do not assume that any training data is free from bias which is often the case in real-world situations. Our goal is to find a way to mitigate the confounder from the data rather than relying on a small amount of bias-free data. In the following sections, we discuss implementation details in 5.1, taking closer look to environment, model settings, dataset manipulation and preprocessing, and lastly, we also review the produced results in section 5.2. PyTorch code implementation exists on the author’s github [40].

5.1 Implementation Details

5.1.1 Environment settings

All following models have been trained and executed using PyTorch, on a machine with an RTX 4090 24GB VRAM GPU, 128GB RAM, NVMe M.2 storage and an Intel Xeon W-2235 CPU at 3.80GHz. For both settings, as our strong final classifier, we used transfer learning to fine tune some of the most common image classification models, including ResNet18, ResNet50, ResNet152, DenseNet121, DenseNet201, EfficientNetB0, and EfficientNetB5. We deployed these different model versions to observe how the model’s size would impact performance.

5.1.2 Model settings

For the VQ-VAE model, we used the following implementation [126] with hyper-parameters: 512 hidden layer size, 2 residual layers with size 32 each, embedding dimension of 64 with 2056 embeddings, 128 batch size, 0.35 commitment cost in the loss function and a learning rate of 0.01 with learning rate scheduler bringing the learning down to 0.0001. The Adam optimizer without amsgrad was used as the model’s optimizer along with MSE loss. The VQ-VAE was trained until it scored satisfactory results and acceptable codebook usage, specifically after an estimated 900 epochs, VQ-VAE scored $\sim 70\%$ codebook usage.

Due to the large size of our image dataset, the simple classifier deviates from the original seen in [103] and instead of just one linear layer with size 100 we use four layers with relu activation of 1024, 512, 128, and 2, respectively, with batch normalization and dropout rate with possibility 0.2 and batch size of 128. Additionally, the classifier has been trained with the “Reduce learning rate on Plateau” learning rate scheduler, which

reduces the learning when there is no major improvement in the model along with the Adam optimizer instead of SGD optimizer for 50 epochs.

For the adversarial walk, we used 2 steps and for the alpha hyper-parameter, settings were tested on 0.005, 0.01, 0.085 and 0.1 to evaluate the impact of low and high image perturbations.

For the transfer learning of models, we used the default best weights and the following settings in all models: 5 epochs for the classifier head of the model, early stopping with $patience = 15$ at 100 epochs maximum, cross entropy loss, AdamW optimizer with learning rate of 0.0005 and learning rate scheduler with $patience = 2$ and a batch size of 32. Models include ResNet20, ResNet50, ResNet150, Dense121, DenseNet201, EfficientNetB0 and EfficientNetB5.

5.1.3 Pre-processing

Before the training phase of our model, some pre-processing practices were applied to make training procedure easier for both human interpretation and model performance. Since the images are extracted from different sources, PyTorch requires all inputs have the same size. For this reason, we decided to resize our images to 256x256 to balance training time and visual quality. The metadata of our datasets were also merged to match the ISIC format as the majority of images are already using this format. Specifically, we cross-referenced similar attributes from the two datasets and matched the corresponding information. Redundant attributes like diagnosis and target, were also dropped to make metadata structure easier to understand. From the PAD-UEFS dataset, we filtered all Fitzpatrick instances of type 4-6 (the ones what were less present in the ISIC dataset) and inserted them into one unified dataset with ISIC's metadata structure and were assigned the $protected = 1$ attribute.

All image augmentations were pre-computed once and then saved to the training set to avoid generating augmentations every time we run the model. It is worth noting that the augmentations where done after the train/test split of the dataset to avoid intersection between splits that would otherwise overfit our model to the training data. New augmentations retain the original image's metadata. The following data augmentations were also used: random horizontal flip with $\text{probability (p)} = 0.5$, random vertical flip with $\text{p} = 0.5$, color jitter with $\text{brightness} = 0.1, \text{constrast} = 0.1, \text{saturation} = 0.1$ and $\text{hue} = 0.05$, random affine with $\text{p} = 0.5$ and $\text{scale} = [0.98,1.02]$, random perspective with $\text{p} = 0.5$ and $\text{distortion scale} = 0.2$, random rotation with $\text{degrees} = [0,20]$, gaussian blur with $\text{p} = 0.5$ and random erasing with $\text{p} = 1$. To compensate for the black edges that cropping or random perspective leave on the image, the inpainting technique TELEA was used from the OpenCV library [127], [128]. It uses a mask composed of a dilated kernel to detect the regions that need to be inpainted. Regions that are not zero inside the mask are inpainted. The algorithm performs a weighted

average on the neighboring pixels and replaces the center pixel. We use a kernel of 3x3 dilation and a grayscale mask of [0,0] to try and inpaint the black corners left by the augmentations.

5.2 Results

In the following subsections, we present the study's results, stating each model's performance and highlighting better outcomes. We then compare them to conclude on the impact of LAD in lesion classification and find which is the most optimal model to implement LAD on in order to fairly classify skin lesion data.

5.2.1 ResNet Results

The ResNet architecture has achieved decent results with the introduction of the LAD mechanism on the fully biased setting. An improvement of $\sim 15\%$ in F1 score, AP and DI can be observed in ResNet20 with the use of $alpha = 0.085$. Other minor improvements in recall of class 0 precision of class 1 and EO should also be considered. Overall, the model seems to have better performance on the protected demographic due to AP and DI being higher than the default model while also retaining an overall accuracy of 82.25% which is only 0.35% lower than the default. We can also observe that higher alpha values see more meaningful changes than lower values. On the other hand, the more complex ResNet50 and ResNet150 tend to perform better with lower alpha values. Although they did not manage to significantly deviate from the default performance, a small improvement in fairness can be observed.

Table 2 Full bias setting results of three ResNet architectures, red denotes best value in column and blue indicates the second best. Default in alpha signifies that the default architecture was used instead of the lad mechanism.

FULL BIAS

Model	ALPHA	EO	DI	AP	ACC	F1 | 0	F1 | 1	PR | 0	PR | 1	REC | 0	REC | 1
RESNET20	_	96.77%	68.48%	36.23%	82.60%	87.23%	72.71%	92.20%	79.38%	60.03%	96.80%
	0.1	97.92%	84.23%	53.70%	81.98%	72.83%	86.52%	87.20%	79.98%	62.53%	94.22%
	0.01	97.03%	71.01%	40.47%	82.18%	72.86%	86.74%	88.49%	79.85%	61.93%	94.93%
	0.085	96.43%	81.13%	52.04%	82.25%	73.32%	86.70%	87.41%	80.25%	63.17%	94.28%
	0.005	99.01%	73.97%	42.91%	82.21%	73.21%	86.68%	87.49%	80.18%	62.94%	94.34%
RESNET50	_	98.56%	70.21%	36.69%	84.13%	74.87%	88.40%	96.37%	80.15%	61.21%	98.55%
	0.085	96.77%	69.30%	39.62%	81.76%	72.69%	86.31%	86.22%	80.02%	62.82%	93.68%
	0.01	93.15%	79.14%	46.13%	82.25%	75.29%	86.51%	83.08%	82.30%	68.83%	91.18%
	0.1	92.10%	66.50%	41.63%	80.31%	71.54%	84.98%	81.13%	79.99%	63.98%	90.64%
	0.005	98.53%	75.68%	44.30%	83.32%	75.44%	87.37%	87.46%	81.60%	66.32%	94.02%
RESNET150	_	98.59%	72.78%	40.05%	84.60%	75.84%	88.69%	96.17%	80.71%	62.61%	98.43%
	0.085	94.85%	73.43%	41.26%	79.08%	66.18%	84.86%	88.11%	76.35%	53.00%	95.50%
	0.01	93.45%	72.55%	46.82%	82.10%	73.71%	86.43%	85.14%	80.82%	64.99%	92.86%
	0.1	95.40%	74.13%	41.84%	79.81%	67.43%	85.37%	89.42%	76.87%	54.12%	95.97%
	0.005	94.81%	72.43%	44.46%	83.09%	74.59%	87.33%	88.90%	80.84%	64.24%	94.95%

For the semi-biased dataset, we see a slight increase in default model accuracy, while the LAD accuracy falling behind by $\sim 2-4\%$. Fairness metrics EO and DI fall behind on ResNet20 but perform better on the more complex models ResNet50 and ResNet150. Specifically, a $\sim 20\%$ increase in EO and AP for ResNet150 can be observed as well as some small increases in fairness of ResNet50. Overall higher alpha values seem to have a better effect on fairness in this setting for all three models. Other metrics like F1 score recall and precision do not greatly deviate from the default in the best performing LAD settings.

Table 3 Semi biased setting results of three ResNet architectures, red denotes best value in column and blue indicates the second best. Default in alpha signifies that the default architecture was used instead of the lad mechanism.

