Comparative Analysis of Deep Convolutional Neural Networks for Detecting Medical Image Deepfakes

Abdel Rahman Alsabbagh[1,2] Omar Al-Kadi[1]
[1]University of Jordan, Jordan
[2]King Abdullah University of Science and Technology, Saudi Arabia
abdelrahman.sabbagh@kaust.edu.sa o.alkadi@ju.edu.jo

Abstract

Generative Adversarial Networks (GANs) have exhibited noteworthy advancements across various applications, including medical imaging. While numerous state-of-the-art Deep Convolutional Neural Network (DCNN) architectures are renowned for their proficient feature extraction, this paper investigates their efficacy in the context of medical image deepfake detection. The primary objective is to effectively distinguish real from tampered or manipulated medical images by employing a comprehensive evaluation of 13 state-of-the-art DCNNs. Performance is assessed across diverse evaluation metrics, encompassing considerations of time efficiency and computational resource requirements. Our findings reveal that ResNet50V2 excels in precision and specificity, whereas DenseNet169 is distinguished by its accuracy, recall, and F1-score. We investigate the specific scenarios in which one model would be more favorable than another. Additionally, MobileNetV3Large offers competitive performance, emerging as the swiftest among the considered DCNN models while maintaining a relatively small parameter count. We also assess the latent space separability quality across the examined DCNNs, showing superiority in both the DenseNet and EfficientNet model families and entailing a higher understanding of medical image deepfakes. The experimental analysis in this research contributes valuable insights to the field of deepfake image detection in the medical imaging domain[1].

1. Introduction

In the realm of medical imaging, the advent of generative modeling marks a transformative era. Traditional data augmentation techniques, effective in many domains, encounter limitations when applied to medical images like Computed Tomography (CT) scans or Magnetic Resonance (MR) scans. Geometric transformations, such as random flipping, cropping, rotation, or translation, prove inadequate, failing to enhance neural network generalization capabilities beyond the initial training population and often resulting in the generation of highly correlated samples [1].

Recognizing these challenges, recent strides have been made in leveraging Generative Adversarial Networks (GANs) as a solution [2, 3]. GANs contribute by generating authentic-looking medical images, augmenting datasets, and positively impacting model accuracy. This not only simplifies data synthesis within the medical imaging domain but also offers a cost-effective alternative. However, the application of GANs introduces challenges, such as incorporating intentional manipulation, forgery, and tampering in the medical images, potentially giving rise to future apprehensions among clinicians considering the integration of AI in the field of medicine. Authenticating the generated images is crucial, given the potential consequences of misinterpretation regardless of the intent behind the application. To this end, this paper revolves around the utilization of state-of-the-art Deep Convolutional Neural Network (DCNN) architectures to discern between authentic and synthetic CT scan images, generated by the CT-GAN [4]. The hypothesis driving our investigation is rooted in the critical need for a reliable approach to authenticate medical images, especially in scenarios where the visual realism of GAN-generated images poses challenges to accurate diagnosis. Our contributions are:

(a) Conducting a set of extensive experiments on the most prominent DCNNs known to be used by the machine learning community for medical image deepfake detection tasks;

(b) Analyzing time complexity and model efficiency for a finer understanding of DCNNs in the medical domain;

(c) Exploring the embeddings of various DCNNs after training on the medical image deepfake detection task, and examining the latent space separability quality.

[1]TECHNICAL REPORT, UNIVERSITY OF JORDAN, ARTIFICIAL INTELLIGENCE DEPARTMENT, TR-01,23.

Through this paper, we present a comprehensive overview of the study and related work in section 2. Subsequently, we detail the dataset preprocessing, methodology, and results analysis in subsequent sections (3, 4, and 5), providing valuable insights into the potential of DCNNs in addressing authenticity challenges in medical imaging. Finally, section 6 further discuss and summarize our findings and conclusion.

2. Overview

The field of deepfake detection has experienced a surge in research activity, driven by milestones like the introduction of StyleGAN [5]. This Generative Adversarial Network (GAN) marked a transformative shift by excelling in the creation of highly realistic images, surpassing its predecessors. Researchers have responded to the challenge of detecting forgery in such images through a diverse array of approaches. These range from machine learning-based methods, such as tree-based methods [6] claiming enhanced interpretability, to the utilization of vanilla Convolutional Neural Networks (CNNs) [7].

The adoption of well-established or novel Deep Convolutional Neural Network (DCNN) architectures has been a prevalent theme in recent research efforts. For instance, [8] utilized VGG16 [9], MobileNetV2 [10], Xception [11], and InceptionV3 [12] to detect deepfakes on generic data, [13] employed a combination of ensemble learning techniques with CNNs, and [14] developed a lightweight deepfake detector using the successive subspace learning principle. Some researchers have explored hand-crafted features, considering biological signals within images [15].

Beyond image manipulation, the creation of deepfakes has extended to videos and audio, necessitating the application of Recurrent Neural Networks (RNNs) to model forgery in sequential data. [16] addressed audio deepfakes using a large margin cosine loss function and frequency masking, [17] used recurrent neural networks for video deepfake detection, and [18] and [19] developed methods to detect joint audio-visual deepfakes. While Transformers have revolutionized natural language processing [20] and extended to images [21], attention-based blocks have been introduced in various deepfake detection techniques. [22], for instance, utilized the Transformer to detect multimodal deepfakes on multiple scales, and [23] assembled a hybrid Transformer-CNN architecture to capture both local and global contexts within an image.

