EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

GAN Data Augmentation for Improved Automated Atherosclerosis Screening from Coronary CT Angiography

Amel Laidi[1,*], Mohammed Ammar[2], Mostafa El Habib Daho[3], Said Mahmoudi[4]

[1]LIMOSE Laboratory M'Hamed Bougara University Boumerdes, Algeria
[2]LIST Laboratory, University M'Hamed Bougara, Boumerdes, Algeria
[3]Biomedical Engineering Laboratory, University of Tlemcen, Tlemcen, Algeria
[4]Computer Science Department, Faculty of Engineering ,University of Mons. , Belgium

Abstract

INTRODUCTION: Atherosclerosis is a chronic medical condition that can result in coronary artery disease, strokes, or even heart attacks. early detection can result in timely interventions and save lives.
OBJECTIVES: In this work, a fully automatic transfer learning-based model was proposed for Atherosclerosis detection in coronary CT angiography (CCTA). The model's performance was improved by generating training data using a Generative Adversarial Network.
METHODS: A first experiment was established on the original dataset with a Resnet network, reaching 95.2% accuracy, 60.8% sensitivity, 99.25% specificity and 90.48% PPV. A Generative Adversarial Network (GAN) was then used to generate a new set of images to balance the dataset, creating more positive images. Experiments were made adding from 100 to 1000 images to the dataset.
RESULTS: adding 1000 images resulted in a small drop in accuracy to 93.2%, but an improvement in overall performance with 89.0% sensitivity, 97.37% specificity and 97.13% PPV.
CONCLUSION: This paper was one of the early research projects investigating the efficiency of data augmentation using GANs for atherosclerosis, with results comparable to the state of the art.

Received on 28 January 2022; accepted on 16 May 2022; published on 17 May 2022

Keywords: Atherosclerosis, CCTA, Transfer learning, Generative Adversarial Networks, GAN, Data augmentation.

doi:10.4108/eai.17-5-2022.173981

1. Introduction

Cardiovascular conditions are qualified as the most fatal diseases in the world; there severity grows silently in time, without showing clinical symptoms. Atherosclerosis is a chronic medical condition where fat build-up in the artery wall causes the arteries to harden and narrow, which causes a restriction in blood flow. When it becomes severe, this problem can result in coronary artery disease, stroke, and sometimes heart attacks. Early detection of coronary artery disease (CAD) could result in timely medical interventions and save lives.

Cardiac computed tomography is the primeval diagnostic tool in the evaluation of possible coronary artery diseases [1]. This modality is traditionally used for the assessment of artery luminal stenosis, coronary artery calcification, and high-risk plaque features. However, one of the major restrictions of Cardiac CT is the difference of technical parameters used during the acquisition, reconstruction, and analysis of CT images across centers and patients which may cause an irregularity in analysis.

Recent fast advances in computer aided diagnosis systems caused a revolution in the prognosis of cardiovascular diseases allowing fast and accurate heartbeat classification [2] and arrhythmia detection from ECG recordings[3]. Which urged the need for

[*] Corresponding author. Email: a.laidi@univ-boumerdes.dz

a similarly efficient automatic analysis of cardiac CT to help physicians process images faster and more accurately, which saves time and effort and can lead to eventually preserving patients' lives.

