MACHINE LEARNED DATA AUGMENTATION TECHNIQUES FOR IMPROVING PATHOLOGY OBJECT DETECTION

By

Ethan Tu

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Biomedical Engineering – Doctor of Philosophy

2023

ABSTRACT

Artificial intelligence (AI) has evolved immensely in recent years, with AI achieving human levels of performance on a wide variety of tasks. However, AI has had limited adoption in clinical settings despite its promising prediction, classification and pathology detection applications. For a machine learned (ML) model to train effectively, the observed data must be a diverse, accurate representation of the true distribution. Therefore, to properly estimate the true distribution, extremely large datasets become necessary. In healthcare scenarios, datasets of sufficient size may be rare or absent, thus hindering the training of ML models. One of the ways to mitigate this problem is through data augmentation, where we supplement our datasets with slightly modified copies of already existing data or newly created synthetic data. Recently, sophisticated data augmentation methods are based on a class of neural networks (NNs) called Generative Adversarial Networks (GANs), which can generate new images of high perceptual quality. This dissertation describes the design and development of a new type of GAN, named near-pair patch cycleGAN (NPP-cycleGAN), which generates realistic pathology-present images. Here, we train and test this network using pediatric chest radiographs. We demonstrate that the proposed GAN can generate high quality fracture-present pediatric chest radiographs. With the addition of these synthetic images to an object detector's training dataset, we are able to improve the fracture detection performance. These results suggest that our proposed method can be applied to other pathology detection tasks and could potentially enable improved object detector performance in multiple clinical scenarios.

ACKNOWLEDGEMENTS

I would like to thank my friends, family, and peers for their endless support. Many thanks to my advisor, Dr. Adam Alessio, who without his guidance and wisdom this dissertation would not have been possible. Finally, thank you to all the Spartans who have made this community feel fun and welcoming.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AHA - American Heart Association

AUC – Area Under the Curve

CAD – Coronary Artery Disease

CNN – Convolutional Neural Network

CT – Computed Tomography

CTA or CCTA – Coronary Computed Tomography Angiography

dCTP or DCE-CT – Dynamic Contrast-Enhanced Computed Tomography Perfusion

DL – Deep Learning

DNN – Deep Neural Network

FFR – Fractional Flow Reserve

FFR-CTPA – Fractional Flow Reserve from CT Perfusion and Anatomy

FID – Fréchet Inception Distance

GAN – Generative Adversarial Network

IRB – Institutional Review Board

IQR – Interquartile Range

JI – Jaccard Similarity Index

LAD – Left Anterior Descending Coronary Artery

LCX – Left Circumflex Coronary Artery

MBF – Myocardial Blood Flow

MFR – Myocardial Flow Reserve

ML – Machine Learning

MRI – Magnetic Resonance Imaging

MSE – Mean Squared Error

NN – Neural Network

NPP – Near Pair Patch

PCL – Polycaprolactone

PET – Positron Emission Tomography

PLGA – Poly(lactic-co-glycolic acid)

PSNR – Peak Signal to Noise Ratio

TAC - Time Attenuation Curve

RMSE - Root Mean Squared Error

PCA – Principal Component Analysis

RCA – Right Coronary Artery

ROC – Receiver Operating Characteristics

SSIM – Structural Similarity Index Measure

TaOX – Tantalum Oxide

CHAPTER 1: PREVIOUS AND RELATED WORK

In this chapter I will describe three of my major research projects. I will review the key findings and methodologies of the relevant studies, highlighting their significance. For source code and data, please refer to Appendix A. These efforts were presented in the following outlets: Section 2.1 [1], Section 2.2 (to be submitted to Journal of Cardiovascular Computed Tomography), Section 2.3 [2].

1.1 Machine Learned Approach for Estimating Myocardial Blood Flow from Dynamic CT and Coronary Artery Disease Risk Factors

Heart disease is one of the leading causes of death in the US, with about 1 in 20 adults over the age of 20 diagnosed with some form of coronary artery disease. The estimation of myocardial blood flow (MBF) is crucial for diagnosing and risk stratifying myocardial ischemia. Currently, the gold standard for non-invasive, quantitative MBF measurements is to use positron emission tomography (PET). However, we seek to use low radiation dosage dynamic contrast-enhanced computed tomography perfusion (dCTP) as an alternative approach due to its wide availability and lower cost. This work uses machine learning techniques to estimate MBF from a combination of dCTP derived time attenuation curves (TACs) and 9 risk factors for coronary artery disease (CAD). We compare our machine learned MBF estimates to PET derived estimates, and for a control, we used a 2-compartmental model that has been previously presented and verified with simulation studies. Four machine learning regression techniques were explored: 1) Binary regression tree, 2) Ensemble of Learners regression, 3) Support vector machine, and 4) Kernel regression. Our best performing model (ensemble of trees) had a root mean squared error (RMSE) of 0.47 ml/min/g. Comparatively, the compartmental model achieved an RMSE of 0.80 ml/min/g. In general, the inclusion of risk factors neither improved nor worsened estimates. Overall, our machine learning approach produces comparable MBF estimations to verified DCE-CT and PET estimates and can

provide rapid assessments for myocardial ischemia.

1.1.1 Introduction

The non-invasive quantitative assessment of myocardial perfusion is essential for grading coronary artery disease. Quantitative MBF provides valuable prognostic and diagnostic information and can be measured through the use of positron emission tomography (PET) [3]–[5], magnetic resonance imaging (MRI) [6]–[9], and computed tomography (CT) [7], [10]–[12]. However, PET remains the gold standard for quantitative MBF measurements. Recently, cardiac perfusion related PET has been focused on reducing costs and improving patient outcomes through the development of new radiotracers [13], determining optimal thresholds to stratify CAD [14], [15], or diagnostic accuracy studies [16]–[18]. Even with these studies, the use of PET for quantifying ischemia has remain limited due to the cost of PET perfusion tracers, methodologic complexity, and insurance reimbursement issues [4]. MRI, likewise, shares the same cost, availability. and complexity drawbacks [10]. CT, on the other hand, is low cost, rapid, widely available, and produces images of better spatial resolution than PET [12]. However, CT based estimates are generated with compartmental modeling that requires the entire time course of the contrast agent and that do not implicitly model noise properties in the data (requiring relatively high radiation dose).