As the field of deepfake detection matures, it finds novel applications in the medical imaging domain, addressing unique challenges. So far, [24] employed convolutional reservoir networks, and [25] conducted a comparative study on a limited set of machine learning and deep learning architectures. Our work is inspired by this evolving landscape, with a specific focus on the capabilities of DCNNs. Unlike previous works, our concentration on DCNNs aims to discern differentiating features in medical images, providing a deeper understanding of their capabilities in capturing complex features within the medical imaging domain.

In the following sections, we delve into the methodologies employed, dissect encountered obstacles and vulnerabilities, and analyze the solutions proposed to overcome challenges in our exploration of DCNNs for detecting deepfakes in medical imaging. This comprehensive overview critically examines state-of-the-art approaches, laying the foundation for subsequent sections in our work.

3. Data Preprocessing

In this section, we will discuss both the datasets used and how we preprocessed the data before employing it in our models.

3.1. Datasets

In constructing a model capable of discerning between *real* and *fake* medical images, the utilization of diverse and representative datasets is imperative. For the *real* medical image dataset, we employed the Lung Image Database Consortium image collection (LIDC-IDRI), a comprehensive repository of lung CT scans encompassing data from a thousand distinct patients [26]. This dataset serves as a benchmark for authentic medical images, providing a diverse and well-established foundation for our model's understanding of genuine medical scans.

Conversely, to introduce the dimension of *fake* medical images, we leveraged tampered data generated by the CT-GAN. Notably, the CT-GAN utilized the LIDC-IDRI dataset during its training phase, making it an intriguing source for simulated or manipulated medical images. This dual dataset approach, incorporating both authentic and manipulated medical images, enables our model to learn discriminative features essential for distinguishing between *real* and *fake* instances. Through this balanced and diverse dataset selection, we aim to enhance the robustness and generalization capabilities of our deep learning model in the domain of medical image deepfake detection.

3.2. Preprocessing

Both datasets contain files in Digital Imaging and Communications in Medicine (DICOM) format, and it is a widely used standard for storing and transmitting medical images. The format includes not only the image itself but also the metadata of the scan such as the diagnosis of the scan and the patient's information. A medical scan folder contains a stack of 2D image slices in DICOM format that can be reconstructed into a 3D volume. The CT-GAN dataset has a total of a hundred scans, each scan is a series of 512×512 images, and the series ranges from $100 - 300$ slices long. In the metadata file provided by [4], there were

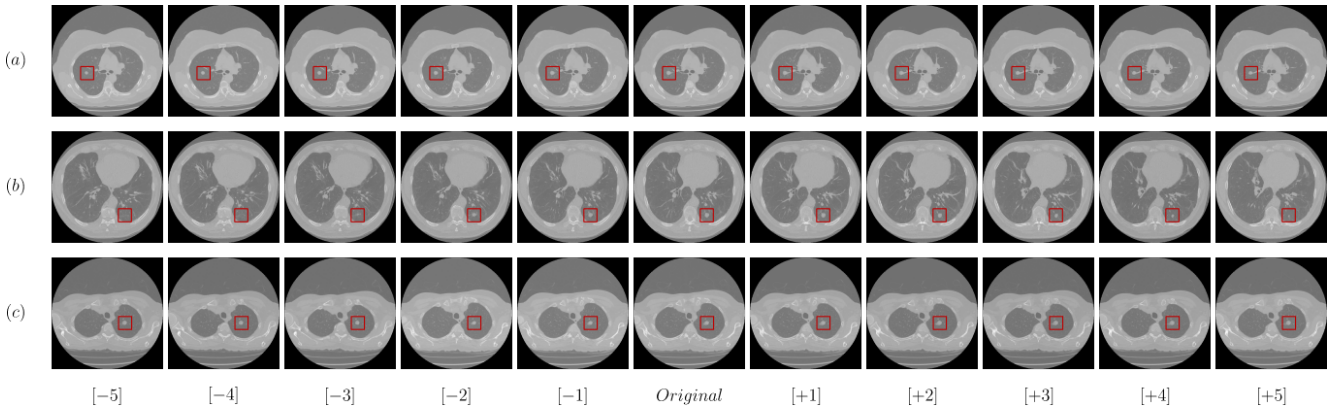

Figure 1. Sequential scan slices of three patients denoted as (a), (b), and (c) on each row, where the column *Original* denotes the label provided, and columns $[\pm i]$ denote the adjacent slices within a range of i of the *Original* slice. (a) shows that the tumor starts and ends exactly within the range $[-5, +5]$, (b) shows that the tumor starts and ends within the range $[-3, +4]$, and finally (c) shows that the tumor starts and ends out of the range $[-5, +5]$.

only 113 slices labeled as *false malignant* and *false benign*, i.e. *tampered / fake*. It appears to suffer from severe imbalance, but the CT-GAN model injects and removes nodules in a 3D volume and not only one slice of the whole scan. So in principle, the neighboring slices should also extend the nodule of a labeled slice, as this is how the 3D pix2pix network works [27].