Semi-Biased

MODEL	ALPHA	EO	DI	AP	ACC	F1 | 0	F1 | 1	PR | 0	PR | 1	REC | 0	REC | 1
RESNET20	_	79.74	87.82	61.54	86.26	81.1	89.21	86.52	86.12	76.31	92.52
	0.085	58.49	55.06	71.46	81.88	74.45	85.96	81.73	81.95	68.35	90.38
	0.01	31.98%	54.12%	54.44%	82.15%	77.34%	85.28%	75.88%	86.35%	78.85%	86.35%
	0.1	62.43%	55.92%	73.64%	81.78%	73.90%	86.00%	82.68%	81.37%	66.81%	91.19%
	0.005	51.68%	71.04%	56.07%	83.10%	78.16%	86.22%	78.02%	86.31%	78.30%	86.12%
RESNET50	_	86.33%	91.05%	49.63%	86.54%	80.43%	89.75%	91.76%	84.29%	71.59%	95.95%
	0.085	79%	93.48%	45.04%	80.06%	70.12%	85.03%	83.21%	78.81%	60.59%	92.32%
	0.01	61.20%	90.44%	47.65%	83.54%	77.62%	86.98%	81.72%	84.51%	73.91%	89.59%
	0.1	68.30%	92.47%	51.37%	80.42%	72.97%	84.65%	78.16%	81.57%	68.42%	87.97%
	0.005	76.45%	98.04%	49.59%	84.24%	77.90%	87.76%	84.98%	83.88%	71.91%	92.00%
RESNET150	_	74.10%	90.22%	54.95%	85.80%	80.02%	88.98%	87.60%	84.93%	73.63%	93.44%
	0.085	99.83%	91.83%	75.41%	80.48%	72.93%	84.74%	78.53%	81.46%	68.07%	88.29%
	0.01	41.72%	59.37%	59.56%	82.45%	78.59%	85.12%	74.28%	88.70%	83.43%	81.82%
	0.1	97.15%	90.53%	72.99%	82.48%	75.51%	86.36%	82.05%	82.69%	69.94%	90.37%
	0.005	74.11%	89.28%	59.32%	84.09%	79.18%	87.12%	80.03%	86.56%	78.35%	87.70%

5.2.2 DenseNet Results

The DenseNet architecture in the full biased setting has achieved good overall performance. It is the only model in the test results to surpass the accuracy of the default model rather than hindering it to increase fairness. DenseNet121 with $alpha = 0.1$ scored better DI and AP and a slightly lower EO score, while simultaneously increasing almost all other metrics (except PR|0 and REC|1). DenseNet201 DI and AP scores also prove that the model is fairer than the default with only a small decrease in accuracy. Both models seem to surpass their corresponding default architecture and are performing better on higher alpha values, as shown by row $alpha = 0.085, alpha = 0.1$.

Table 4 Full bias settings tested on two DenseNet models. Red denotes best value, and blue indicates the second best. Default in alpha signifies that the default architecture was used instead of the lad mechanism.

FULL BIAS

MODEL	ALPHA	EO	DI	AP	ACC	F1 | 0	F1 | 1	PR | 0	PR | 1	REC | 0	REC | 1
DENSENET121	DEFAULT	98.28%	70.11%	36.66%	83.96%	74.56%	88.29%	96.22%	80.00%	60.86%	98.50%
	0.085	91.53%	84.21%	65.28%	83.03%	77.60%	86.34%	79.16%	85.32%	76.11%	87.11%
	0.01	93.30%	67.08%	40.15%	81.35%	72.37%	85.92%	84.57%	80.04%	63.24%	92.74%
	0.1	96.23%	84.71%	56.36%	85.32%	78.38%	88.88%	90.86%	83.02%	68.92%	95.64%
DENSENET201	DEFAULT	99.36%	69.32%	34.91%	83.78%	74.22%	88.16%	96.06%	79.83%	60.48%	98.44%
	0.085	95.14%	75.48%	46.98%	82.31%	73.21%	86.80%	88.21%	80.09%	73.21%	86.80%
	0.01	84.71%	76.13%	36.37%	81.18%	72.76%	85.62%	82.50%	80.60%	65.07%	91.31%
	0.1	96.29%	75.63%	45.05%	82.22%	72.43%	86.88%	90.31%	79.40%	60.46%	95.92%

In the semi-biased setting, both DenseNet architectures demonstrate notable improvements in fairness compared to the default baseline. While DenseNet121 achieved slightly better balance across subgroups, its fairness metrics and overall accuracy show small declines relative to the default. On the other hand, the more complex DenseNet201 scored major improvements in fairness, with over a 20% increase in EO and DI compared to the default setting. This fairness boost, however, decreased most other performance metrics, such as precision, recall, F1, and AP by 1-10%. Overall, the results suggest that while both architectures benefit from LAD perturbations, DenseNet201 achieves a more favorable fairness–performance trade-off, highlighting the role of model capacity in mitigating bias.

Table 5 Semi biased setting results of the two DenseNet architectures, red denotes best value in column and blue indicates the second best. Default in alpha signifies that the default architecture was used instead of the lad mechanism.

Semi-Biased

MODEL	ALPHA	EO	DI	AP	ACC	F1 | 0	F1 | 1	PR | 0	PR | 1	REC | 0	REC | 1
DENSENET121	DEFAULT	67.72%	80.90%	57.17%	87.41%	82.72%	90.10%	88.05%	87.07%	77.99%	93.34%
	0.085	46.37%	56.50%	62.19%	84.33%	79.91%	87.16%	79.14%	87.70%	80.69%	86.62%
	0.01	52.62%	67.84%	58.95%	84.95%	80.89%	87.58%	79.35%	88.71%	82.50%	86.49%
	0.1	37.09%	49.14%	62.98%	82.91%	79.10%	85.55%	74.97%	88.93%	83.70%	82.42%
DENSENET201	DEFAULT	47.00%	61.70%	57.00%	87.77%	84.09%	90.07%	84.53%	89.78%	83.65%	90.37%
	0.085	69.03%	78.86%	60.84%	83.01%	76.61%	86.67%	81.82%	83.63%	72.03%	89.93%
	0.01	28.53%	51.62%	54.30%	83.67%	80.00%	86.20%	75.89%	89.55%	84.59%	83.09%
	0.1	63.42%	75.93%	58.85%	82.43%	75.76%	86.23%	81.12%	83.11%	71.06%	89.59%

5.2.3 EfficientNet Results

EfficientNet models are achieving similar results with the pervious models in the full biased setting. However, compared to the other models, there is a significant drop in accuracy for the more complicated model B5. The simpler B0 model bests B5 scores in every metric, possibly hinting that larger EfficientNet models struggle with LAD perturbations. EfficientNet B0 also marks decent results compared to the default model with increased fairness metrics, specifically a $\sim 15\%$ increase in DI and AP while decreasing overall accuracy by less than 2%. Additionally, we observe that EfficientNet performs better with lower alpha values, like $alpha = 0.01$ and $alpha = 0.005$.

Table 6 Full bias settings tested on two EfficientNet models. Red denotes the best value, and blue indicates the second best. Default in alpha signifies that the default architecture was used instead of the lad mechanism.

FULL BIAS

MODEL	ALPHA	EO	DI	AP	ACC	F1 | 0	F1 | 1	PR | 0	PR | 1	REC | 0	REC | 1
EFFICIENTNET B5	DEFAULT	94.23%	68.34%	39.52%	81.18%	71.45%	85.97%	86.29%	79.27%	60.97%	93.90%
	0.085	87.94%	69.73%	41.95%	71.64%	54.84%	79.33%	71.24%	71.77%	44.58%	88.68%
	0.01	87.36%	69.80%	52.19%	75.32%	66.04%	80.62%	70.47%	77.83%	62.14%	83.61%
	0.005	86.35%	69.56%	50.56%	76.98%	67.15%	82.28%	74.80%	77.98%	60.92%	87.09%
EFFICIENTNET B0	DEFAULT	97.10%	74.53%	44.11%	84.56%	76.34%	88.54%	93.52%	81.31%	64.49%	97.19%
	0.085	91.83%	84.91%	59.48%	80.38%	70.56%	85.29%	83.94%	79.00%	60.86%	92.67%
	0.01	97.91%	87.36%	60.24%	82.88%	76.14%	86.64%	82.41%	83.10%	70.76%	90.50%
	0.005	97.93%	78.59%	48.31%	81.21%	73.14%	85.55%	81.64%	81.01%	66.25%	90.62%

In the Semi-biased setting, the two EfficientNet models show great disparities, hinting that more complex architectures even when scaled evenly in height width and depth are considerably hindering their performance. EfficientNet B5 performs similar to the full bias setting in terms of overall accuracy, scoring $\sim 10\%$ lower than the default model. Fairness metrics do not see any major improvements other than a 10% increase in DI. B0 model on the other hand, scores more than 80% in all fairness metrics while being subject to only a 4% decrease in accuracy and other performance metrics, further reinforcing the stated assumption. Overall both models seem to perform better on larger alpha values like 0.1 and 0.085.

Table 7 Semi biased setting tested on two EfficientNet models. Red denotes the best value, and blue indicates the second best. Default in alpha signifies that the default architecture was used instead of the lad mechanism.