This work compares four different machine learning algorithms to derive MBF from dCTP and patient risk factors. Using simple machine learning methods has many potential benefits: a) bypassing computationally expensive compartmental models, b) inherently learning noise properties of the data, and c) identifying future candidate approaches for simplifying CT acquisitions. Other studies have tried to use machine learning (ML) techniques for cardiology tasks due to its ability to handle large volumes of data and its ability to model hidden patterns within data [19]. ML has been used to predict the likelihood of revascularization in patients with CAD

[20], assessment of coronary stenosis through FFR prediction [21], to predict major adverse cardiac events [22], and for calcium scoring [23]. We are the first, to our knowledge, to directly estimate MBF using a ML model trained on a fusion of DCE-CT derived time attenuation curves and patient risk factors. Here, we compare our ML derived estimates to CT-derived 2-compartmental model and quantitative PET estimates.

1.1.2 Materials and Methods

Twenty-nine patients underwent clinical rest and stress regadenoson rubidium-82 PET scans. QPET software (Cedars-Sinai, Los Angeles, CA) was used to produce quantitative PET MBF estimations. Each patient then underwent a DCE-CT exam, using a Revolution CT scanner (GE Healthcare, Waukesha, WI) with a 16 cm z-axis coverage, within 30 days of the initial PET scan. Using a custom, previously verified 2-compartmental model written in MATLAB (ver 2017b; MathWorks, Natick, MA) we estimated MBF from the CT images [24]. For each PET and CT scan, we segmented the heart into the recommended 17-region myocardial AHA model yield a total of 493-time attenuation curves and segments for flow estimation [25].

In Figure 1, representative time attenuation curves (TAC) for one segment are shown. Quantitative MBF was expressed as mL/min/gram of myocardial tissue. We explored 4 ways of summarizing the TAC data for our ML models: (1) trained on only the myocardial output response TAC, (2) trained on both the arterial input and myocardial output TAC, (3) trained on semantic feature selection of our myocardial output TAC, and (4) trained on principal component analysis (PCA) of the myocardial output TAC. For our semantic feature selection, we choose predictors that we already know are important for flow estimation. These predictors include the rising slope, area under the curve normalized by time, and time to maximum concentration of our myocardial output TAC. For dimensionality reduction through PCA, we chose to keep the first 5 coefficients. For each variation we also concatenated 9 different patient risk factors and trained a separate

model. These patient risk factors include age, gender, BMI, hypertension, and presence of diabetes. We applied four machine learning regression techniques: 1) Binary regression tree, 2) Ensemble of learners regression, 3) Support vector machine, and 4) Kernel regression to the four sets of summarized data in MATLAB (ver. 2020b). For all methods the data was divided into 70% training and 30% testing cases. Results report RMSE of the tests cases for each estimate using quantitative PET derived flow as our true values.

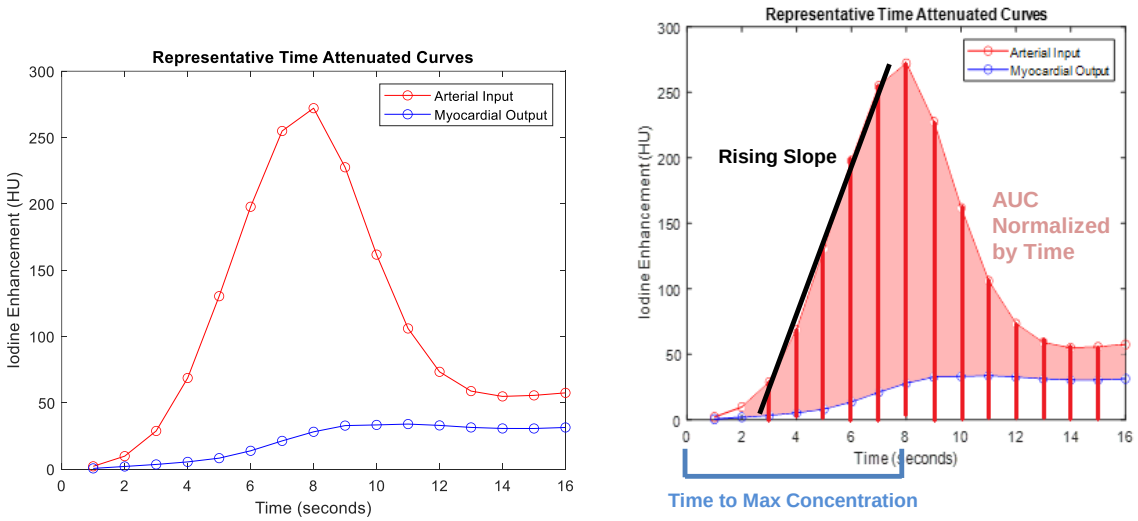

Figure 1. Representative time attenuation curve of 1 region of a 17-segment model (Left). In red is the injected bolus and in blue is the myocardial response. Visualization of the semantic features for data summarization (Right).

1.1.3 Results and Discussion

On average, the MBF estimates for PET were 1.76±1.05 mL/min/g and for the conventional compartmental modeling estimates from DCE-CT were 1.58±0.84 mL/min/g. Table 1 summarizes the model performance for each regression model trained on various TAC data summaries. As a measurement for performance, root mean squared error was calculated for the predicted flows in our test data set. On average, ensemble of learners had the lowest RMSE across all types of predictors, including the best performing model (RMSE = 0.47) which trained on semantic features + risk factors. For comparison, verified dCTP flow estimations using a two

compartmental model yielded an RMSE of 0.80 ml/g/min. The addition of risk factors as predictors had mixed effects. Of the 16 different data summary + model combinations, 8 were improved by the addition of risk factors and 8 worsened. It is interesting to note that semantic features + risk factors gave both the best and worst performing models. Overall, any variation of semantic features estimates outperformed PCA estimates. When only using risk factors and not accounting for any TAC data, all four ML methods yield poor estimates. SVM and kernel models that training on complete TAC curve data yielded lower RMSE than training on summarized data. The opposite is true for binary tree and ensemble models. In general, training on either the full myocardial TAC or with the addition of the input TAC performed better than summarized data. Additionally, the semantic features of rising slope, time to max, and area under the curve normalized by time were found to be a better data summary technique than PCA, having a generally lower RMSE across all models.