To examine our preprocessing approach, we visualized the labeled slices and the adjacent ones within different ranges. Figure 1 shows three separate *False Malignant* scans with their tumor region with a range of $[-5, +5]$ of the original label. We can see that indeed the tumor does not show up only on the labeled slice, but also on the neighbor slices, which means in return that there are more than 113 tampered images. At this stage, there were two choices to get a hold of as much tampered image data as possible: a) visualize all labeled slices and their adjacent slices and select manually all of the slices that include a tumor at the region of interest, or b) automatically select the labeled slices and their adjacent slices within a range of $[-5, +5]$ of the labeled slice. The initial option is deemed impractical; therefore, the latter is chosen. However, it is not a strict rule that all slices within the range of $[-5, +5]$ of the labeled slice are consistently tampered, as illustrated in Figure 1. We acknowledge the potential for errors associated with this choice. Nevertheless, in a significant majority of cases, nearly all adjacent slices within the range $[-5, +5]$ do exhibit tampering. After this preprocessing step, we ended up having $1,243$ images for the *fake* class.

The LIDC-IDRI dataset contains scans of a thousand patients with precisely $244,385$ scan slices for all patients. To maintain a balanced overall dataset with a 50:50 *real:fake* ratio, we down-sampled the LIDC-IDRI dataset to have $1,243$ images, which was achieved by choosing slices at random, making the overall dataset that will be used in training a total of $2,486$ images.

4. Methodology

To solve the medical deepfake detection task, we experimented with a diverse set of DCNN architectures, these include ConvNeXtTiny [28], DenseNet121, DenseNet169, DenseNet201 [29], EfficientNetB4 [30], EfficientNetV2S [31], InceptionV3 [12], MobileNetV3Large [32], RegNetX040, RegNetY040 [33], ResNet50V2 [34], VGG19 [9], and Xception [11]. The rationale behind selecting these models particularly is detailed in Appendix A.

We modified the DCNNs by appending a global average pooling layer at the end, followed by a couple of fully connected layers as additional feature extractors tailored to the binary classification problem at hand and for the later latent space visualization. The full architecture is depicted in Figure 2.

All experiments were conducted using Google Colab's NVIDIA T4 GPU. We loaded the DCNNs directly from the `TensorFlow` [35] and initiated them with pre-trained weights from the ImageNet dataset [36], keeping the base models frozen while allowing the remaining architecture to be trainable. The input image size was fixed at 512×512 pixels, and we uniformly trained all models for 80 epochs, saving models with the lowest validation loss. A batch size of 8 was used, and we employed Binary Cross-Entropy (BCE) as the loss function as follows:

$$\text{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \quad (1)$$

where N is the number of samples in the dataset, y_i is the true binary label for the i-th sample, and $\hat{y}_i$ is the predicted

Table 1. Results of different Deep Convolutional Neural Networks (1 run)

Model	Accuracy	Precision	Recall/Sensitivity	Specificity	F1-Score	AUC
ConvNeXtTiny	0.9544	0.9444	0.9605	0.9490	0.9524	0.9968
DenseNet121	0.9678	0.9412	**0.9944**	0.9439	0.9671	**0.9988**
DenseNet169	**0.9786**	0.9669	0.9887	0.9694	**0.9777**	0.9985
DenseNet201	0.9571	0.9448	0.9661	0.9490	0.9553	0.9971
EfficientNetB4	0.9678	0.9661	0.9661	0.9694	0.9661	0.9980
EfficientNetV2S	0.9678	0.9609	0.9718	0.9643	0.9663	0.9980
InceptionV3	0.9517	0.9344	0.9661	0.9388	0.9500	0.9968
MobileNetV3Large	0.9598	0.9451	0.9718	0.9490	0.9583	0.9975
RegNetX040	0.9491	0.9593	0.9322	0.9643	0.9456	0.9949
RegNetY040	0.9544	0.9348	0.9718	0.9388	0.9529	0.9945
ResNet50V2	0.9732	**0.9883**	0.9548	**0.9898**	0.9713	0.9982
VGG19	0.9517	0.9492	0.9492	0.9541	0.9492	0.9964
Xception	0.9651	0.9659	0.9605	0.9694	0.9632	0.9977

Table 2. Results of top-performing Deep Convolutional Neural Networks (3 runs)

Model	Accuracy	Precision	Recall/Sensitivity	Specificity	F1-Score	AUC
DenseNet121	0.9705 ±0.0047	0.9547 ±0.0153	0.9849 ±0.0214	0.9592 ±0.0216	0.9694 ±0.0050	**0.9984 ±0.0004**
DenseNet169	**0.9732 ±0.0047**	0.9500 ±0.0147	**0.9962 ±0.0007**	0.9567 ±0.0180	**0.9738 ±0.0055**	0.9982 ±0.0003
EfficientNetB4	0.9642 ±0.0040	0.9507 ±0.0137	0.9755 ±0.0086	0.9541 ±0.0135	0.9623 ±0.0040	0.9940 ±0.0062
EfficientNetV2S	0.9615 ±0.0086	0.9502 ±0.0109	0.9699 ±0.0086	0.9541 ±0.0102	0.9599 ±0.0090	0.9976 ±0.0005
ResNet50V2	0.9723 ±0.0016	**0.9746 ±0.0288**	0.9680 ±0.0279	**0.9762 ±0.0281**	0.9710 ±0.0001	0.9981 ±0.0001
Xception	0.9418 ±0.0295	0.9246 ±0.0538	0.9579 ±0.0029	0.9274 ±0.0560	0.9404 ±0.0284	0.9936 ±0.0066

probability that the i-th sample belongs to the *real* class. We used Adam [37] as an optimizer with a learning rate of 10^{-4}, β_1 set to 0.9, and β_2 set to 0.999. Following the initial training, we unfroze the base models and fine-tuned them for an additional 20 epochs with a learning rate of 10^{-5}. Once again, we saved the models with the least validation loss, following the same protocol as in the initial training phase. To evaluate the performance of our model, we used several evaluation metrics, including accuracy, precision, recall, specificity, F1-score, and area under the curve (AUC). Extensive details about the metrics are in Appendix B.