Semi-Biased

MODEL	ALPHA	EO	DI	AP	ACC	F1 | 0	F1 | 1	PR | 0	PR | 1	REC | 0	REC | 1
EFFICIENTNET B5	DEFAULT	86.66%	70.62%	80.76%	87.59%	84.15%	89.80%	82.99%	90.61%	85.35%	89.00%
	0.085	85.62%	81.08%	76.01%	76.00%	68.01%	80.79%	70.08%	79.38%	66.05%	82.25%
	0.01	76.66%	75.18%	75.25%	79.09%	74.10%	82.47%	71.04%	84.95%	77.43%	80.14%
	0.005	78.98%	75.76%	75.76%	79.43%	74.60%	82.72%	71.31%	85.40%	78.21%	80.20%
	0.1	87.92%	82.16%	76.36%	75.58%	66.86%	80.67%	70.27%	78.45%	63.76%	83.03%
EFFICIENTNET B0	DEFAULT	80.87%	81.57%	65.97%	88.56%	84.27%	91.01%	89.88%	87.88%	79.31%	94.31%
	0.085	95.06%	75.57%	92.19%	82.28%	75.97%	85.97%	79.77%	83.64%	72.51%	88.43%
	0.01	60.54%	62.08%	70.08%	83.79%	79.78%	86.47%	76.95%	88.65%	82.83%	84.39%
	0.005	52.50%	58.71%	66.37%	84.22%	80.39%	86.80%	77.29%	89.21%	83.75%	84.52%
	0.1	94.40%	86.63%	83.35%	81.38%	73.49%	85.05%	78.84%	82.02%	69.29%	88.31%

5.2.4 Model Comparison

Having evaluated model performance across both semi-biased and fully biased settings, we next compare architectures under their best-performing LAD configurations. In the full bias setting (Figs. 13–16), three architectures emerge as clear leaders: EfficientNet B0, DenseNet121, and ResNet20. All three achieve strong fairness outcomes, surpassing 50% in Accuracy Parity (AP), 80% in Disparate Impact (DI), and 85% in Equal Opportunity (EO), indicating considerable mitigation of bias relative to other baselines. Among these (table 8), DenseNet121 and EfficientNet B0 slightly outperform the best-performing ResNet20 variant, achieving not only higher fairness metrics but also stronger overall accuracy. Notably, DenseNet121 surpasses EfficientNet B0 across most metrics, positioning it as the most consistent architecture under high-bias conditions. These results highlight the importance of balancing model depth and size when addressing the fairness–accuracy trade-off. DenseNet121, being shallower and more compact than DenseNet201, achieves stronger fairness and accuracy scores despite its smaller capacity. In contrast, deeper variants such as DenseNet201 may introduce unnecessary complexity that does not translate into better debiasing performance.

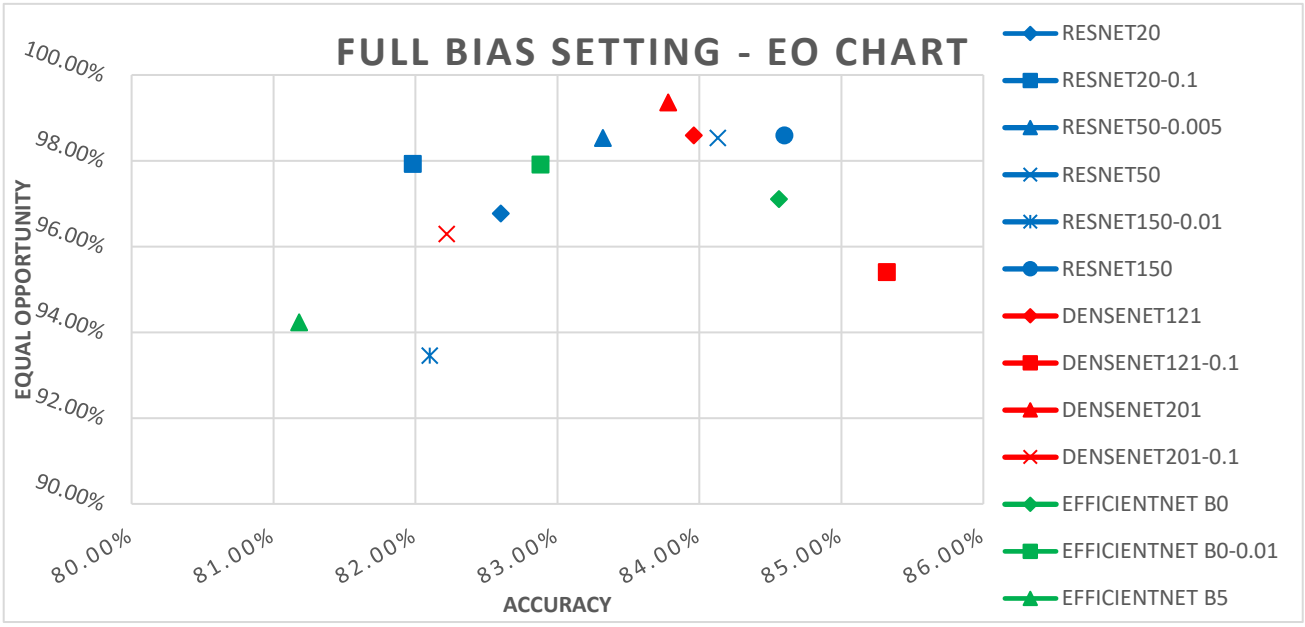

Fig. 20 Equal Opportunity versus Accuracy in the fully biased setting. ResNet models are colored blue, DenseNet red, and EfficientNet green. LAD models include the value of alpha inside their name (i.e. resnet20-0.1). EfficientNet B5-0.01 is not included inside the graph because the value was too low and it would make the graph unreadable.

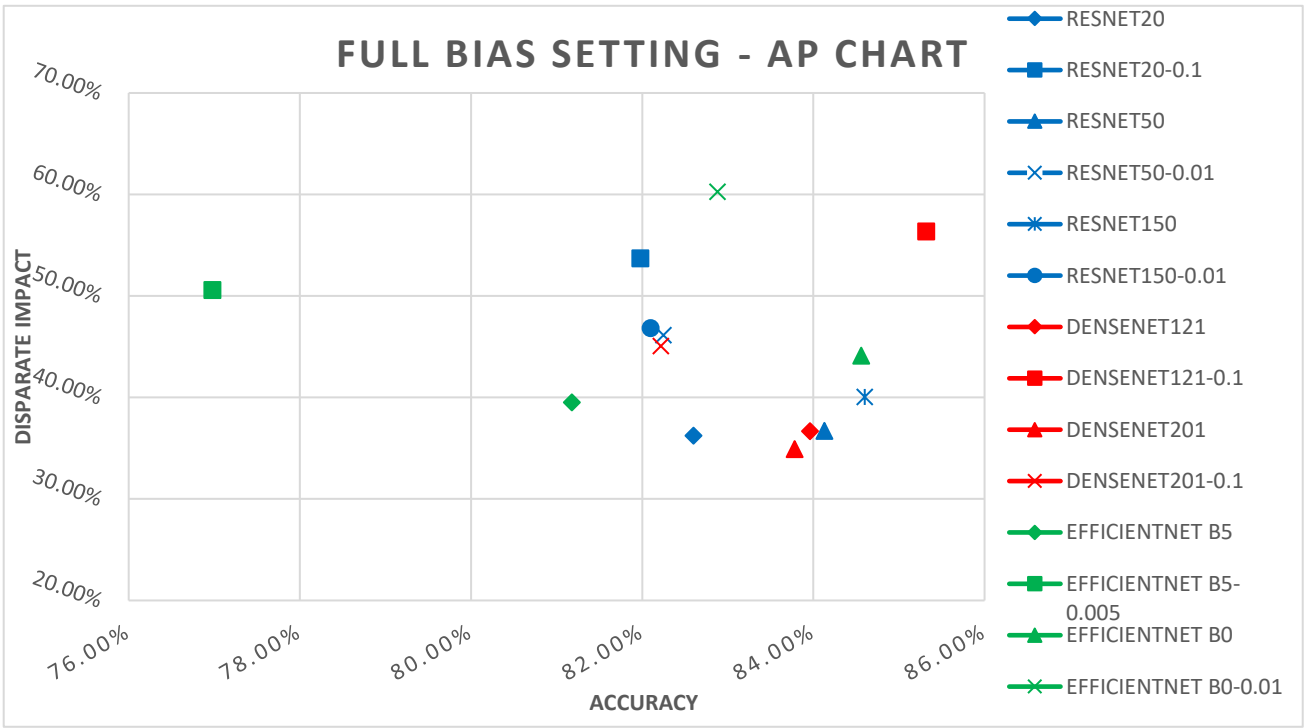

Fig. 21 Accuracy Parity versus Accuracy in the semi biased setting. ResNet models are colored blue, DenseNet red, and EfficientNet green. LAD models include the value of alpha inside their name (i.e. resnet20-0.1)