Table 1. Root mean squared error (RMSE) of MBF estimates for each regression model and TAC data summary technique. Units in ml/min/g. Best and worst performing approach shown with highlighted cells (green for best, red for worst). For reference, our compartmental model has an RMSE of 0.80.

Model	**Binary Regression Tree**	**Ensemble**	**Support Vector Machine**	**Kernel Regression**	**Compartmental Model**
Myo TAC	0.86	0.72	0.64	0.67	0.80
Myo TAC + Risk Factors	0.89	0.72	0.79	0.76	
Input + Myo TAC	1.23	0.71	0.75	0.72	
Input + Myo TAC + Risk Factors	1.19	0.80	0.71	0.69	
Semantic Features	0.94	0.54	0.78	1.48	
Semantic Features + Risk Factors	1.00	0.47	1.59	0.71	
PCA	0.96	0.66	0.78	0.85	
PCA + Risk Factors	1.00	0.58	0.63	1.10	
Only Risk Factors	1.12	0.77	1.10	0.81	

Across all models, the higher MBF values tended to be underestimated compared to the PET derived estimates. In Figure 2A-B, a Bland-Altman plot of the 2-Compartmental model

derived estimates vs PET estimates is shown. The 2-comp model performs poorly in flows above 2 mL/min/g; overestimating between 2-3 mL/min/g and underestimating at < 3 mL/min/g. Figure 2C-D presents correlation and Bland-Altman plots of the "Semantic Features" data technique + Ensemble Tree, comparing our best fitting model and the PET. The semantic feature technique reduced each TAC to 3 features, for a total of 6 predictors as inputs the regression tree (features from both input function and myocardial response function). Likewise, Figure 2E-F presents results from the "Semantic Features + Risk Factors" approach that included 3 TAC features and 9 risk factors as inputs to the regression tree.

With multi-observation correction, the "Semantic Features" estimates have a mean estimate of 1.55±0.56 mL/min/g. We can see a general linear correlation (Fig. 2C) between PET and our ensemble regression tree model flow estimates, which indicates a moderate agreement between both methods. The Bland-Altman plots also show where the slight overestimation of our model occurs. In Figure 2D, the negative bias seems to increase with respect to higher flows, indicating our model performs poorly, and underestimates more significantly in this area. Both the general linear correlation (Fig. 2E) and underestimation in high flow areas (Fig. 2F) are also seen with the "Semantic Feature + Risk Factors" estimates. The addition of risk factors as predictors increased our tree depth and conversely improved the regression estimates significantly. The linear correlation is much stronger, and the negative bias is much smaller than the "Semantic Features" only estimates.