5. Results & Discussion

The experiments involved the execution of a binary classification task designed to discriminate between medical images categorized as *real* and *fake* across a collection of 13 distinct models. In the initial phase, we conducted a single round of evaluation, wherein we computed the evaluation metrics (defined in Appendix B), after which we calculated the harmonic mean of prominent performance metrics, including accuracy, F1-Score, and specificity, for each of the models. Subsequently, we employed this assessment to identify the top-performing six models, subjecting them to two additional evaluation iterations. This way we not only expedited the process of model selection but also facilitated a more comprehensive investigation of their stability characteristics.

ResNet50V2 Exhibits Distinctive Performance With consistent robustness in comparison to other models. This distinction is evident in the evaluation metrics detailed in Appendix B, summarized in both Table 1 and Table 2. ResNet50V2 particularly excels, achieving the highest precision and recall, both around 99%.

Figures 5 and 6 visually highlight ResNet50V2's unique characteristics, as its outcomes exhibit a notable lack of substantial correlation with those of other models. This is measured using Pearson's correlation coefficient, defined as:

$$\rho = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{2}$$

where x and y represent the results of each model in vector form, constructed as $[\text{Accuracy}_m, \text{Precision}_m, \text{Recall}_m, \text{Specificity}_m, \text{F1-Score}_m, \text{AUC}_m]^T$.

This distinct performance of ResNet50V2 is attributed to the intrinsic characteristics of the residual blocks within the

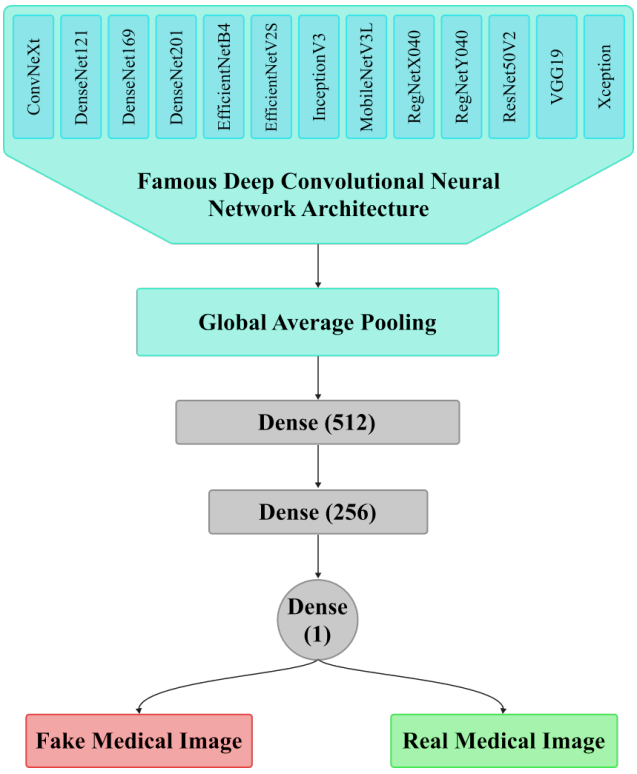

Figure 2. Full architecture used for our experiments, which consists of a Deep Convolutional Neural Network (DCNN), a global average pooling layer, a Dense (fully-connected) layer of size 512 followed by a size of 256, and finally a Sigmoid activation function indicating a probability of how much an image is *real*.

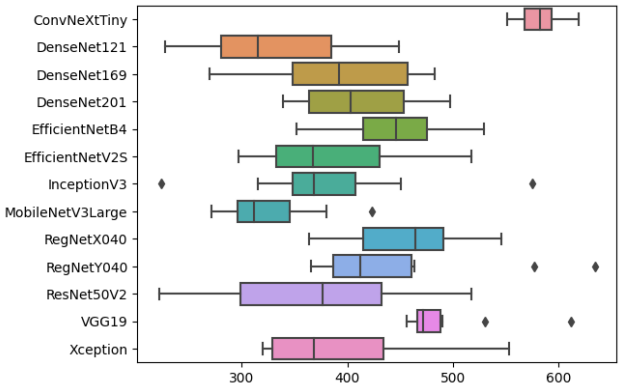

Figure 3. Time recorded per inference step after ten inferences on various Deep Convolutional Neural Networks

ResNet family. These blocks effectively preserve features from the input image and consistently integrate them across subsequent layers, contributing to the model's robustness. It is noteworthy to observe that all DCNNs demonstrated commendable proficiency in capturing differentiating features, as illustrated in Figure 7.