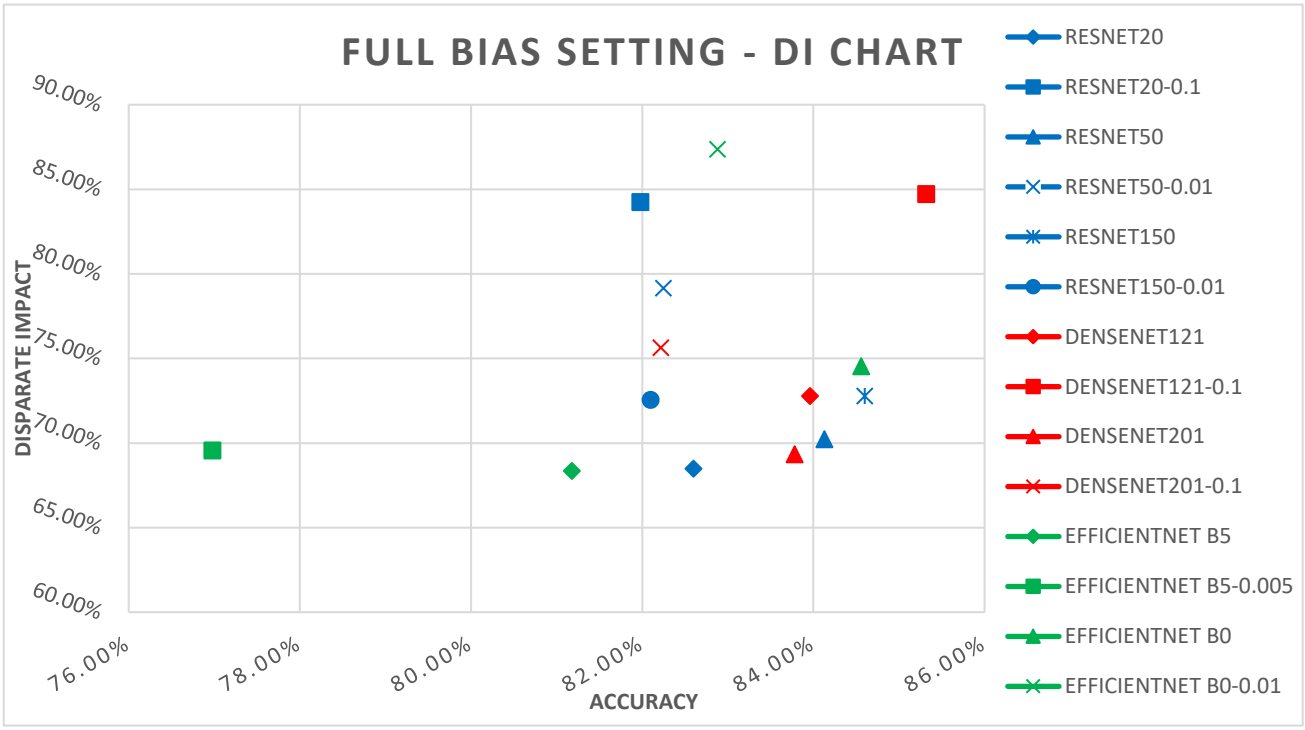

Fig. 22 Disparate Impact versus Accuracy in the fully biased setting. ResNet models are colored blue, DenseNet red, and EfficientNet green. LAD models include the value of alpha inside their name (i.e. resnet20-0.1)

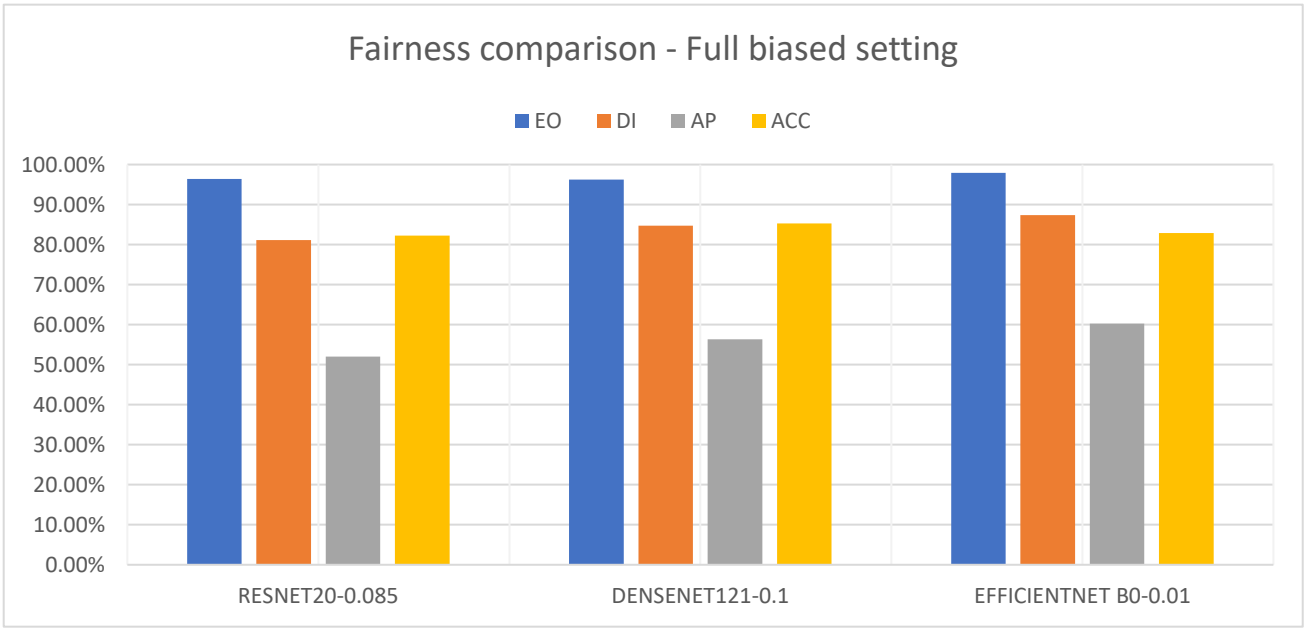

Fig. 23 Comparison of all fairness metrics and accuracy between the best performing models in the fully biased setting.

In the semi biased setting (Fig. 17-20) we observe different models scoring better results. For the DI chart, ResNet50 with $alpha = 0.005$ stands out as the highest DI score followed by the other ResNet models and the two EfficientNet B0-B5 with $alpha = 0.1$. The DenseNet architecture scored decent results in accuracy but did not perform as well as the other architectures on the DI metric.

For the EO chart, we again see both ResNet and EfficientNet achieving the highest outcomes, with ResNet150 being at the top followed by the B0 architecture, while B1 scored the better accuracy while keeping EO relatively high. DenseNet architecture underperformed on the EO chart with results under 70%.

For the AP chart, ResNet and EfficientNet are emerging as the top models specifically B0 and ResNet150 with outcomes over 80% and 70% respectively. DenseNet, while not scoring as high results as the other two architectures, the accuracy was higher than the other models.

Comparing the three best models in Fig. 20, we see that both EfficientNet and ResNet architectures are offering similar performance with EfficientNet surpassing ResNet on AP score, while ResNet achieved higher DI and EO scores. These results show that each model comes with its own trade-offs and that both models can be applied in the present challenge effectively.

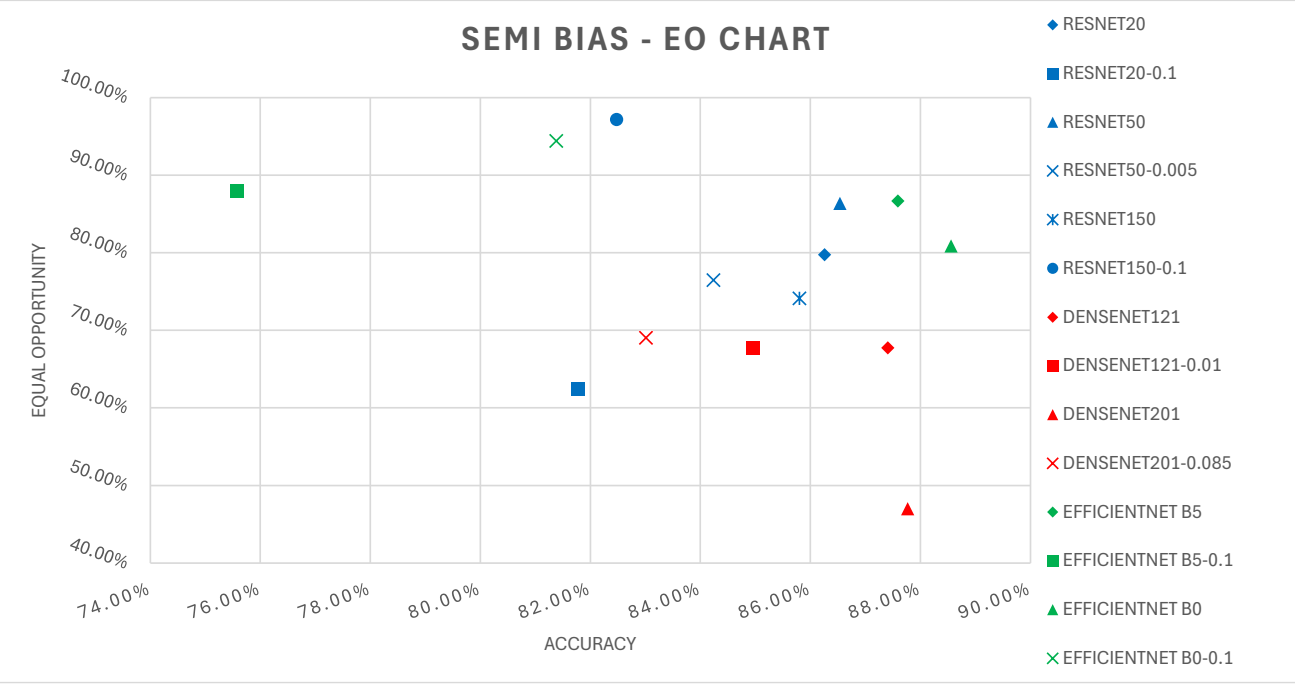

Fig. 24 Accuracy Parity versus Accuracy in the semi biased setting. ResNet models are colored blue, DenseNet red, and EfficientNet green. LAD models include the value of alpha inside their name (i.e. resnet20-0.1)