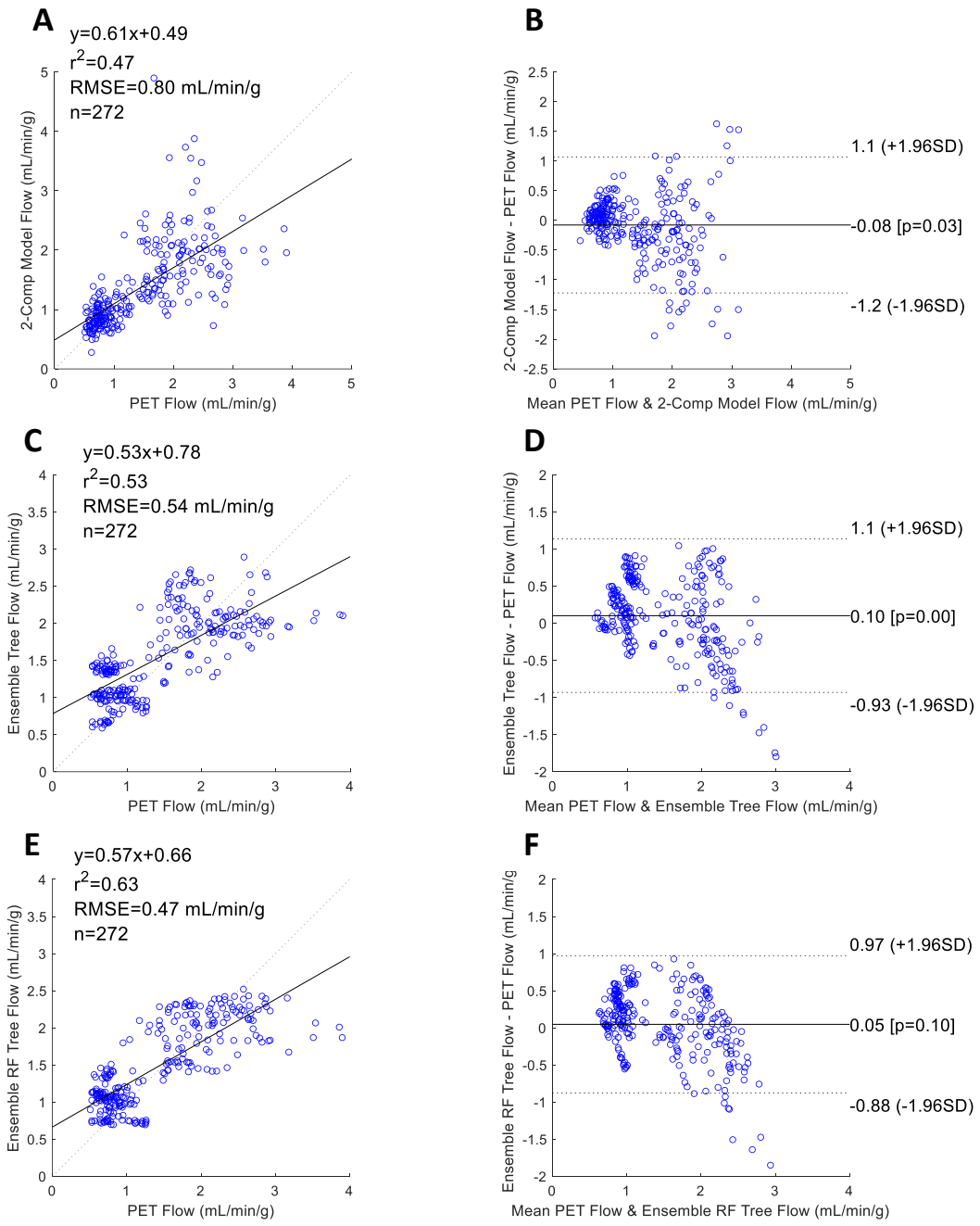

Figure 2. Performance on test set**.** Correlation and Bland-Altman plots of 2-Compartmental model (row 1), Ensemble of Learners Tree regression with Semantic features (row 2) and Ensemble of Learners Tree regression with Semantic features and risk factors (row 3). **A)** Basic correlation plot between 2-compartment model and PET flow estimates. Best fit line, r^2, sum of squared error, and number of samples are reported. **B)** Bland-Altman plot. A mean difference of -0.08 indicates no inherent biases. **C)** Basic correlation plot between ensemble regression tree estimates and PET flow estimates using semantic TAC features. **D)** Bland-Altman plot. A mean difference of 0.1 indicates no inherent biases. **E)** Basic correlation plot of ensemble regression tree estimates using TAC features and nine risk factors. **F)** Bland-Altman plot. A mean difference of 0.05 indicates no inherent biases.

We assessed the performance of four different machine learning regression models using summarized TAC curve and coronary artery disease risk factors as predictors. In general, machine learning provided better MBF estimates compared to conventional compartmental modeling. The use of semantic features in an ensemble regression tree led to estimates with an RMSE of 0.54 ml/min/g, compared to conventional compartmental modeling estimates with an RMSE of 0.80 ml/min/g. The addition of patient risk factors to the TAC data further improved machine learned estimates to 0.47 ml/min/g. Overall, an ensemble regression tree model trained on semantic features such as rising slope, area under the curve normalized by time, and time to maximum concentration had the lowest RMSE. The large variance in both the compartmental model and machine learned estimates can partially be attributed to significant noise in PET MBF estimates. We plan to continue to fine tune our model to allow for better identification of ischemic areas, as well as look into using reduced temporal sampling techniques to reduce radiation dosage.

1.2 Diagnostic Accuracy of Combined Dynamic Myocardial Perfusion CT and Coronary CT Angiography Compared with PET

Estimating myocardial blood flow (MBF) is valuable for diagnosing and risk stratifying myocardial ischemia. Positron emission tomography (PET) is the standard for non-invasive, quantitative MBF measurements. However, its high cost and limited availability have limited its use. Dynamic contrast-enhanced computed tomography perfusion (dCTP) offers a widely accessible approach for MBF measurements offering the potential for similar diagnostic information as PET. In this work, we compare the ischemia detection performance of dCTP and cardiac CT angiography (CTA) to cardiac PET. We propose a new metric (FFR-CTPA) for combining dCTP derived myocardial blood flow and coronary CTA stenosis information. CT derived myocardial flow reserve (MFR) and stress MBF detected regional PET-confirmed ischemia with area under the curve (AUC) of 0.84±0.04 and 0.85±0.04. Combining CTA

information with the MFR and stress MBF in the proposed FFR-CTPA improved the detection of ischemia ($p<0.001$), with AUC of 0.85±0.04 and 0.89±0.03 respectively. The combination of CTA anatomical information with stress MBF yielded the highest detection performance. This work demonstrates that dCTP + CTA can generate better ischemia detection performance than stenosis information or CT-derived flow alone.

1.2.1 Introduction

Assessment of the coronary arteries is often performed with invasive coronary angiography. However, approximately 40-50% of all invasive angiographies do not find evidence of stenosis [26]. Consequently, in low to moderate stenosis risk cases, CT angiography (CTA) is preferred since it is noninvasive [27]. While it offers high sensitivity for CAD, the quality of a CT angiogram can be affected by the motion of heart, presence of arterial calcification, and presence of a coronary stent [28], [29]. Another limitation of CTA is noisy assessment of stenosis in distal coronary arteries [30]. Most importantly, it lacks functional information for a given stenosis, leading to poor evaluation of ischemia [27]. Therefore, many efforts have sought to add functional information, such as myocardial blood flow (MBF) information, to non-invasive CTA exams [31].

Cardiac positron emission tomography (PET) is considered to be the gold standard in quantitative MBF measurements [32]–[34]. Numerous efforts have advanced the use of different myocardial perfusion tracers, such as ^{82}Rb or ^{13}N-ammonia for PET [35], [36]. Despite the validation of these tracers and the clinical tools for generating estimates [37], the cost of the tracers and imaging technology of PET has limited the wide-spread use of cardiac PET perfusion imaging. Measuring MBF using dCTP offers a cheaper, faster, and more accessible alternative over PET [10], [38]. Numerous studies have validated the quantitative accuracy of DCE-CT for MBF measurements and its use in grading ischemia [10], [24], [39]–[41].

In this work, we are among the first to a) evaluate CT myocardial perfusion for the detection of regional ischemia with PET as the reference test and b) use a quantitative scoring metric for the combination of perfusion and anatomical information. Recent efforts have reported the diagnostic accuracy of CT myocardial perfusion imaging compared to different reference standards. Pontone et al. presented a meta-analysis of 77 studies of leading non-invasive tests, including 7 studies of stress CT perfusion and CTA, for the detection of abnormal invasive fractional flow reserve (FFR) [42]. Similarly, Lu et al. conducted a meta-analysis of just dynamic myocardial perfusion CT compared to either another myocardial perfusion imaging modality (SPECT/PET/MRI) or invasive FFR [43]. At the time, their search revealed thirteen prior studies for this purpose, although none of them used PET as a reference perfusion test. Recent work by Nous et al. reported dynamic CT perfusion compared to invasive FFR in a study of 132 patients from 9 centers [44]. Expert readers combined the CT perfusion and CTA information in a non-quantitative, although rigorous, fashion. This leads to perfusion+anatomical information that is not reported on a continuous scale to allow for area under the receiver operating curve assessment and adjustment of sensitivity/specificity performance.