Early research in automatic screening of atherosclerosis in CCTA focused mainly on recognizing and segmenting the cardiac arteries for a better analysis [4, 5]. Other papers were interested in segmenting stenosis [6] and plaques [7], which are the main index of atherosclerosis. Lately, White et al. [8] developed a model for direct atherosclerosis screening from CCTA with a high Negative Predictive Value (NPV) which makes it useful to confidently eliminate the presence of atherosclerosis and allow a safe discharge of patients with chest pain from the ER. In [9] the authors used a 3D CNN for feature extraction from coronary artery CT, those features are then fed to a recurrent neural network RCNN that detects and characterizes the plaque's type and the anatomical significance of the coronary artery stenosis, simultaneously. The model achieved an accuracy of 77% for coronary plaque detection and characterization and 80% for stenosis detection and recognition. One of the first papers aiming to study the possibility of an intelligent model for atherosclerosis screening in CCTA was proposed in [[10] where they used a 3D CNN. They started by locating and extracting the coronary arteries from CCTA by using a deformed mean shape model with the coronary ostia and cardiac chambers as anchor points. The initial centerline estimation was then refined through ROI masks, the surrounding volume was considered part of the vessels. The final multi-planar reformations (MPR) were straitened providing a longitudinal view of the vessels. After a pre-processing task, the MPR volumes went through a 3D CNN . The authors also ran a gradient-based class activation map (Grad-CAM) [11], to highlight images regions that affected the CNN's decision. By doing so they identified the abnormalities in the images, in order to help the physician by noticing it for an easier diagnosis. The model achieved 90.9% of accuracy, 58.8% of Positive Predictive Value, 68.9% of sensitivity, 93.6% of specificity, and 96.1% of Negative Predictive Value (NPV).

Most of the existing studies are interested in the detection and characterization of stenosis and plaque, which are main indices of atherosclerosis. Very few papers take an interest in direct atherosclerosis detection which can be much faster and more efficient. In this paper we develop an automatic deep learning framework for atherosclerosis screening straight from coronary CT Angiography images.

We used a pre-trained Resnet network for atherosclerosis screening, the network achieved a good accuracy of 95.2% but a low sensitivity of 60.8%. In order to improve sensitivity, we assumed the poor performance was due to the imbalance dataset and the lack of positive images. As a solution, we applied data augmentation by generating new images using a Generative Adversarial Network. After a number of tests involving the new generated images, we improved the sensitivity up to 89.0%.

The remaining of this paper is organized as follows: we conserve the next section 2 for The material and methods used in this paper . In section 3 we present in detail the experimentations realised, and we finish up with discussions of the results in sectoion 4 and conclusions 5.

2. Materials and methods

2.1. Dataset

The dataset we used was collected and provided by the Ohio State University Wexner Medical Center [12]. The images were obtained from 200 randomly selected CCTA exams, 108 of male patients and 92 of female patients. The subjects are around 50.6 years old with a deviation of 12.1 years. 100 patients have atherosclerosis, while the other 100 are atherosclerosis-free according to the reviewing of an investigator-expert (RDW with 33-year experience in cardiac imaging and ACC/AHA level III CCT certification). All images were acquired using a multi-detector CT system (Siemens Healthineers, Erlangen, Germany), and reviewed.[13] The dataset contains a Mosaic Projections View (MPV) of coronary artery images from 500 patients. Each image consists of a vertical series of 18 different straightened 2D projections of a coronary artery. The dataset is primarily divided into 300 training images, 100 test images and 100 validation images. 6-fold augmentation was applied to the Artery images derived from 300 training cases, creating 2,364 images; meanwhile, all the normal components of the training data, the entire validation, and the test datasets were kept intact. The training data contains 2364 positive images and 2304 negative images, test data contains 125 positive images and 1066 negative images, while the validation data contains 50 positive and 50 negative images.

2.2. Transfer learning/RESNET

Transfer learning is a modern machine learning technique that takes a model, trains it on a big available dataset, then keeps weights from the initial layers and only train the final layers on the target data for a better fine-tuning. It is usually used when: 1) there is not sufficient data to train a model from scratch leading to lower performance, starting with a pre-trained model that has already learned basic features would make better use of the available data for tuning and reaching a better performance. 2) there is not enough time to build a new model, when working with new data it can be difficult to come up with the best architecture that can adopt to it, experimenting with a pre-trained models is the fastest way to solve the problem. 3) there is not enough computing power, as even with enough data, bigger models usually require powerful computers to perfectly train the model.