Table 3. Number of parameters and mean inference step time of different Deep Convolutional Neural Networks (10 runs)

Model	Parameters (M)	Inference step (ms)
ConvNeXtTiny	28.3	582.1 ±18.6
DenseNet121	7.7	334.9 ±72.4
DenseNet169	13.6	396.9 ±67.2
DenseNet201	19.4	409.2 ±51.8
EfficientNetB4	18.7	442.9 ±52.7
EfficientNetV2S	21.1	382.6 ±65.7
InceptionV3	23.0	380.7 ±86.6
MobileNetV3Large	**3.6**	**325.4 ±44.7**
RegNetX040	21.7	457.2 ±56.1
RegNetY040	20.3	446.3 ±86.3
ResNet50V2	24.7	368.3 ±87.2
VGG19	20.4	490.2 ±45.1
Xception	22.0	395.3 ±76.9

DenseNet169 Consistently Excels Across Metrics Achieving the highest levels of accuracy, recall, and F1-score, as observed over three iterative runs. This consistent excellence underscores its stability and efficacy in addressing the non-uniform distribution of tumor types and nodules within CT scan images. Comparatively, DenseNet121 demonstrates a performance comparable to DenseNet169, achieving similar results with nearly half the number of parameters and quicker inference times. These findings are summarized in Table 3 and visually presented in Figure 3. Notably, this suggests that DenseNet121 offers a favorable trade-off between model complexity and performance. However, DenseNet201 exhibits a degradation in performance compared to its counterparts in the DenseNet family. This departure from the expected trend suggests that the conclusions drawn in [38] may not readily extend to our specific case. We attribute this discrepancy to potential overfitting, likely exacerbated by the increased complexity of DenseNet201's architecture.

Comparable Efficiency of MobileNetV3Large Exhibiting remarkable efficiency, and boasting minimal parameters and inference time while delivering similar results to other DCNNs. With approximately 6.81 times and 3.78 times fewer parameters than ResNet50V2 and DenseNet169, respectively, MobileNetV3Large maintains a maximum difference of only around 4% in comparison to the highest values across all evaluation metrics. This establishes MobileNetV3Large as a promising option, particularly when facing hardware limitations. Conversely, VGG19 demonstrates fair stability in its inference time, aligning with the consistency observed in ConvNeXtTiny and MobileNetV3Large, as illustrated in Figure 3. On the contrary, ResNet50V2 exhibits notable inconsistency in inference time.

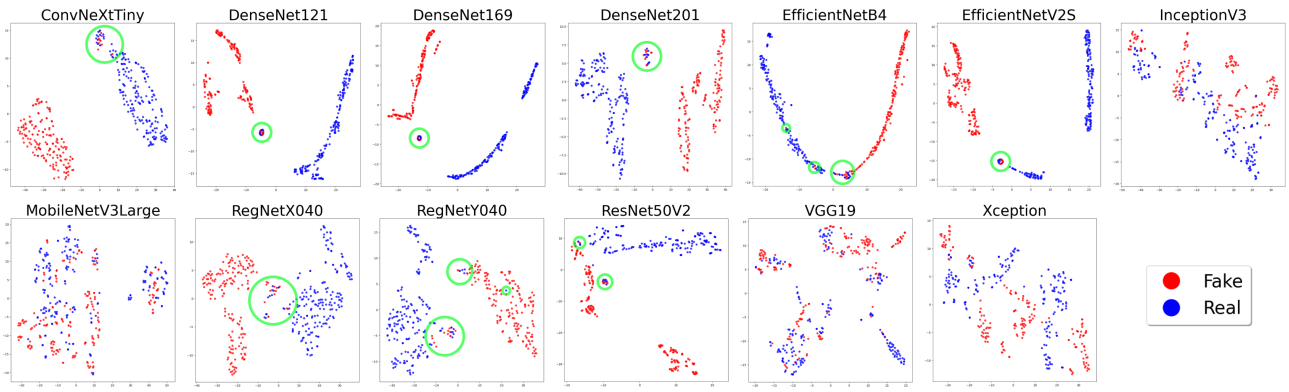

Figure 4. Latent space separability quality between *real* and *fake* examples, showing penultimate layer embedding of each Deep Convolutional Neural Network (DCNN). (minor areas of collision circled in green)

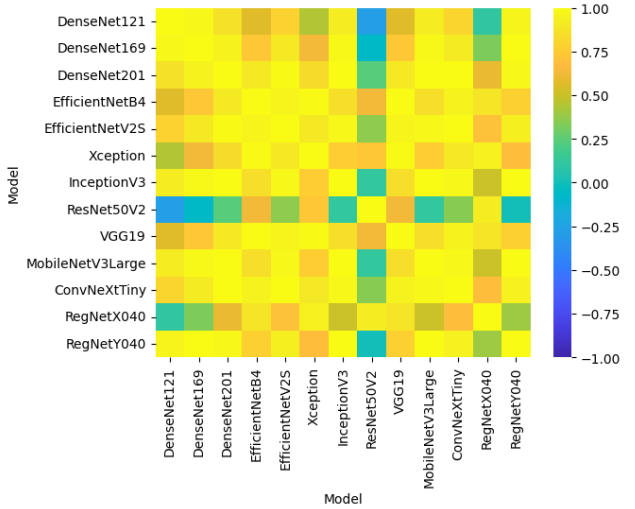

Figure 5. Pearson's correlation matrix showing interrelationships among Deep Convolutional Neural Networks (DCNNs) results.

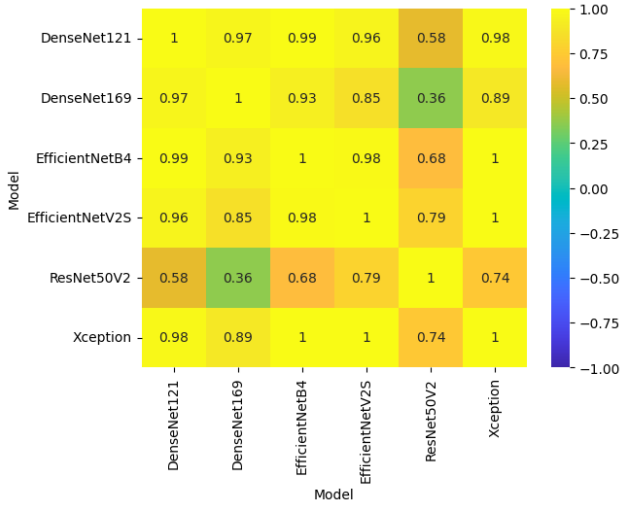

Figure 6. Pearson's correlation matrix for the top-performing Deep Convolutional Neural Networks results.