In this work, we combine the functional information of dCTP MBF measurements and anatomical information from CTA to grade ischemia and compare it to quantitative PET. We propose a new scoring method for combining the MBF and CTA information into a continuous variable representative of flow reduction from a stenosis. The dCTP MBF measurements were presented in our previous work that evaluated the quantitative (not diagnostic) accuracy of global MBF estimation of CT compared to PET [24]. This work uses regional MBF estimates and seeks to determine the diagnostic accuracy of CT assessment compared to PET for the detection of regional ischemia. Here, we determine the diagnostic accuracy of the dCTP values and the

combined CT + CTA information.

1.2.2 Materials and Methods

Study Design

Anonymized data, CT-MBF estimation tools, and MBF estimates that we generated in our previous work are publicly available at the Dataverse and can be accessed at https://doi.org/10.7910/DVN/VUP5TC. Details on the study protocol, image acquisition, and patient demographics were previously presented [24]. Briefly, thirty-four patients received a rest and regadenoson stress rubidium-82 PET scan and then within 30 days a dCTP with CTA exam. All CT exams were performed on a Revolution CT scanner (GE Healthcare, Waukesha, WI) with a 16 cm z-axis coverage.

Of the 34 total DCE-CT scans, 5 were excluded due to injection errors or mismatched hemodynamics. All dCTP, PET, and CTA images were aligned along the short axis view and segmented according to the standard American Heart Association 17-region model (Figure 3) [25]. The myocardial blood flow was estimated for each region and modality to provide regional absolute quantitative MBF estimates in units of mL/min/gram. This led to a total of 493-time attenuation curves and segments for comparison between PET and CT.

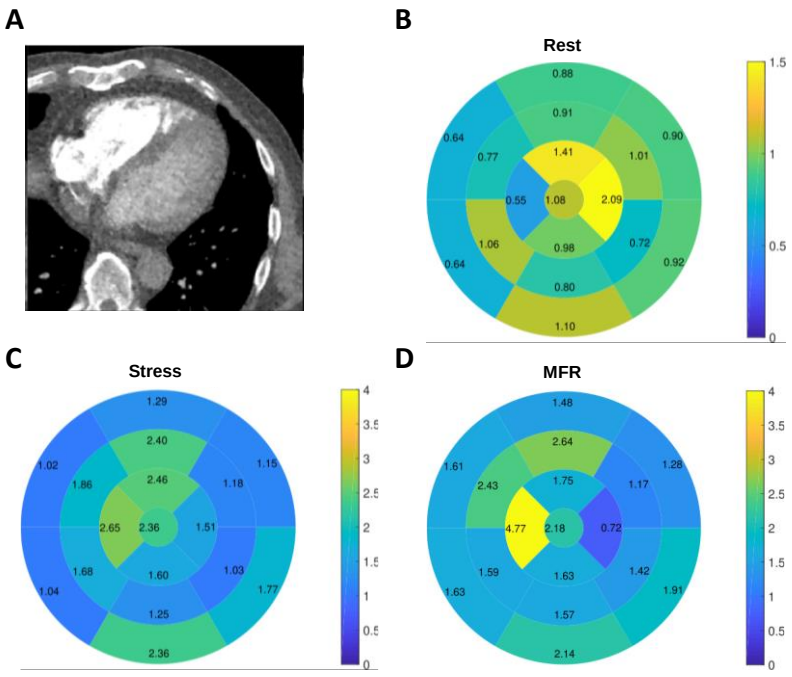

Figure 3. Example frame from a DCE-CT image (A) and example CT-derived MBF (B) Rest, (C) Stress, and (D) MFR estimates for single patient performed at the 17-segment level.

PET MBF estimation

PET estimations were generated using the QPET software (Cedars-Sinai, Los Angeles, CA). Following ischemic definitions proposed by Johnson and Gould, these regional absolute MBF estimates were used to assign each region as either A) normal (stress flow > 1.12 mL/g/min or myocardial flow reserve (MFR) > 2.03) or B) at moderately to definitely ischemic (stress flow < 1.12 mL/g/min and a myocardial flow reserve (MFR) > 2.03) [45]. This definition for each region served as the reference test for evaluating the diagnostic accuracy of the CT-derived information. Their study suggests that the threshold for a binary definition of ischemia vs non-ischemia is a stress flow less than 0.91 mL/g/min and a myocardial flow reserve (MFR) of less than 1.74.

CT MBF estimation

The CT MBF estimates were generated with custom processing using MATLAB (ver

2017b; MathWorks, Natick, MA) and JSim. The left ventricular myocardium was isolated using semiautomated edge detection with manual interaction to account for any interframe motion, when necessary. The median CT number within the myocardium was extracted from each frame over time to generate the myocardial TAC. The median CT number in the descending aorta was extracted for the input function TAC. We used a 2-compartment model that has been previously presented and verified with simulation studies [10], [24]. Driven by the input function TAC, the model was optimized across 4 free parameters (MBF, volume of interstitial fluid, baseline correction, and delay between input and myocardial TAC) to fit the myocardial TAC to generate MBF estimates in units of milliliters per minute per gram. Myocardial flow reserve (MFR) was calculated as the ratio between MBF at stress to MBF at rest.

CCTA stenosis evaluation

Assessment of anatomic CTA vessel information was performed through joint interpretation by a cardiology fellow and cardiologist. One-beat, whole heart axial scans were acquired for all CTA exams with padding from 60-80% of the cardiac cycle, gantry rotation 280ms, tube voltage 120 kVp, and an effective tube current of approximately 500 mA. Images were reconstructed every 5% phase with 0.625 mm slice thickness and standard reconstruction kernel. The CTA interpretation involved the visual match of the coronary arteries to downstream myocardial segments. Each myocardial segment was assigned a percent stenosis ranging from 0 for coronary trees with no apparent stenosis to 100% for an upstream branch with one or more fully occluded stenoses. For analysis purposes, any unevaluated CTA segments due to heavy artifacts were imputed by conservatively assuming the max percent stenosis of the patient. The cardiologists were blinded to the myocardial perfusion information during the CTA interpretation.