Transfer learning had proven to be very efficient in medical image processing, if done right. Adequately fine-tuned pretrained models can compete with custom made neural networks, if not outperform them. In [14], the authors obtained a maximum accuracy of 88.3% for multi-class classification of diabetic eye disease using a pretrained VGG16 model. While in [15], the authors developed a new convolutional neural network (CNN) model from scratch for the same task that achieved a maximum accuracy of 81.33%.

Most available pre-trained models have been developed by powerful research teams using the proper hardware and enough data, most of them were trained on ImageNet, a dataset containing more than 15 million labeled high-resolution images Transfer learning has proven particularly useful applications in medical problems where annotated data is unavailable and expensive to obtain. [16] We have tested several pre-trained models before settling on Resnet. Resnet or residual network was first introduced at ILSVRC 2015 (ImageNet Large Scale Visual Recognition Challenge). It came as a solution to avoid overfitting when allowing training of deeper networks, but without having a problem of exploding gradient, which is a recurrent issue with deeper models. It was trained on a subset of ImageNet composed of 1000 categories each containing 1000 images, coming down to about 1.2 million training images, 50,000 validation images and 100,000 testing images. Resnet uses “skip connections” to leave out training from a few layers and connects directly to the output. Instead of skipping to random layers, it uses "residual mapping" where it directly fits a desired underlying mapping, the ResNet explicitly lets these layers fit a residual mapping of:

$$F(x) = H(x)\check{\ }x \tag{1}$$

where : F(x) + x. refers to the original.

In order to find a solution for vanishing/exploding gradients, ResNet architecture uses “skip connections” to leave out training from a few layers and adds it directly to the output: Even if the weight layers caused a vanishing of the gradient, we would still have the identity x added back to earlier layers as shown in Figure 1. We used a model with 101 layers. We tested a number of hyperparameters in order to fine-tune the model for optimal results. Every hyperparameter has a specific impact on the final result.

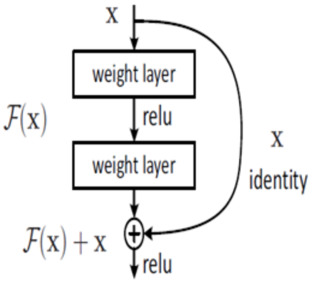

Figure 1. A Building Block of Residual Network [17]

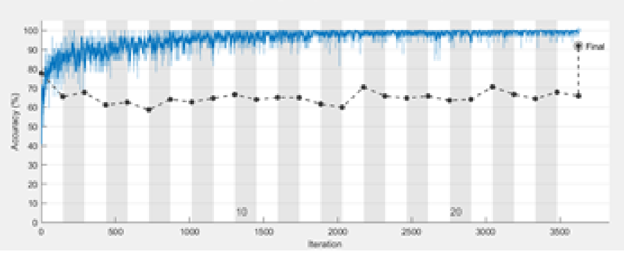

Figure 2. Training progress of Resnet 101

Minibatch is a subset sample from the training dataset, its size can vary from 1 the total number of examples in the training dataset. Stochastic gradient descent is sequential and computes the gradient on minibatches, as opposed to the entire training dataset, because it tends to converge more quickly, since it doesn’t need to make a full pass through the training data to update the weights. Using larger batch sizes permit a parallel computation [18], which speeds up model training. However, if too large, batch sizes generalize poorly on unseen data. The perfect balance is a batch size that significantly speed up training without sacrificing model accuracy. In our case we tried 32, 64 and 128. And eventually settled on a batch size of 64.

Training epochs are the number of cycles the model would go through the whole dataset. Training for too long epochs can cause the model to overfit, while not training enough can lead to underfitting. As it is impossible to know the right epoch before training, it is best to assess learning curve graphs as the model learns. Figure 2 shows that reaching 25 epochs increase training performance without overfitting.

Learning rate is the hyperparameter that controls at which pace the model learns. It is the amount of error assigned to the layers’ weights at each update. Larger learning rates permit faster learning, at the cost of missing a convergence of weights. Smaller learning rates however allow more optimal learning but can be significantly costly in terms of time and computation. We tested learning rates varying between 10-3, 10-4 and 10-5. Using the latter caused the training time to triple, so we settled on 10-4 for a better and slightly faster training.