Distinguished Latent Space Separability of DenseNet169 Showing high quality separability between *real* and *fake* data points as seen in Figure 4 with visualization of the second-to-last layer embeddings. Similarly, this is seen in other members of the DenseNet family, particularly in DenseNet121. Interestingly, EfficientNetB4, EfficientNetV2S, and ResNet50V2 have a likewise outstanding ability to separate the different classes. The DCNNs mentioned above all share a common property in their latent space which is the *dense* concentration across the embeddings of the same class. This trait is not seen in the other DCNNs, which tells us that these models are less appropriate for the task of medical image deepfake detection. Moreover, there is a phenomenon seen across all DCNNs, which is what we like to call the *areas of collision*, in which examples from opposing classes overlap at different regions in the latent space.

Our exploration of various DCNN architectures has unveiled nuanced differences in their performance, prompting the question: In what context is a particular DCNN most suitable?

In medical image deepfake detection, the relationship between model architecture, complexity, and performance is intricate. For maintaining robust performance resilient to noise and out-of-distribution changes, ResNet50V2 emerges as a top choice. Its near-perfect precision and specificity are attributed to its *residual* connections, which mitigate the vanishing gradients problem, facilitate deep network training, encourage feature reuse, and enhance model expressiveness. This outstanding property is seen in

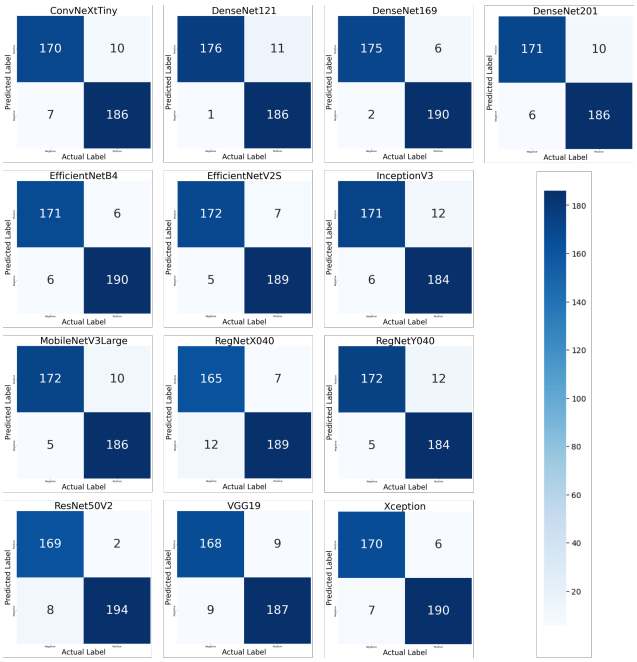

Figure 7. Confusion matrices of Deep Convolutional Neural Networks.

Figures 5 and 6, where the Pearson correlation's activations are merely shown in ResNet50V2. However, considering ResNet50V2's shortcomings in other evaluation measures, the DenseNet family, especially DenseNet169, proves valuable. Its superior accuracy, recall, and F1-score indicate a comprehensive and sensitive approach to identifying relevant cases and addressing imbalanced class distributions effectively. The unique *dense* connectivity in DenseNet, promoting feature reuse and efficient gradient flow, is a key factor behind its improved performance.

Despite the distinctive strengths of ResNet50V2 and DenseNet169, they face challenges in meeting time and hardware constraints. MobileNetV3Large addresses these limitations by offering comparable performance while executing the fastest. Designed for optimal performance on mobile phone CPUs, MobileNetV3Large utilizes a blend of hardware-aware network architecture search (NAS) and the NetAdapt [39] algorithm, contributing to its efficient execution. One thing is that important to mention is that the time complexity is not merely dependent on the architecture and number of parameters, implementation efficiency also plays a pivotal role, which can be the reason why a DCNN like ConvNeXtTiny has a substantially higher inference time while having a little increase of parameters over ResNet50V2.

These trade-offs between efficiency, stability, and performance offer valuable insights for selecting models tailored to specific medical image deepfake detection tasks under varying restrictions. One intriguing prospect is the development of hybrid architectures that combine DCNNs capturing different semantics, aiming to maintain collective efficacy when prioritizing reliable performance, albeit with a less complex model and a subtle loss of performance.

Moreover, effectively learning class-differentiating features is a complicated task that all DCNNs failed to fully succeed in as previously mentioned. We can infer from the *areas of collision* (see Figure 4) that DCNNs can either have intrinsic difficulty in learning dissimilar features that map each data point to its label or that the deepfakes are sufficiently *real*-looking, even to the machine. Another reason can be due to the data mislabeling probability as mentioned in Section 3 or because of the embedding size that can be inadequate to fully capture all the features within a given image.