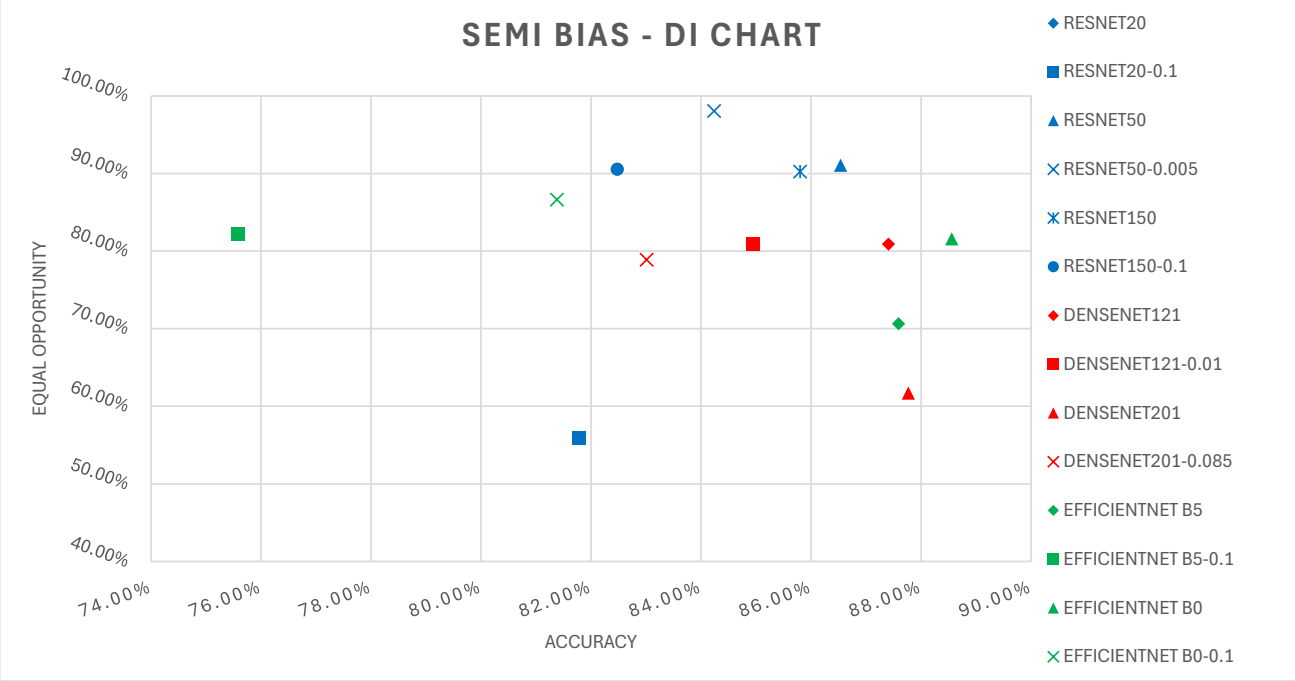

Fig. 25 Disparate Impact versus Accuracy in the semi biased setting. ResNet models are colored blue, DenseNet red, and EfficientNet green. LAD models include the value of alpha inside their name (i.e. resnet20-0.1). ResNet20 is not present in the graph because of the low EO value.

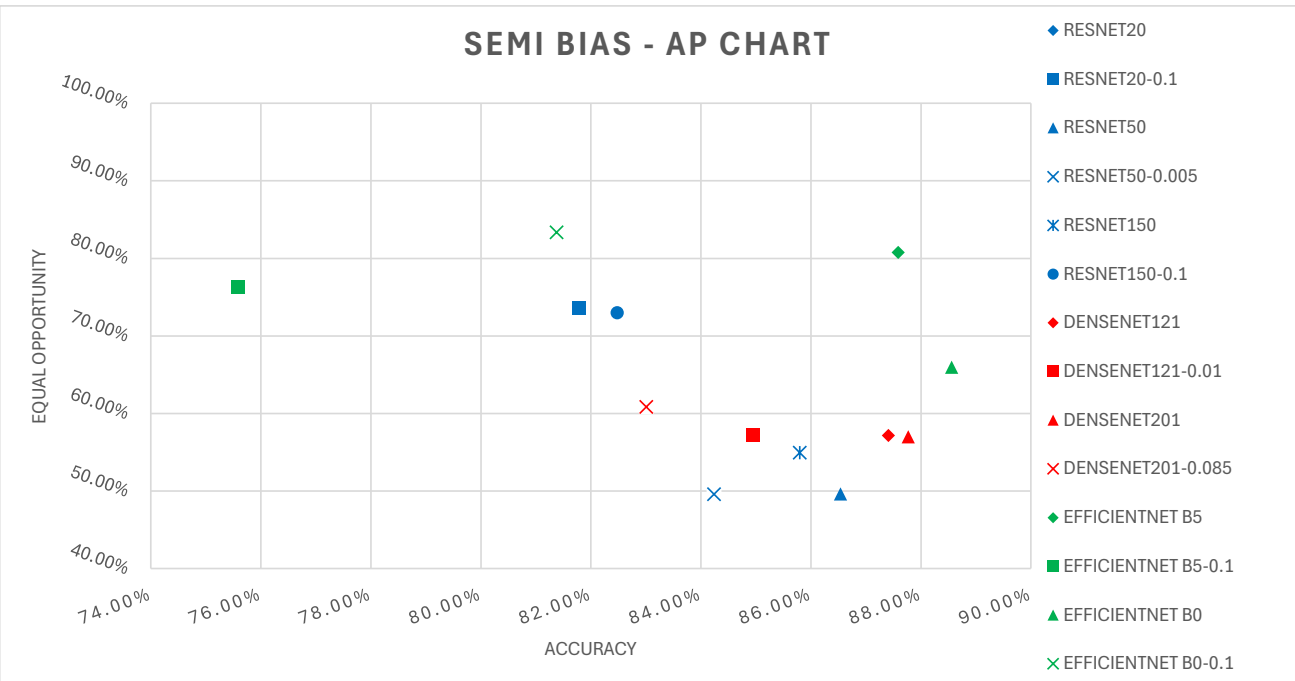

Fig. 26 Accuracy Parity versus Accuracy in the semi biased setting. ResNet models are colored blue, DenseNet red, and EfficientNet green. LAD models include the value of alpha inside their name (i.e. resnet20-0.1)

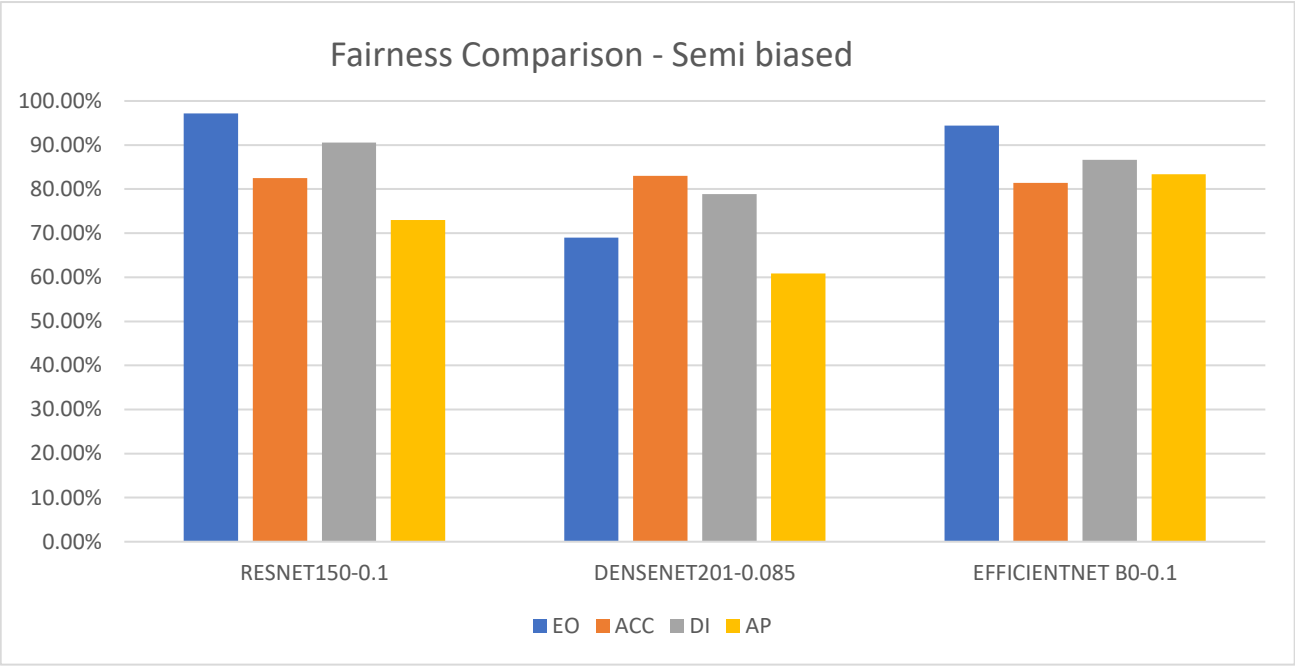

Fig. 27 Comparison of all fairness metrics and accuracy of the best performing models using the LAD augmentations in the semi biased setting.

6 Conclusion & Future Work

This study examined the issue of bias in AI models, with a specific focus on skin cancer detection in medical imaging. Building upon the theoretical foundation of machine and deep learning, addressing critical healthcare challenges, issues, prior work done to mitigate them, and the Latent Adversarial Debiasing framework designed to combat confounding features present in the data. The experimental setup evaluated the performance of state-of-the-art CNN architectures and provided evidence of the framework's effectiveness.

We conclude with two different important findings. First, the use of LAD has evidently proven to consistently increase all fairness-related metrics such as EO, DI and AP without hindering the overall classification accuracy. This suggests that the LAD framework can be used to reduce demographic group disparities while maintaining crucial clinical performance, striving for a fairer classification environment. Secondly, the experiments show that the LAD framework generally favored smaller capacity models like ResNet20, DenseNet121, and EfficientNetB0. While larger models have shown a better overall accuracy, the majority of them were not subjected to major fairness improvements. Therefore, there exists no need to align model complexity to the present bias mitigation technique.

As deep learning approaches are being increasingly integrated in the field of healthcare, present biases that hurt model accuracy for a particular demographic subgroup can often be the deciding factor for a patient's life. Therefore, this study highlights the importance of integrating and developing fairness-aware models for medical imaging and many other applications. While LAD shows promising results, there exists no bias mitigation technique that truly eliminates systemic biases due to data incompleteness or imbalance.