Dropout is a regularization technique. It cancels out several neurons from the network at the training stage, following a predetermined probability, keeping them from contributing to forward and backward propagation. Thus, changing the model's architecture at each iteration , which leads to more a robust training. [19]

We trained our model without dropout, on a "rigid" network, then tested an increasing rate of dropout (10%, 20%, 30%, and 50%). The best performance was associated with a 50% dropout. Table 1 shows different results of testing and finetuning the hyperparameters. we finally settled on a residual network with 101 layers, trained it for 25 epochs, using a 64 minibatch size , an initial learning rate of 10^{-4} and a drop rate of 30%. The model achieved an accuracy of 95.21%.

2.3. Generative Adversarial Networks and data augmentation

First defined in the paper "Generative Adversarial Networks" by Ian Goodfellow and colleagues in 2014 [20], Generative advertiser networks (GAN) are the type of deep learning networks that uses adversarial methods to understand and acquire the generative model of data distribution to produce new images as similar to the original images as possible, they contain the input vector, generator, and discriminator. The generative model produces fake samples of data that can fool the discriminator in order to learn the data distribution. The abstract mathematical explanation of Generative Adversarial Networks was illustrated in [21] as follows:

Considering how difficult it is to determine the fixed distribution of data $P_{data}(x)$, it is conventional to consider that $P_{data}(x)$ follows a Gaussian mixture distribution and make use of the maximum likelihood instead. However, when dealing with a complicated model , it is often impossible to calculate the distribution which restricts the end performance. As a result, the distribution $P_g(x)$ can be calculated by using artificial neural networks (ANN).

The generator is an ANN with parameter g. It starts by first collecting the random variable z from the preceding distribution , and then mapping it to the pseudo-sample distribution through the ANN, the new generated data is registered as G(z) while its distribution is registered as $P_g(z)$. According to the parameter $_g$, several complex distributions can be generated from one simple input distribution. The generator's goal is to minimize the difference between $P_g(x)$, the images it generates and $P_{data}(x)$, the input image distribution.

$$G^* = argmin_G Div(P_g, P_{data}) \tag{2}$$

Since the exact forms of the distributions are unknown, the difference can't be calculated directly, which leads to building another neural network called the discriminator to learn the difference between the two distributions. In the original GAN [20] the authors used a binary classifier [22] with d as the discriminator, which outputs 1 for a real sample x, and 0 otherwise. They measured the loss using binary cross entropy, a commonly used function for binary classification.

$$Loss = -(y\log(\hat{y}) + (1-y)\log(1-\hat{y})) \tag{3}$$

y : the sample label
ŷ : the odds for the prediction sample to be a positive example.

If the model prediction sample is correctly classified, ŷ is set to 1; otherwise, ŷ is set to 0. The positive cases are substituted into P_{data} , and the negative cases into P_g. In short, The GANs are formulated so that the Discriminator works on minimizing its outcome V(D, G), while the Generator reduces the Discriminator's outcome and increases its loss. This is summarized with the following formula :

$$V(G, D) = \ E_{x\sim Pdata}[logD(x)] + E_{x\sim Pg}[log(1 - D(x))] \tag{4}$$

Where:
P(z) = distribution of generator
x = sample from $P_{data}(x)$
z = sample from P(z)
D(x) = Discriminator neural network
G(z) = Generator neural network

In order for the generator G to fool the discriminator D, it works on maximizing the discriminator's output when it is presented with a generated (fake) sample. On the other hand, the discriminator tries to differentiate real data from generated samples, Therefore, the discriminator works on maximizing V (G, D) as the generator minimizes it, which creates a minimax relationship.

$$min_G max_D \ V(G, D) = min_G max_D E_{xPdata}[logD(x)] + E_{z\sim Pz}\ [log(1 - D(G(z)))] \tag{5}$$

The training of a Generative Adversarial Network happens in two steps:

•**Step 1:** The parameters of the generator are fixed while the Discriminator trains. The network only forward

Table 1. Accuracy turnout during hyperparameter fine-tuning

MiniBatch Size	32	32	32	64	64	64	64	64	128
EPOCHS	10	25	10	6	10	20	20	25	25
Learning rate	10-5	10-5	10-4	10-4	10-5	10-4	10-3	10-4	10-5
dropout	0	0.2	0.3	/	0.1	0.5	0.5	0.5	0.5
Accuracy (%)	91.10	91.94	94.71	90.43	86.15	94.29	89.50	95.21	91.18

propagates. The Discriminator is trained on a positive sample x from the real data set. while the generator creates a negative sample G(z).

•**Step 2:** The parameters of the discriminator are fixed while the Generator trains. The generated data is fed to the discriminator, the difference between the output of the discriminator D (G(z)) and the sample label is calculated, and the error is used to update the generator's parameters of using backpropagation algorithm.

The steps above are repeated until approaching $P_{data}(x) = P_g(x)$.

Generative Adversarial Networks have gained an interest in image processing and computer vision as a mean of data augmentation, and they showed even more valuable added value in medical applications, as labeled medical data is more difficult to acquire. Even though traditional augmentation methods have been widely used in the past, the generated images usually have a comparable distribution to the original images and can't mimic the differences that exist between different patients.

GANs have been widely used in cardiology for various tasks including generating realistic cardiac images [23], creating synthetic electrocardiography signals [24], synthetic CMR images [25], and imitations of electronic medical histories. [26] in order to increase and balance out our dataset, we propose in this work to use our dataset set to train a Generative Adversarial Network model allowing to generate new data. We trained our model with a mini-batch size of 128 for 50 epochs. We used Adam optimization with a learning rate of $2*10^{-4}$, a gradient decay factor of 0.5, and a squared gradient decay factor of 0.999.

After training for approximately 1800 iterations (Figure 3), the synthetic images provided by the generator resembles real data, and looks identical to the naked eye (Figure 4)

3. Experimentations and results

We have applied ResNet to the original data from the dataset [12]. This first experiment achieves 95.2% of accuracy, 60.8% of sensitivity, 99.26% of specificity, 90.48% of positive predictive value and 95.57% of Negative Predictive value. Then we experimented by adding 100, 300, 500 and 1000 images to the dataset, Table 2 shows in detail the outcome of each test.

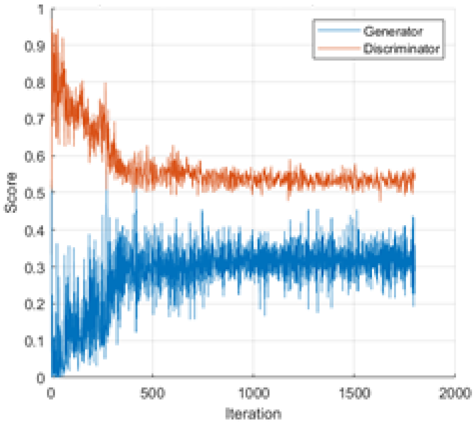

Figure 3. Progression of generator and discriminator during GAN training

For this experimentation we have different tests that we describe in the following sections:

First test: we started by generating and adding 100 images to the positive training folder, and 100 images to the negative training folder. We recorded a slight drop in accuracy (94.0%), and an improvement in sensitivity (75.2%). This step served as a reference to check the impact of using generated data on the system accuracy, although there wasnt a big loss of accuracy, the model had a 30% drawback in Positive Predictive Value, which went from 90.5% to 60.6%.

Second test: we then added 300 positive generated images to the training folder, we recorded a drop in both accuracy (86.9%) and sensitivity (61.6%). By adding more positive images to the training data, we expected that the model would learn to identify them better, instead we had a big drop in Positive Predictive Value from 90.5% in the first model to 43.4%

Third test: we moved the extra 300 generated images from the second test into the test folder, in order to insure a better balance of the test dataset. We recorded an improvement in accuracy, sensitivity and Positive Predictive Value, which were 88.2%, 75.6%, 94.2%, respectively. This suggests that the best use of the generated data should be in balancing the testing dataset.