6. Conclusion

In this paper, we addressed the pivotal challenge of authenticating medical images amidst the complexities introduced by deepfakes. Employing a diverse range of DCNN architectures, we utilized a dataset that amalgamates CT images generated as *fake* through CT-GAN with *real* CT images sourced from the LIDC-IDRI dataset. Our training of 13 distinct DCNN models, guided by the minimization of binary cross-entropy loss, allowed us to uncover distinctive strengths inherent to these models.

Remarkably, ResNet50V2 emerged as the preeminent model, distinguished by exceptional precision and specificity. Its consistent and reliable performance profile signifies its efficacy in the context of medical image authentication, addressing the challenges posed by GAN-generated images. In contrast, DenseNet169 showcased notable excellence in accuracy, recall, and F1-score, highlighting its stability and effectiveness in navigating the nuanced distribution within CT scan images. Furthermore, our findings unveiled that MobileNetV3Large, while offering comparable performance to leading models, stands out due to its minimal parameters and swift inference time. For future work, we plan to further our experiments with a larger set of DCNNs and explore the effects of hyperparameter choice. We are also interested in investigating the causality behind the DCNN choice to select a certain label over another and have a strong explainability framework.

These insights collectively lay the groundwork for future research endeavors within the field of DCNN applications in medical imagery, particularly in the authentication of CT scans. Additionally, they provide a valuable resource for selecting an appropriate DCNN model tailored to the unique attributes of a given medical imaging scenario, aligning with the critical need for reliable approaches in the face of GAN-generated challenges.

References

[1] Fabio Garcea, Alessio Serra, Fabrizio Lamberti, and Lia Morra. Data augmentation for medical imaging: A systematic literature review. *Computers in Biology and Medicine*, 152:106391, 1 2023. 1

[2] Richard Osuala, Kaisar Kushibar, Lidia Garrucho, Akis Linardos, Zuzanna Szafranowska, Stefan Klein, Ben Glocker, Oliver Diaz, and Karim Lekadir. Data synthesis and adversarial networks: A review and meta-analysis in cancer imaging. *Medical Image Analysis*, 84:102704, 2023. 1

[3] Alaa Abu-Srhan, Israa Almallahi, Mohammad AM Abushariah, Waleed Mahafza, and Omar S Al-Kadi. Paired-unpaired unsupervised attention guided gan with transfer learning for bidirectional brain mr-ct synthesis. *Computers in Biology and Medicine*, 136:104763, 2021. 1

[4] Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. Ct-gan: Malicious tampering of 3d medical imagery using deep learning. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 461–478, USA, 2019. USENIX Association. 1, 2

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[6] Md. Shohel Rana, Beddhu Murali, and Andrew H. Sung. Deepfake detection using machine learning algorithms. In *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 458–463, 2021. 2

[7] Anuj Badale, Lionel Castelino, Chaitanya Darekar, and Joanne Gomes. Deep fake detection using neural networks. In *15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, 2018. 2

[8] Nishika Khatri, Varun Borar, and Rakesh Garg. A comparative study: Deepfake detection using deep-learning. In *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 1–5, 2023. 2

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3

[10] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2

[11] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 2, 3

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2, 3

[13] Md. Shohel Rana and Andrew H. Sung. Deepfakestack: A deep ensemble-based learning technique for deepfake detection. In *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 70–75, 2020. 2

[14] Hong-Shuo Chen, Mozhdeh Rouhsedaghat, Hamza Ghani, Shuowen Hu, Suya You, and C.-C. Jay Kuo. Defakehop: A light-weight high-performance deepfake detector. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. 2

[15] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[16] Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. Generalization of audio deepfake detection. In *Odyssey*, pages 132–137, 2020. 2

[17] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2018. 2

[18] Yipin Zhou and Ser-Nam Lim. Joint audio-visual deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14800–14809, 2021. 2

[19] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2823–2832, 2020. 2

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[22] Junke Wang, Zuxuan Wu, Wenhao Ouyang, Xintong Han, Jingjing Chen, Yu-Gang Jiang, and Ser-Nam Li. M2tr: Multi-modal multi-scale transformers for deepfake detection. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, ICMR '22, page 615–623, New York, NY, USA, 2022. Association for Computing Machinery. 2

[23] Shreyan Ganguly, Aditya Ganguly, Sk Mohiuddin, Samir Malakar, and Ram Sarkar. Vixnet: Vision transformer with xception network for deepfakes based video and image forgery detection. *Expert Systems with Applications*, 210:118423, 12 2022. 2

[24] Rajat Budhiraja, Manish Kumar, M.K. Das, Anil Singh Bafila, and Sanjeev Singh. Medifaked: Medical deepfake detection using convolutional reservoir networks. In *2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT)*, pages 1–6, 2022. 2

[25] Siddharth Solaiyappan and Yuxin Wen. Machine learning based medical image deepfake detection: A comparative study. *Machine Learning with Applications*, 8:100298, 6 2022. 2

[26] Armato iii, s. g., mclennan, g., bidaut, l., mcnitt-gray, m. f., meyer, c. r., reeves, a. p., zhao, b., aberle, d. r., henschke, c. i., hoffman, e. a., kazerooni, e. a., macmahon, h., van beek, e. j. r., yankelevitz, d., biancardi, a. m., bland, p. h., brown, m. s., engelmann, r. m., laderach, g. e., max, d., pais, r. c. , qing, d. p. y. , roberts, r. y., smith, a. r., starkey, a., batra, p., caligiuri, p., farooqi, a., gladish, g. w., jude, c. m., munden, r. f., petkovska, i., quint, l. e., schwartz, l. h., sundaram, b., dodd, l. e., fenimore, c., gur, d., petrick, n., freymann, j., kirby, j., hughes, b., casteele, a. v., gupte, s., sallam, m., heath, m. d., kuhn, m. h., dharaiya, e., burns, r., fryd, d. s., salganicoff, m., anand, v., shreter, u., vastagh, s., croft, b. y., clarke, l. p. (2015). **Data From LIDC-IDRI** [data set]. the cancer imaging archive. 2