Future work should expand the current study in three main ways. First, the construction of a larger and diverse skin lesion dataset with exact metadata about lesion condition, diagnosis, and skin tone, is crucial to bias reduction, as we believe there needs to exist a combination of pre-processing, in-processing, and post-processing methods to significantly mitigate biases. Second, is the finding of a lightweight enough model to fit on resource constraint clinical infrastructure or even mobile phones. The model needs to produce satisfactory results and must be constructed alongside experienced hospital physicians to verify the usefulness of the model. Third, the generalization of LAD to other clinical imaging tasks like radiology, pathology or ophthalmology should be explored to further utilize the framework's bias mitigation mechanism.

References

[1] "Cancer." Accessed: Aug. 08, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer

[2] "Cancer Statistics - NCI." Accessed: Aug. 29, 2025. [Online]. Available: https://www.cancer.gov/about-cancer/understanding/statistics

[3] G. P. Guy, S. R. Machlin, D. U. Ekwueme, and K. R. Yabroff, "Prevalence and costs of skin cancer treatment in the U.S., 2002-2006 and 2007-2011," *Am J Prev Med*, vol. 48, no. 2, pp. 183–187, Feb. 2015, doi: 10.1016/J.AMEPRE.2014.08.036.

[4] N. Alipour, T. Burke, and J. Courtney, "Skin Type Diversity in Skin Lesion Datasets: A Review," *Curr Dermatol Rep*, vol. 13, no. 3, p. 198, Sep. 2024, doi: 10.1007/S13671-024-00440-0.

[5] H. Sheikh, C. Prins, and E. Schrijvers, "Artificial Intelligence: Definition and Background," pp. 15–41, 2023, doi: 10.1007/978-3-031-21448-6_2.

[6] "IEEE Xplore Full-Text PDF:" Accessed: Jul. 28, 2025. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8259629

[7] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol Rev*, vol. 65, no. 6, pp. 386–408, Nov. 1958, doi: 10.1037/H0042519.

[8] "Perceptron | Encyclopedia of Computer Science." Accessed: Jul. 18, 2025. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/1074100.1074686

[9] I. El Naqa and M. J. Murphy, "What Are Machine and Deep Learning?," *Machine and Deep Learning in Oncology, Medical Physics and Radiology, Second Edition*, pp. 3–15, Jan. 2022, doi: 10.1007/978-3-030-83047-2_1.

[10] J. M. Amigo, "Data Mining, Machine Learning, Deep Learning, Chemometrics Definitions, Common Points and Trends (Spoiler Alert: VALIDATE your models!)," *Brazilian Journal of Analytical Chemistry*, vol. 8, no. 32, pp. 45–61, 2021, doi: 10.30744/brjac.2179-3425.AR-38-2021.

[11] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, Jul. 2018, doi: 10.1109/ICCUBEA.2018.8697857.

[12] J. M. Amigo, "Data Mining, Machine Learning, Deep Learning, Chemometrics Definitions, Common Points and Trends (Spoiler Alert: VALIDATE your models!)," *Brazilian Journal of Analytical Chemistry*, vol. 8, no. 32, pp. 45–61, 2021, doi: 10.30744/brjac.2179-3425.AR-38-2021.

[13] B. Jenkins and A. Tanguay, "Handbook of neural computing and neural networks," 1995, *MIT Press*.

[14] Y. chen Wu and J. wen Feng, "Development and Application of Artificial Neural Network," *Wirel Pers Commun*, vol. 102, no. 2, pp. 1645–1656, Sep. 2018, doi: 10.1007/S11277-017-5224-X/FIGURES/2.

[15] Y. X. C. Z. ZH Luo, "The study of convergence of CMAC learning process," *Acta Automatic Sinica*, vol. 23, no. 4, pp. 455–461, 1997.

[16] Y. chen Wu and J. wen Feng, "Development and Application of Artificial Neural Network," *Wirel Pers Commun*, vol. 102, no. 2, pp. 1645–1656, Sep. 2018, doi: 10.1007/S11277-017-5224-X/FIGURES/2.

[17] J. L. Balcázar, R. Gavaldà, and H. T. Siegelmann, "Computational power of neural networks: A characterization in terms of Kolmogorov complexity," *IEEE Trans Inf Theory*, vol. 43, no. 4, pp. 1175–1183, 1997, doi: 10.1109/18.605580.

[18] Z. Zhang, "Artificial Neural Network," *Multivariate Time Series Analysis in Climate and Environmental Research*, pp. 1–35, 2018, doi: 10.1007/978-3-319-67340-0_1.

[19] M. Goyal, R. Goyal, P. Venkatappa Reddy, and B. Lall, "Activation functions," *Studies in Computational Intelligence*, vol. 865, pp. 1–30, 2020, doi: 10.1007/978-3-030-31760-7_1/FIGURES/15.

[20] S. Sharma, S. Sharma, A. A.-T. D. Sci, and undefined 2017, "Activation functions in neural networks," *academia.edu*, Accessed: Jul. 26, 2025. [Online]. Available: https://www.academia.edu/download/89662883/310-316_Tesma412_IJEAST.pdf

[21] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989, doi: 10.1016/0893-6080(89)90020-8.

[22] T. Chakraborty and U. Kumar, “Loss Function,” *Encyclopedia of Earth Sciences Series*, vol. 2020, pp. 1–6, 2023, doi: 10.1007/978-3-030-26050-7_187-2.
[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533A0;KWRD=SCIENCE.
[24] A. C. Aitken, “IV.—On Least Squares and Linear Combination of Observations,” *Proceedings of the Royal Society of Edinburgh*, vol. 55, pp. 42–48, 1936, doi: 10.1017/S0370164600014346.
[25] L. Fahrmeir, T. Kneib, S. Lang, and B. D. Marx, “Regression Models,” *Regression*, pp. 23–84, 2021, doi: 10.1007/978-3-662-63882-8_2.
[26] P. Christoffersen and K. Jacobs, “The importance of the loss function in option valuation,” *J financ econ*, vol. 72, no. 2, pp. 291–318, May 2004, doi: 10.1016/J.JFINECO.2003.02.001.
[27] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, Accessed: Aug. 06, 2025. [Online]. Available: https://arxiv.org/pdf/1412.6980
[28] “Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude | CiNii Research.” Accessed: Aug. 06, 2025. [Online]. Available: https://cir.nii.ac.jp/crid/1370017282431050757
[29] S. ichi Amari, “Backpropagation and stochastic gradient descent method,” *Neurocomputing*, vol. 5, no. 4–5, pp. 185–196, Jun. 1993, doi: 10.1016/0925-2312(93)90006-O.
[30] D. Biskup, “A state-of-the-art review on scheduling with learning effects,” *Eur J Oper Res*, vol. 188, no. 2, pp. 315–329, Jul. 2008, doi: 10.1016/J.EJOR.2007.05.040.
[31] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” Jun. 01, 2015, *PMLR*. Accessed: Aug. 07, 2025. [Online]. Available: https://proceedings.mlr.press/v37/ioffe15.html
[32] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization”.
[33] Z. Liu, Z. Xu, J. Jin, Z. Shen, and T. Darrell, “Dropout Reduces Underfitting,” Jul. 03, 2023, *PMLR*. Accessed: Aug. 07, 2025. [Online]. Available: https://proceedings.mlr.press/v202/liu23aq.html
[34] L. Prechelt, “Early Stopping - But When?,” pp. 55–69, 1998, doi: 10.1007/3-540-49430-8_3.
[35] S. García, J. Luengo, and F. Herrera, “Intelligent Systems Reference Library 72 Data Preprocessing in Data Mining”, Accessed: Aug. 06, 2025. [Online]. Available: http://www.springer.com/series/8578
[36] H. Yang, “Data Preprocessing-Chapter 3”, Accessed: Aug. 06, 2025. [Online]. Available: http://cose-stor.sfsu.edu/~huiyang
[37] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, “Big data preprocessing: methods and prospects,” *Big Data Analytics 2016 1:1*, vol. 1, no. 1, pp. 1–22, Nov. 2016, doi: 10.1186/S41044-016-0014-0.
[38] K. Maharana, S. Mondal, and B. Nemade, “A review: Data pre-processing and data augmentation techniques,” *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/J.GLTP.2022.04.020.
[39] A. Parvat, J. Chavan, S. Kadam, S. Dev, and V. Pathak, “A survey of deep-learning frameworks,” *Proceedings of the International Conference on Inventive Systems and Control, ICISC 2017*, Oct. 2017, doi: 10.1109/ICISC.2017.8068684.
[40] “Adversarial de-biasing on skin cancer data VIA Lantent Adversarial De-biasing.” Accessed: Aug. 16, 2025. [Online]. Available: https://github.com/Panaghs01/Thesis
[41] C. Aliferis, … G. S. and machine learning in health care and, and undefined 2024, “Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI,” *Springer*, pp. 477–524, 2024, doi: 10.1007/978-3-031-39355-6_10.
[42] H. Jabbar, R. K.-C. science, communication and, and undefined 2015, “Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study),” *academia.edu*, Accessed: Aug. 11, 2025. [Online]. Available: https://www.academia.edu/download/37902828/017.pdf
[43] K. Mavrogiorgos, A. Kiourtis, A. Mavrogiorgou, A. Menychtas, and D. Kyriazis, “Bias in Machine Learning: A Literature Review,” *Applied Sciences 2024, Vol. 14, Page 8860*, vol. 14, no. 19, p. 8860, Oct. 2024, doi: 10.3390/APP14198860.