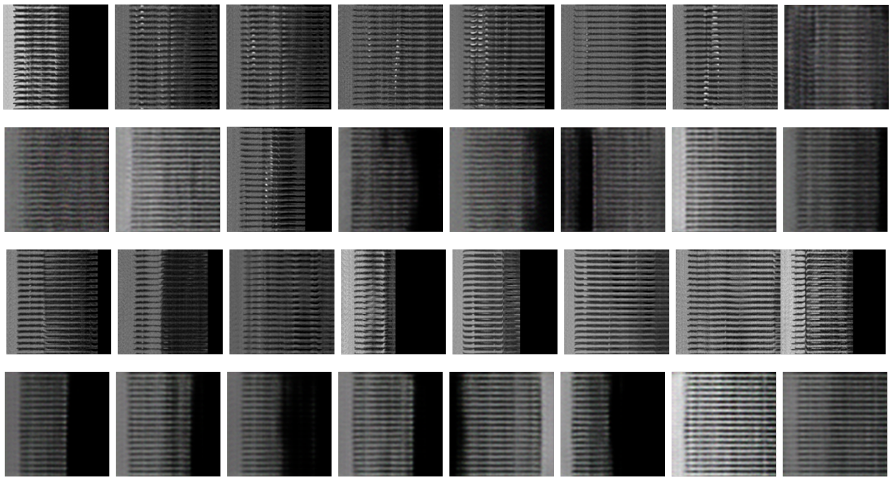

Figure 4. Images used during testing, (first line) positive images from the dataset, (second line) generated positive images, (third line) negative images from the dataset, (fourth line) generated negative images

Table 2. Performance of Resnet after each test

	Accuracy	sensitivity	specificity	PPV	NPV
Original	95.2	60.8	99.25	90.48	95.57
Test 1	94.0	75.2	96.2	60.6	97.1
Test 2	86.9	81.6	87.5	43.4	97.6
Test 3	88.2	75.6	96.8	94.2	85.4
Test 4	93.2	89.0	97.4	97.1	89.9

Fourth test: we moved 1000 positive images from the training dataset into the testing dataset and replace them with 1000 generated images. This was done to ensure that the testing samples were similar to unseen data, which would make the model more robust and prone to recognize new data. We recorded the best results yet: 93% accuracy, which is slightly less than the results of the first test, but with 89% sensitivity and 97.1% PPV.

4. Discussions

In this paper we have developed an automatic framework for screening atherosclerosis from Coronary CT angiography images, based on a pertained residual network. The initial model achieved a high accuracy of 95.2% . Interestingly, it obtained a high negative predictive value of 95.57% , which qualifies it to accurately predict the absence of atherosclerosis when suspected. This can be useful for safely discharging patients with chest pain in the ER, without need for further tests. However, this model presented a low sensitivity of 60.8%, and it was less prompt to recognize the cases with atherosclerosis. We assumed this was due to the lack of POSITIVE images in the test data set (125 positive images, vs. 1066 negative images). To confirm our theory, we enlarged the test set by adding new images generated from a GAN model. After number of experiences, we added 500 generated positive images to the dataset in order to balance it out. the final training with the new dataset resulted in a slight draw back in accuracy to 93.2%, and specifity to 97.37%, but an improvement in both sensitivity (89.0%) and positive predictive value (97.13%). Atherosclerosis screening straight from CCTA images is a new field of research, which makes comparing our results with state of the art a little challenging. Candemir et al. [10] stands out as a recent innovative study, their model achieved 90.9% of accuracy, 58.8% of Positive Predictive Value, 68.9% of sensitivity , 93.6% of specificity, and 96.1% of Negative Predictive Value (NPV). In comparison, the model proposed in this paper had a better performance. The drop in accuracy and specifity after adding the generated images can be due to the expand in the dataset’s size, which can be fixed by training the model for longer.