Combined dCTP and CCTA Diagnostic Score, FFR-CTPA

dCTP and CTA information were combined to yield a new diagnostic score, Fractional Flow Reserve from CT perfusion and anatomy (FFR-CTPA), and is calculated by summing their individual quantitative estimates in a weighted fashion:

$$S_1 = 0.6\left(\frac{S_p}{\tau_p}\right) - 0.4\left(\frac{S_s}{\tau_s}\right)$$

$$S_{\text{FFR-CTPA}} = \begin{cases} 2, & \text{if} \quad S_1 > 2 \\ 0, & \text{if} \quad S_1 < 0 \\ S_1, & \text{otherwise} \end{cases} \qquad \textbf{(1)}$$

Where $S_{FFR\text{-}CTPA}$ indicates our proposed diagnostic score, which combines the patient-specific myocardial perfusion estimate, S_p, and percent stenosis, S_s. This score includes constants: τ_p, which is the perfusion threshold for ischemic vs. non-ischemic regions, and τ_s, which is the percent stenosis threshold for ischemic vs. non-ischemic regions. This score can be calculated for the three different measures of perfusion: resting state, stress state, or myocardial flow reserve.

The new score incorporates three concepts: 1) the contribution of the perfusion and stenosis information is normalized (divided by) the threshold for ischemia detection for that information, 2) perfusion information is slightly more predictive of ischemia than stenosis information and therefore receives more weight; 3) the score is clipped to a range of 0-2 to enable easy interpretation.

A lower $S_{FFR\text{-}CTPA}$ suggests a higher severity of ischemia; as MBF or MFR decreases or percent coronary stenosis increases $S_{FFR\text{-}CTPA}$ will decrease. The FFR-CTPA score was calculated for rest, stress, and MFR perfusion states, requiring different threshold, τ_p, values in the calculation. The relative weighting of each contribution was adjusted to achieve reasonable performance on the data and provide round numbers for ease of implementation. Specifically, the weighting was changed in intervals of 10% until the highest AUC was achieved. This led to the

dCTP information receiving 60% weight in the new score and CTA stenosis a 40% weight. Values for each variable are given in Table 2.

Table 2. Example frame from a DCE-CT image (A) and example CT-derived MBF (B) Rest, (C) Stress, and (D) MFR estimates for single patient performed at the 17-segment level.

Parameter	Symbol	Rest MBF	Stress MBF	MFR
Constant Values				
Threshold for Ischemic vs Non-Ischemic, Perfusion	τ_p	0.50 mL/g/min	0.91 mL/g/min	1.74
Threshold for Ischemic vs Non-Ischemic, Percent Stenosis	τ_s	70%	70%	70%
Summary of segments				
Myocardial Perfusion Estimate	S_p	0.96±0.36 mL/g/min	2.04±0.89 mL/g/min	2.30±1.60
Percent Stenosis	S_s		46±36%	
Combined Diagnostic Score	FFR-CTPA	0.87±0.41	1.05±0.51	0.52±0.39

Comparative and Statistical Analysis

We used an unpaired parametric t-test to determine group differences in MBF between ischemic and non-ischemic regions. ROC analysis was performed using dCTP diagnoses as classifier predictions and PET diagnoses as true labels. Area under the ROC (AUC), accuracy at 90% sensitivity, and specificity at 90% sensitivity were all reported. An unpaired, two sample, t-test was used to determine the group differences between AUC, accuracy, and specificity of rest vs stress vs MFR. Similar analysis was performed using a 3-region model, where the original 17 regions were regrouped into three regions based on the supply beds of the three main coronary arteries: left anterior descending, right coronary artery, and left circumflex. Bootstrapping methods were employed, with replacement, to generate error bar on all performance measures. A total of 1000 resamples were used for each error bar.

1.2.3 Results

Figure 4 presents boxplots of dCTP derived MBF estimates grouped according to PET diagnosed non-ischemic vs ischemic. There is a significant separation between non-ischemic and

ischemic dCTP derived MBF values for resting, stressed, and MFR (top row). This separation increases using our combined score, $S_{FFR\text{-}CTPA}$, suggesting better stratification of ischemia (bottom row). The bottom left graph shows the values from CTA stenosis information alone.