[44] M. A. Sufian, L. Alsadder, W. Hamzi, S. Zaman, A. S. M. S. Sagar, and B. Hamzi, “Mitigating Algorithmic Bias in AI-Driven Cardiovascular Imaging for Fairer Diagnostics,” *Diagnostics*, vol. 14, no. 23, p. 2675, Dec. 2024, doi: 10.3390/DIAGNOSTICS14232675.
[45] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol Cybern*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251/METRICS.
[46] 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. 2278–2323, 1998, doi: 10.1109/5.726791.
[47] K. O’shea and R. Nash, “An Introduction to Convolutional Neural Networks”.
[48] D. Dai, “An Introduction of CNN: Models and Training on Neural Network Models,” *Proceedings - 2021 International Conference on Big Data, Artificial Intelligence and Risk Management, ICBAR 2021*, pp. 135–138, 2021, doi: 10.1109/ICBAR55169.2021.00037.
[49] J. Wu, “Introduction to Convolutional Neural Networks,” 2017.
[50] “Deep Learning: An Overview of Convolutional Neural Network(CNN),” 2020.
[51] D. Bank, N. Koenigstein, and R. Giryes, “Autoencoders,” *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook, Third Edition*, pp. 353–374, Jan. 2023, doi: 10.1007/978-3-031-24628-9_16/FIGURES/8.
[52] E. Oja, “Simplified neuron model as a principal component analyzer,” *J Math Biol*, vol. 15, no. 3, pp. 267–273, 1982, doi: 10.1007/BF00275687/METRICS.
[53] K. P. F.R.S., “LIII. On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, Nov. 1901, doi: 10.1080/14786440109462720.
[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Internal Representations by Error Propagation,” Sep. 1985, doi: 10.21236/ADA164453.
[55] M. A. Kramer, “Autoassociative neural networks,” *Comput Chem Eng*, vol. 16, no. 4, pp. 313–328, Apr. 1992, doi: 10.1016/0098-1354(92)80051-A.
[56] L. Theis, W. Shi, A. Cunningham, and F. Huszár, “Lossy Image Compression with Compressive Autoencoders,” *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Mar. 2017, Accessed: Aug. 09, 2025. [Online]. Available: https://arxiv.org/pdf/1703.00395
[57] U. Michelucci, “An Introduction to Autoencoders,” Jan. 2022, Accessed: Aug. 09, 2025. [Online]. Available: https://arxiv.org/pdf/2201.03898
[58] A. Creswell, K. Arulkumaran, and A. A. Bharath, “On denoising autoencoders trained to minimise binary cross-entropy,” Aug. 2017, Accessed: Aug. 09, 2025. [Online]. Available: https://arxiv.org/pdf/1708.08487
[59] Y. Zhang, “A Better Autoencoder for Image: Convolutional Autoencoder”.
[60] Y.-W. Chen and L. C. Jain, “Intelligent Systems Reference Library 171 Deep Learning in Healthcare Paradigms and Applications”, Accessed: Aug. 14, 2025. [Online]. Available: http://www.springer.com/series/8578
[61] I. J. Goodfellow *et al.*, “Generative Adversarial Nets,” *Adv Neural Inf Process Syst*, vol. 27, 2014, Accessed: Aug. 11, 2025. [Online]. Available: http://www.github.com/goodfeli/adversarial
[62] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative Adversarial Networks: An Overview,” *IEEE Signal Process Mag*, vol. 35, no. 1, pp. 53–65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.
[63] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative Adversarial Networks: An Overview,” *IEEE Signal Process Mag*, vol. 35, no. 1, pp. 53–65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.
[64] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, Nov. 2015, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/1511.06434

[65] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” Nov. 2014, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/1411.1784
[66] W. Wang *et al.*, “Medical Image Classification Using Deep Learning,” 2020, doi: 10.1007/978-3-030-32606-7_3.
[67] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, “Transfer learning: a friendly introduction,” *J Big Data*, vol. 9, no. 1, pp. 1–19, Dec. 2022, doi: 10.1186/S40537-022-00652-W/FIGURES/6.
[68] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016. Accessed: Aug. 11, 2025. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/
[69] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” 2017. Accessed: Aug. 11, 2025. [Online]. Available: https://github.com/liuzhuang13/DenseNet.
[70] M. Tan and Q. V Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” May 24, 2019, *PMLR*. Accessed: Aug. 11, 2025. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html
[71] K. Shailaja, B. Seetharamulu, and M. A. Jabbar, “Machine Learning in Healthcare: A Review,” *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, pp. 910–914, Sep. 2018, doi: 10.1109/ICECA.2018.8474918.
[72] H. P. Chan, R. K. Samala, L. M. Hadjiiski, and C. Zhou, “Deep Learning in Medical Image Analysis,” *Adv Exp Med Biol*, vol. 1213, p. 3, 2020, doi: 10.1007/978-3-030-33128-3_1.
[73] S. C. B. Lo, H. P. Chan, J. S. Lin, H. Li, M. T. Freedman, and S. K. Mun, “Artificial convolution neural network for medical image pattern recognition,” *Neural Networks*, vol. 8, no. 7–8, pp. 1201–1214, Jan. 1995, doi: 10.1016/0893-6080(95)00061-5.
[74] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, “Early diagnosis of Alzheimer’s disease with deep learning,” *2014 IEEE 11th International Symposium on Biomedical Imaging, ISBI 2014*, pp. 1015–1018, Jul. 2014, doi: 10.1109/ISBI.2014.6868045.
[75] A. Esteva *et al.*, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature 2017 542:7639*, vol. 542, no. 7639, pp. 115–118, Jan. 2017, doi: 10.1038/nature21056.
[76] T. Pham, T. Tran, D. Phung, and S. Venkatesh, “DeepCare: A deep dynamic memory model for predictive medicine,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9652 LNAI, pp. 30–41, 2016, doi: 10.1007/978-3-319-31750-2_3/TABLES/2.
[77] R. Miotto, L. Li, B. Kidd, J. D.-S. reports, and undefined 2016, “Deep patient: an unsupervised representation to predict the future of patients from the electronic health records,” *nature.comR Miotto, L Li, BA Kidd, JT DudleyScientific reports, 2016•nature.com*, vol. 6, May 2016, doi: 10.1038/srep26094.
[78] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: review, opportunities and challenges,” *Brief Bioinform*, vol. 19, no. 6, p. 1236, May 2017, doi: 10.1093/BIB/BBX044.
[79] R. Miotto, L. Li, and J. T. Dudley, “Deep learning to predict patient future diseases from the electronic health records,” *SpringerR Miotto, L Li, JT DudleyEuropean conference on information retrieval, 2016•Springer*, vol. 9626, pp. 768–774, 2016, doi: 10.1007/978-3-319-30671-1_66.
[80] R. Fakoor, F. Ladhak, … A. N.-P. of the, and undefined 2013, “Using deep learning to enhance cancer diagnosis and classification,” *admis.tongji.edu.cnR Fakoor, F Ladhak, A Nazi, M HuberProceedings of the international conference on machine learning, 2013•admis.tongji.edu.cn*, 2013, Accessed: Aug. 14, 2025. [Online]. Available: https://admis.tongji.edu.cn/_upload/article/files/a1/28/1a8d7ac94b05860e2d1fe6c3bf2a/dffeeaae-5d26-4e84-9b60-42bb4b47ee41.pdf
[81] J. Lyons *et al.*, “Predicting backbone Cα angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network,” *Wiley Online LibraryJ Lyons, A Dehzangi, R Heffernan, A*

Sharma, K Paliwal, A Sattar, Y Zhou, Y YangJournal of computational chemistry, 2014•Wiley Online Library, vol. 35, no. 28, pp. 2040–2046, Oct. 2014, doi: 10.1002/JCC.23718.

[82] H. A. Haenssle *et al.*, "Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists," *Annals of Oncology*, vol. 29, no. 8, pp. 1836–1842, Aug. 2018, doi: 10.1093/ANNONC/MDY166.

[83] A. G. C. Pacheco and R. A. Krohling, "The impact of patient clinical information on automated skin cancer detection," *Comput Biol Med*, vol. 116, p. 103545, Jan. 2020, doi: 10.1016/J.COMPBIOMED.2019.103545.

[84] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "A review of deep learning based methods for medical image multi-organ segmentation," *Physica Medica*, vol. 85, pp. 107–122, May 2021, doi: 10.1016/J.EJMP.2021.05.003.

[85] K. Suzuki, "Overview of deep learning in medical imaging," *Radiol Phys Technol*, vol. 10, no. 3, pp. 257–273, Sep. 2017, doi: 10.1007/S12194-017-0406-5/FIGURES/5.

[86] N. Nayak, H. Chang, A. Borowsky, P. Spellman, and B. Parvin, "Classification of tumor histopathology via sparse feature learning," *Proceedings - International Symposium on Biomedical Imaging*, pp. 410–413, 2013, doi: 10.1109/ISBI.2013.6556499.

[87] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," *2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*, pp. 844–848, 2014, doi: 10.1109/ICARCV.2014.7064414.

[88] B. Shi *et al.*, "Prediction of Occult Invasive Disease in Ductal Carcinoma in Situ Using Deep Learning Features," *Journal of the American College of Radiology*, vol. 15, no. 3, pp. 527–534, Mar. 2018, doi: 10.1016/J.JACR.2017.11.036.

[89] M. Talo, "Automated classification of histopathology images using transfer learning," *Artif Intell Med*, vol. 101, p. 101743, Nov. 2019, doi: 10.1016/J.ARTMED.2019.101743.

[90] B. R. Solunke and S. R. Gengaje, "A Review on Traditional and Deep Learning based Object Detection Methods," *2023 International Conference on Emerging Smart Computing and Informatics, ESCI 2023*, 2023, doi: 10.1109/ESCI56872.2023.10099639.