5. Conclusion

In this paper we have developed an automatic framework for screening atherosclerosis from Coronary CT angiography images, based on a pertained residual network with 101 layers. For fine tuning, we trained the model for 25 epochs using a 64 minibatch size, an initial learning rate of 10^{-4} and a drop rate of 30%. The model achieved an accuracy of 95.21%, 60.8% sensitivity, 99.25% specificity and 90.48% PPV. The model performed considerably well as is, but we tried to improve its sensitivity. We assumed the low sensitivity was due to the lack of positive images in the dataset (test data contains 125 positive images and 1066 negative images). To overcome the issue, we used a Generative Adversarial Network to generate new images resembling the original data. After experimenting with generated data and distributing it during training, our model achieved 93.2% of accuracy, 89.0% of sensitivity, and 97.13% of Positive Predictive Value. The results obtained were comparable to the best results from state of the art studies.

To the best of our knowledges, this paper is one of the first papers studying the impact of using GAN for generating CCTA images for data augmentation to improve atherosclerosis screening. The results achieved are not only comparable to the state of the art, but also allow a fast and accurate automatic detection of atherosclerosis, which has important clinical implications.

Due to the lack of data, it was not possible to confirm whether training the model on a well-balanced dataset of native images (no GAN generated images) could get the same or better results. The study needs to be evaluated once enough data can be acquired.

For future work, it is important to find out the reason behind the slight drop in accuracy after using the GAN generated images, and how to avoid it.

References

[1] Oikonomou, Evangelos K.; West, H.W.A.C. (2019) Cardiac computed tomography assessment of coronary inflammation and other plaque features. *Arteriosclerosis, Thrombosis, and Vascular Biology* **39**(1): 2207–2219.

[2] He, J., Sun, L., Rong, J., Wang, H. and Zhang, Y. (2018) A pyramid-like model for heartbeat classification from ecg recordings. *PloS one* **13**(11): e0206593.

[3] He, J., Rong, J., Sun, L., Wang, H., Zhang, Y. and Ma, J. (2020) A framework for cardiac arrhythmia detection from iot-based ecgs. *World Wide Web* **23**(5): 2835–2850.

[4] Shen, Ye; Fang, Z.G.Y.X.N.Z.C.T.X. (2019) Coronary arteries segmentation based on 3d fcn with attention gate and level set function. *IEEE Access* **7**: 42826–42835. doi:10.1038/s41598-020-59215-9, URL https://doi.org/10.1038/s41598-020-59215-9.

[5] Huang, W., Huang, L., Lin, Z., Huang, S., Chi, Y., Zhou, J., Zhang, J. *et al.* (2018) Coronary artery segmentation by deep learning neural networks on computed tomographic coronary angiographic images. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*: 608–611. doi:10.1109/EMBC.2018.8512328.

[6] Hong, Y., Commandeur, F., Cadet, S., Goeller, M., Doris, M., Chen, X., Kwiecinski, J. *et al.* (2019) Deep learning-based stenosis quantification from coronary ct angiography. In *Medical Imaging 2019: Image Processing* (International Society for Optics and Photonics), **10949**: 109492I.

[7] Liu, J., Jin, C., Feng, J., Du, Y., Lu, J. and Zhou, J. (2018) A vessel-focused 3d convolutional network for automatic segmentation and classification of coronary artery plaques in cardiac cta. In *International Workshop on Statistical Atlases and Computational Models of the Heart* (Springer): 131–141.

[8] White, R.D., Erdal, B.S., Demirer, M., Gupta, V., Bigelow, M.T., Dikici, E., Candemir, S. *et al.* (2021) Artificial intelligence to assist in exclusion of coronary atherosclerosis during ccta evaluation of chest pain in the emergency department: Preparing an application for real-world use. *Journal of Digital Imaging* : 1–18.