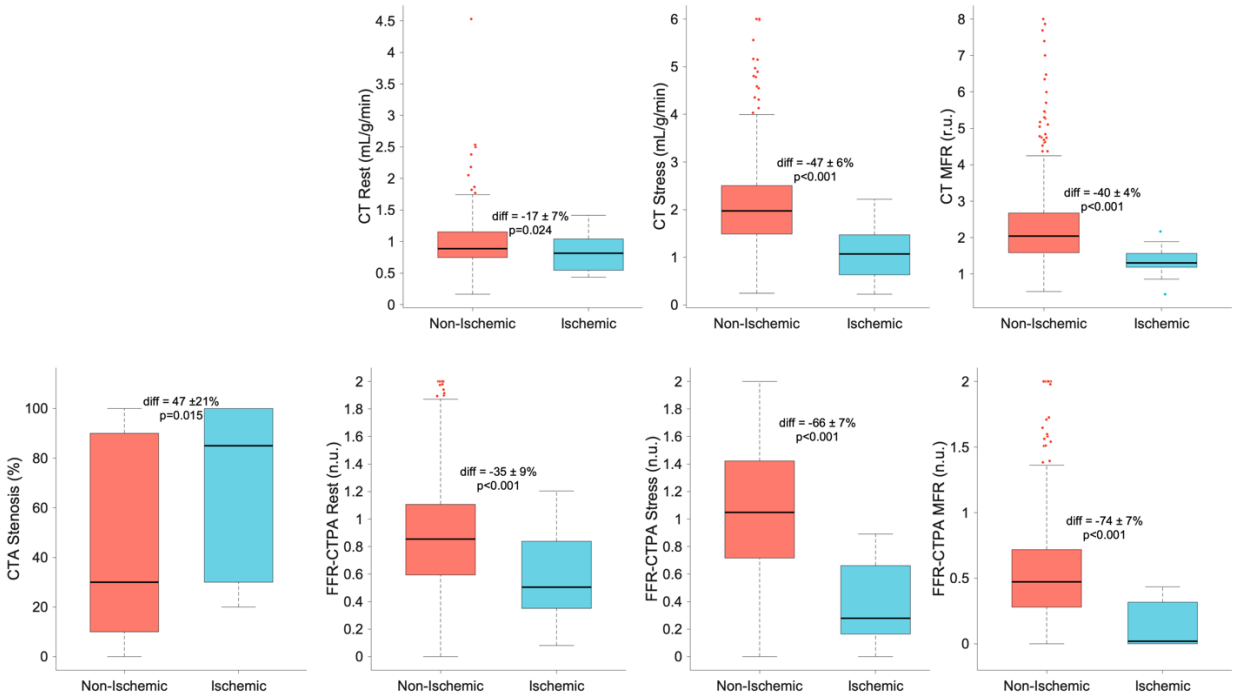

Figure 4. Comparison of the CT derived measures grouped according to PET diagnosed non-ischemic vs ischemia regions. The first row summarizes measures from CT flow information (rest, stress, MFR) and bottom row summarizes measures from CT anatomy (first column) and combined FFR-CTPA using rest, stress, and MFR respectively. The percent difference between groups and p-value are presented on these box plots.

In Figure 5A a ROC curve was constructed for the prediction of ischemia using dCTP derived MFR, rest MBF and stress MBF measurements. Here, we see that stress MBF produces the high AUC (0.85), suggesting decent diagnostic accuracy. In Figure 5B, a similar ROC curve was constructed using $S_{FFR\text{-}CTPA}$ scores to predict ischemia. For all perfusion estimates, the combined score increased AUC. Particularly, $S_{FFR\text{-}CTPA}$ calculated using stress MBF is the best at diagnosing ischemic regions. For reference, a random predictor of ischemia would yield an AUC of 0.5.

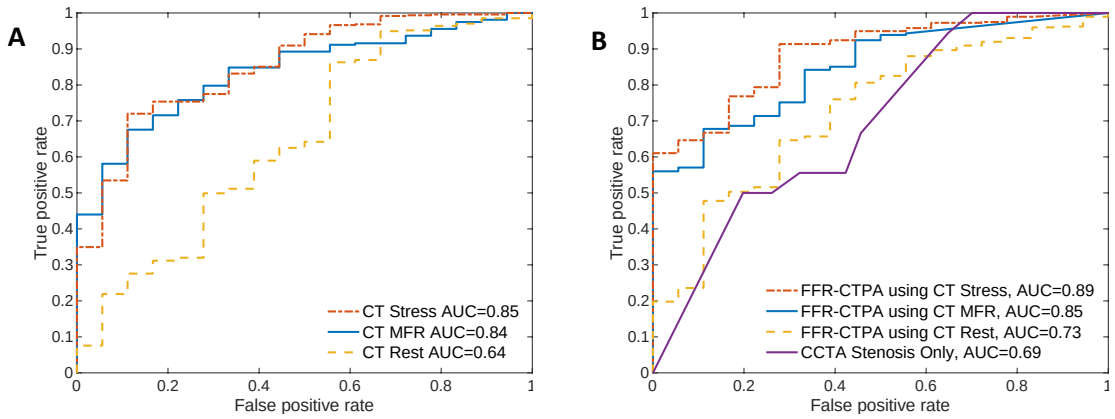

Figure 5. Receiver Operating Characteristic (ROC) curves for CT-derived regional myocardial blood flow estimates (A) and for CT-derived flow estimates combined with stenosis information (B) for the diagnosis of ischemia. ROC was performed on 493 regions, 18 of which are labeled as ischemic via PET.

Table 3 displays summary statistics for dCTP and $S_{FFR\text{-}CTPA}$ results. Rest MBF performed poorly as a detector, with an AUC of 0.64. The stress MBF threshold to achieve 90% sensitivity was 1.93 mL/g/min and is much higher than the 0.91 mL/g/min PET threshold, indicating general overestimation of CT MBF. Both MFR and stress MBF were better classifiers of ischemia, having an AUC of 0.84 and 0.85, respectively. Stress MBF achieved an accuracy at 90% sensitivity score of 0.54 and a specificity at 90% sensitivity of 0.53 where MFR achieved slightly higher performance (0.59 and 0.58, respectively). Stenosis information alone performed relatively poorly with an AUC of 0.69. Combining CTA information with the CT classifiers significantly improved the classification performance over the CT only classifier. The MFR, rest MBF, and stress MBF AUC all increased to 0.85, 0.72, and 0.89, respectively. The accuracy and specificity at 90% sensitivity also improved (Table 3) with the addition of CTA information for both rest and stress MBF. Interestingly, MFR did not benefit from the addition of CTA information as much as rest or stress MBF, despite being a combination of the two metrics. Using bootstrapping with replacement, there was sufficient evidence that all reported mean AUC, accuracy, and specificity estimates are different from each other ($p<0.0001$) except for the CT Stress MBF compared to

FFR-CTPA MFR measures.

Table 3. Summary statistics of detection analysis for CT derived estimates for 17-segment information.

Method	AUC*	Accuracy at 90% Sensitivity	Specificity at 90% Sensitivity	Threshold to achieve 90% Sensitivity
CT Rest MBF	0.64 ± 0.07	0.24 ± 0.02	0.22 ± 0.02	1.19 ± 0.14 mL/g/min
CT Stress MBF	0.85 ± 0.05	0.54 ± 0.02	0.53 ± 0.02	1.93 ± 0.29 mL/g/min
CT MFR	0.84 ± 0.04	0.59 ± 0.02	0.58 ± 0.02	1.89 ± 0.19 r.u.
CCTA Stenosis Only	0.69 ± 0.06	0.37 ± 0.02	0.35 ± 0.02	30.00 ± 5.59 %
FFR-CTPA Rest	0.72 ± 0.06	0.26 ± 0.02	0.24 ± 0.02	1.13 ± 0.06 n.u.
FFR-CTPA Stress	0.89 ± 0.03	0.66 ± 0.02	0.65 ± 0.02	0.85 ± 0.11 n.u.
FFR-CTPA MFR	0.85 ± 0.04	0.58 ± 0.02	0.57 ± 0.02	0.42 ± 0.06 n.u.