[91] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6517–6525, Dec. 2016, doi: 10.1109/CVPR.2017.690.

[92] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2/FIGURES/5.

[93] R. Girshick, "Fast R-CNN," 2015. Accessed: Aug. 16, 2025. [Online]. Available: https://github.com/rbgirshick/

[94] C. Albuquerque, R. Henriques, and M. Castelli, "Deep learning-based object detection algorithms in medical imaging: Systematic review," *Heliyon*, vol. 11, no. 1, p. e41137, Jan. 2025, doi: 10.1016/J.HELIYON.2024.E41137.

[95] Z. Guo, X. Li, H. Huang, N. Guo, and Q. Li, "Deep learning-based image segmentation on multimodal medical imaging," *IEEE Trans Radiat Plasma Med Sci*, vol. 3, no. 2, pp. 162–169, Mar. 2019, doi: 10.1109/TRPMS.2018.2890359.

[96] A. Rajkomar, J. Dean, and I. Kohane, "Machine Learning in Medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, Apr. 2019, doi: 10.1056/NEJMRA1814259/SUPPL_FILE/NEJMRA1814259_DISCLOSURES.PDF.

[97] R. C. Deo, "Machine learning in medicine," *Circulation*, vol. 132, no. 20, pp. 1920–1930, Nov. 2015, doi: 10.1161/CIRCULATIONAHA.115.001593/ASSET/F82A0780-3EFF-4CE8-839D-5FF553B00121/ASSETS/GRAPHIC/1920FIG04.JPEG.

[98] X. Liu, L. Song, S. Liu, and Y. Zhang, "A Review of Deep-Learning-Based Medical Image Segmentation Methods," *Sustainability 2021, Vol. 13, Page 1224*, vol. 13, no. 3, p. 1224, Jan. 2021, doi: 10.3390/SU13031224.

[99] R. Radhika and R. Mahajan, “Medical Image Enhancement: A Review,” *Lecture Notes in Networks and Systems*, vol. 288, pp. 105–118, 2022, doi: 10.1007/978-981-16-5120-5_9/TABLES/3.
[100] D. Dhabliya, S. Singh Dari, A. Dhablia, N. Akhila, V. Khetani, and A. Professor, “Addressing Bias in Machine Learning Algorithms: Promoting Fairness and Ethical Design”, doi: 10.1051/e3sconf/202449102040.
[101] S. Swayamdipta *et al.*, “Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics,” *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 9275–9293, Sep. 2020, doi: 10.18653/v1/2020.emnlp-main.746.
[102] J. Nam, H. Cha, S. Ahn, J. Lee, and J. Shin, “Learning from Failure: De-biasing Classifier from Biased Classifier,” *Adv Neural Inf Process Syst*, vol. 33, pp. 20673–20684, 2020.
[103] L. Darlow, S. Jastrzębski, and A. Storkey, “Latent Adversarial Debiasing: Mitigating Collider Bias in Deep Neural Networks”.
[104] S. Gupta, M. De-Arteaga, and M. Lease, “Fairly Accurate: Fairness-aware Multi-group Target Detection in Online Discussion,” Jul. 2024, Accessed: Aug. 17, 2025. [Online]. Available: https://arxiv.org/pdf/2407.11933
[105] L. H. Nazer *et al.*, “Bias in artificial intelligence algorithms and recommendations for mitigation,” *PLOS Digital Health*, vol. 2, no. 6, p. e0000278, Jun. 2023, doi: 10.1371/JOURNAL.PDIG.0000278.
[106] R. Zemel, Y. (Ledell,) Wu, K. Swersky, T. Pitassi, and C. Dwork, “Learning Fair Representations,” 2013.
[107] P. Adler *et al.*, “Auditing Black-box Models for Indirect Influence”.
[108] A. J. D. Mahamadou and A. A. Trotsyuk, “Revisiting Technical Bias Mitigation Strategies,” *Annu Rev Biomed Data Sci*, vol. 8, no. Volume 8, 2025, pp. 287–303, Aug. 2025, doi: 10.1146/ANNUREV-BIODATASCI-103123-095737.
[109] Z. Wang *et al.*, “Towards Fairness in Visual Recognition: Effective Strategies for Bias Mitigation,” 2020. Accessed: Aug. 11, 2025. [Online]. Available: https://github.com/
[110] M. Benčević, M. Habijan, I. Galić, D. Babin, and A. Pižurica, “Understanding skin color bias in deep learning-based skin lesion segmentation,” *Comput Methods Programs Biomed*, vol. 245, p. 108044, Mar. 2024, doi: 10.1016/J.CMPB.2024.108044.
[111] T. Kalb, K. Kushibar, C. Cintas, K. Lekadir, O. Diaz, and R. Osuala, “Revisiting Skin Tone Fairness in Dermatological Lesion Classification,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14242 LNCS, pp. 246–255, 2023, doi: 10.1007/978-3-031-45249-9_24.
[112] P. J. Bevan and A. Atapour-Abarghouei, “Detecting Melanoma Fairly: Skin Tone Detection and Debiasing for Skin Lesion Classification,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13542 LNCS, pp. 1–11, 2022, doi: 10.1007/978-3-031-16852-9_1/TABLES/3.
[113] A. A. Abdulredah, M. A. Fadhel, L. Alzubaidi, Y. Duan, M. Kherallah, and F. Charfi, “Towards unbiased skin cancer classification using deep feature fusion,” *BMC Med Inform Decis Mak*, vol. 25, no. 1, pp. 1–22, Dec. 2025, doi: 10.1186/S12911-025-02889-W/FIGURES/16.
[114] A. Pundhir, S. Verma, and B. Raman, “Towards Ethical Dermatology: Mitigating Bias in Skin Condition Classification,” *Proceedings of the International Joint Conference on Neural Networks*, 2024, doi: 10.1109/IJCNN60899.2024.10650487.
[115] A. Bissoto, M. Fornaciali, E. Valle, and S. Avila, “(De)Constructing Bias on Skin Lesion Datasets,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2019-June, pp. 2766–2774, Apr. 2019, doi: 10.1109/CVPRW.2019.00335.
[116] C. Bellatreccia, “Bias Mitigation in Skin Disease Classification”.
[117] A. van den Oord DeepMind, O. Vinyals DeepMind, and K. Kavukcuoglu DeepMind, “Neural Discrete Representation Learning,” *Adv Neural Inf Process Syst*, vol. 30, 2017.
[118] “ISIC Challenge.” Accessed: Aug. 19, 2025. [Online]. Available: https://challenge.isic-archive.com/
[119] B. Cassidy, C. Kendrick, A. Brodzicki, J. Jaworek-Korjakowska, and M. H. Yap, “Analysis of the ISIC image datasets: Usage, benchmarks and recommendations,” *Med Image Anal*, vol. 75, p. 102305, Jan. 2022, doi: 10.1016/J.MEDIA.2021.102305.

[120] A. G. C. Pacheco *et al.*, “PAD-UFES-20: a skin lesion dataset composed of patient data and clinical images collected from smartphones,” vol. 1, 2020, doi: 10.17632/ZR7VGBCYR2.1.
[121] V. Rotemberg *et al.*, “A patient-centric dataset of images and metadata for identifying melanomas using clinical context,” *Sci Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1038/S41597-021-00815-Z.
[122] V. Gupta and V. K. Sharma, “Skin typing: Fitzpatrick grading and others,” *Clin Dermatol*, vol. 37, no. 5, pp. 430–436, Sep. 2019, doi: 10.1016/J.CLINDERMATOL.2019.07.010.
[123] A. G. C. Pacheco and R. A. Krohling, “The impact of patient clinical information on automated skin cancer detection,” *Comput Biol Med*, vol. 116, p. 103545, Jan. 2020, doi: 10.1016/J.COMPBIOMED.2019.103545.
[124] O. Rainio, J. Teuho, and R. Klén, “Evaluation metrics and statistical tests for machine learning,” *Sci Rep*, vol. 14, no. 1, pp. 1–14, Dec. 2024, doi: 10.1038/S41598-024-56706-X;SUBJMETA=117,531,639,705;KWRD=COMPUTER+SCIENCE,STATISTICS.
[125] G. Naidu, T. Zuva, and E. M. Sibanda, “A Review of Evaluation Metrics in Machine Learning Algorithms,” *Lecture Notes in Networks and Systems*, vol. 724 LNNS, pp. 15–25, 2023, doi: 10.1007/978-3-031-35314-7_2/FIGURES/3.
[126] “vq-vae.ipynb - Colab.” Accessed: Aug. 20, 2025. [Online]. Available: https://colab.research.google.com/github/zalandoresearch/pytorch-vq-vae/blob/master/vq-vae.ipynb#scrollTo=AV0X-P5m6HBM
[127] A. Telea, “An Image Inpainting Technique Based on the Fast Marching Method,” *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, Jan. 2004, doi: 10.1080/10867651.2004.10487596.
[128] “OpenCV: Image Inpainting.” Accessed: Aug. 29, 2025. [Online]. Available: https://docs.opencv.org/3.4/df/d3d/tutorial_py_inpainting.html
[129] “A typical artificial neural network implemented in an electronic nose... | Download Scientific Diagram.” Accessed: Aug. 29, 2025. [Online]. Available: https://www.researchgate.net/figure/A-typical-artificial-neural-network-implemented-in-an-electronic-nose-system_fig8_318634399
[130] “ResNet Architecture basic block.” Accessed: Aug. 29, 2025. [Online]. Available: https://towardsai.net/p/l/resnet-architecture-deep-learning-with-pytorch