[27] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 3

[28] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 3

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3

[30] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3

[31] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 3

[32] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 3

[33] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020. 3

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016. 3

[35] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 3

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 9

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[38] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017. 5, 9

[39] Tien-Ju Yang, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Mark Sandler, Vivienne Sze, and Hartwig Adam. Netadapt: Platform-aware neural network adaptation for mobile applications. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 285–300, 2018. 7

Appendix

A. Model Selection

Based on our analysis, as highlighted in the CheXNet model presented by [38], the performance of architectures that have demonstrated effectiveness on ImageNet [36] does not necessarily translate to superior results with CheXNet. CheXNet, specifically trained on chest X-ray images, challenges the assumption that models excelling on general datasets, such as ImageNet, will perform equally well on domain-specific data like medical images. Therefore, we derived that existing DCNNs that perform greatly with generic data might not also perform as well in our case. As a result, we have decided to explore alternative well-established DCNN architectures to address this challenge.

To this end, we determined that the models we chose to experiment with are most reasonable to satisfy two conditions: a) experiment with various model families by choosing the model with the largest size from each family, but that is restricted by 2. The model we choose should not

Table A1. Results of the harmonic mean of the accuracy, F1-Score, and specificity on the evaluation metrics results of the proposed models after one run. The best 6 models are in bold.

Model	H
ConvNeXtTiny	0.9519
DenseNet121	**0.9595**
DenseNet169	**0.9752**
DenseNet201	0.9538
EfficientNetB4	**0.9678**
EfficientNetV2S	**0.9661**
InceptionV3	0.9468
MobileNetV3Large	0.9557
RegNetX040	0.9529
RegNetY040	0.9486
ResNet50V2	**0.9780**
VGG19	0.9517
Xception	**0.9659**

exceed 30M parameters due to computational constraints. Given that the CheXNet paper claimed intra-family generalization, we decided to experiment with all of the models in the DenseNet family to test the claim.

To choose which models we will perform 3 runs to validate their stability, we will need to calculate the harmonic mean of accuracy, F1-Score, and specificity as mentioned before. The formula for the harmonic mean H given N numbers is:

$$\mathrm{H} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} \tag{3}$$

For the case of our experiment, the formula is:

$$\mathrm{H} = \frac{3}{\frac{1}{\text{Accuracy}} + \frac{1}{\text{F1-Score}} + \frac{1}{\text{Specificity}}} \tag{4}$$

Table A1 shows the result of H for all models. Given the results of H and after reducing the number of models, we will proceed with experimenting with two more runs with the winner models, i.e. DenseNet121, DenseNet169, EfficientNetB4, EfficientNetV2S, ResNet50V2, and Xception, then calculate the mean and standard deviations of the result, consequently analyzing their results.

B. Evaluation Metrics

To calculate the evaluation metrics, we describe the four possible outcomes of our models, namely, True Positives (TPs), True Negatives (TNs), False Positives (FPs), and False Negatives (FNs).
We get a TP when the model correctly predicts the positive class, a TN when the model correctly predicts the negative class, an FP when the model incorrectly predicts the positive class, and finally, an FN when the model incorrectly predicts the negative class.

Accuracy. Accuracy measures the proportion of correct predictions made by the model out of all the predictions made. In other words, accuracy tells us how often the model's predictions were correct. The formula for accuracy is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision. Precision measures the proportion of true positive predictions out of all the positive predictions made by the model. In other words, precision tells us how many of the positive predictions made by the model were correct. The formula for precision is:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

Recall. Recall measures the proportion of true positive predictions out of all the actual positive instances in the dataset. In other words, recall tells us how many of the positive instances in the dataset were correctly identified by the model. Recall is also called sensitivity or true positive rate (TPR). The formula for recall is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

Specificity. Specificity measures the proportion of true negative predictions out of all the actual negative instances in the dataset. In other words, specificity tells us how many of the negative instances in the dataset were correctly identified by the model as negative. Specificity is also known as selectivity or true negative rate (TNR). The formula for specificity is:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{8}$$

F1-score. F1-score is the harmonic mean of precision and recall, which provides a single measure of a model's accuracy by balancing the trade-off between precision and recall. The formula for F1-score is:

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

Fall-out. Fall-out measures the proportion of actual negative instances that were incorrectly classified as positive by the model. Fall-out is also known as the false positive rate. The formula for fall-out is:

$$\text{Fall-out} = \frac{FP}{FP + TN} \tag{10}$$

We will not directly mention the fall-out in the results table, but we will rather use it to calculate the area under the curve (AUC).

AUC. AUC stands for the Area Under the Curve, specifically the ROC curve. The ROC curve (Receiver Operating Characteristic curve) is a plot of the true positive rate (sensitivity) against the false positive rate (fall-out) for various classification thresholds. AUC represents the area under the ROC curve, which ranges from 0 to 1, with higher values indicating better performance of the model. The formula for AUC is:

$$\text{AUC} = \int_0^1 \text{ROC}(x)\, dx \tag{11}$$