[9] Zreik, M., Van Hamersvelt, R.W., Wolterink, J.M., Leiner, T., Viergever, M.A. and Išgum, I. (2018) A recurrent cnn for automatic detection and classification of coronary artery plaque and stenosis in coronary ct angiography. *IEEE transactions on medical imaging* **38**(7): 1588–1598.

[10] Candemir, S., White, R.D., Demirer, M., Gupta, V., Bigelow, M.T., Prevedello, L.M. and Erdal, B.S. (2020) Automated coronary artery atherosclerosis detection and weakly supervised localization on coronary ct angiography with a deep 3-dimensional convolutional neural network. *Computerized Medical Imaging and Graphics* **83**: 101721.

[11] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D. (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*: 618–626.

[12] Demirer, Mutlu; Gupta, V.B.M.E.B.P.L.W.R., "image dataset for a cnn algorithm development to detect coronary atherosclerosis in coronary ct angiography", mendeley data, https://data.mendeley.com/datasets/fk6rys63h9/1. doi:10.17632/fk6rys63h9.1.

[13] Gupta, V., Demirer, M., Bigelow, M., Little, K.J., Candemir, S., Prevedello, L.M., White, R.D. *et al.* (2020) Performance of a deep neural network algorithm based on a small medical image dataset: incremental impact of 3d-to-2d reformation combined with novel data augmentation, photometric conversion, or transfer learning. *Journal of digital imaging* **33**(2): 431–438.

[14] Sarki, R., Ahmed, K., Wang, H. and Zhang, Y. (2020) Automated detection of mild and multi-class diabetic eye diseases using deep learning. *Health Information Science and Systems* **8**(1): 1–9.

[15] Sarki, R., Ahmed, K., Wang, H., Zhang, Y. and Wang, K. (2022) Convolutional neural network for multi-class classification of diabetic eye disease. *EAI Endorsed Transactions on Scalable Information Systems* : e15–e15.

[16] Kora, P., Ooi, C.P., Faust, O., Raghavendra, U., Gudigar, A., Chan, W.Y., Meenakshi, K. *et al.* (2021) Transfer learning techniques for medical image analysis: A review. *Biocybernetics and Biomedical Engineering* .

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

[18] Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M. and Tang, P.T.P. (2016) On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836* .

[19] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**(1): 1929–1958.

[20] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. *et al.* (2014) Generative adversarial nets. *Advances in neural information processing systems* **27**.

[21] Lan, L., You, L., Zhang, Z., Fan, Z., Zhao, W., Zeng, N., Chen, Y. *et al.* (2020) Generative adversarial networks and its applications in biomedical informatics. *Frontiers in Public Health* **8**: 164.

[22] Deng, L.Y. (2006) The cross-entropy method: A unified approach to combinatorial optimization, monte-carlo simulation, and machine learning. *Technometrics* **48**(1): 147–148. doi:10.1198/tech.2006.s353, URL https://doi.org/10.1198/tech.2006.s353. https://doi.org/10.1198/tech.2006.s353.

[23] Skandarani, Y., Painchaud, N., Jodoin, P.M. and Lalande, A. (2020), On the effectiveness of gan generated cardiac mris for segmentation. 2005.09026.

[24] Wulan, N., Wang, W., Sun, P., Wang, K., Xia, Y. and Zhang, H. (2020) Generating electrocardiogram signals by deep learning. *Neurocomputing* **404**: 122–136. doi:https://doi.org/10.1016/j.neucom.2020.04.076, URL https://www.sciencedirect.com/science/article/pii/S0925231220306615.

[25] Amirrajab, S., Abbasi-Sureshjani, S., Al Khalil, Y., Lorenz, C., Weese, J., Pluim, J. and Breeuwer, M. (2020) Xcat-gan for synthesizing 3d consistent labeled cardiac mr images on anatomically variable xcat phantoms. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer): 128–137.

[26] Skandarani, Y., Lalande, A., Afilalo, J. and Jodoin, P.M. (2021) Generative adversarial networks in cardiology. *Canadian Journal of Cardiology* .