The final column of Table 3 indicates the threshold for that measure to operate at a high sensitivity. For example, we would classify anything below 1.93 mL/g/min as ischemic for dCTP derived stress MBF. Likewise, anything below 0.85 would classify as ischemic for our stress $S_{FFR\text{-}CTPA}$. With this threshold, we are expected to detect 90% of all disease and have a specificity of 0.65.

To see if diagnostic performance is a function of each coronary artery region, we summarized the stress MBF and stress $S_{FFR\text{-}CTPA}$ accuracies for all 17 regions and grouped them together according to the coronary arterial distribution proposed by the American Heart Association (Figure 6). Table 10 shows relevant metrics including AUC, accuracy at 90% sensitivity, specificity at 90% sensitivity. We see in Figure 6B that our stress $S_{FFR\text{-}CTPA}$ information produced accuracies at 90% sensitivity of 0.69, 0.66, and 0.62 for LAD, RCA, and LCX, respectively. This is an improvement over stress MBF alone, who had accuracies of 0.59, 0.50, and 0.53, respectively. AUC is slightly higher in RCA and LCX regions compared to LAD.

Table 4. Summary of detection analysis for Stenosis only, Stress MBF, and FFR-CPTA with Stress by major coronary bed.

Method	Region	Prevalence of ischemia each segment	AUC	Accuracy at 90% Sensitivity	Specificity at 90% Sensitivity
CT Stenosis	LAD	8/203	0.58 ± 0.10	0.51 ± 0.03	0.52 ± 0.04
	RCA	6/145	0.77 ± 0.08	0.58 ± 0.04	0.57 ± 0.04
	LCX	4/145	0.78 ± 0.09	0.57 ± 0.04	0.55 ± 0.04
CT Stress MBF	LAD	8/203	0.78 ± 0.07	0.59 ± 0.03	0.58 ± 0.04
	RCA	6/145	0.96 ± 0.03	0.50 ± 0.04	0.48 ± 0.04
	LCX	4/145	0.82 ± 0.11	0.53 ± 0.04	0.52 ± 0.04
FFR-CTPA Stress	LAD	8/203	0.82 ± 0.06	0.69 ± 0.03	0.68 ± 0.03
	RCA	6/145	0.99 ± 0.01	0.66 ± 0.04	0.64 ± 0.04
	LCX	4/145	0.89 ± 0.05	0.62 ± 0.04	0.61 ± 0.04

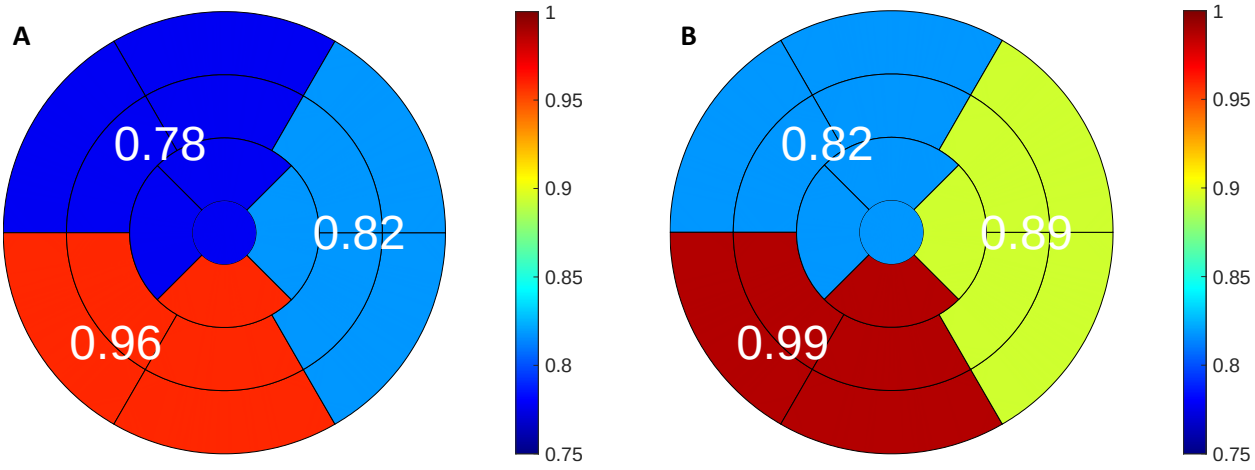

Figure 6. Polar maps of area under the ROC (AUC) for stress MBF (A) and FFR-CTPA (B) using stress MBF information at the 3-region level. Individual segments (1-17) were grouped together according to their common coronary artery region. In this display, upper left segments are supplied by the left anterior descending artery, right by the left circumflex, and lower left by the right coronary artery.

This study demonstrated that MBF estimates derived from stress dCTP combined with CTA information can detect regional myocardial ischemia as identified by PET with an AUC of 0.89± 0.03. We demonstrate that by combining anatomical information about upstream stenoses with myocardial perfusion information will improve detection performance. The combined score specificity and accuracy at 90% sensitivity suggests that dCTP-derived measures of ischemia can

reliably detect ischemia as confirmed by cardiac PET. The proposed new score, FFR-CTPA, offers the best performance when calculated with stress MBF. In high-sensitivity mode (@ 90% sensitivity), $S_{FFR\text{-}CTPA}$ using stress flow achieves a specificity of 0.65 ± 0.02, which is superior to specificity estimates of 0.29-0.61 presented by Meijboom et al. who evaluated the diagnostic accuracy of anatomical assessment with modern CCTA compared against a functional reference test, invasive FFR [28].

We present preliminary evidence that the detection performance may vary slightly with different coronary beds. The LAD supplied myocardial bed had lower AUC, accuracy, and specificity compared to the RCA and LCX regions for all methods evaluated (Table 4). For example, for FFR-CTPA using stress flow, the AUC of the LAD was 0.82 compared to 0.99 and 0.89 of the RCA and LCX respectively. This discrepancy may be caused by the increased difficulty of taking LAD CTA measurements as well as higher levels of average motion in the area, both contributing to higher levels of noise. This also may highlight potential errors in our reference test, the PET estimates of flow; previous studies have demonstrated that patient motion and, to a lesser extent, attenuation correction mis-alignment can lead to large regional errors in PET estimated flow [46].

Study limitations

This proof-of-concept study of a new metric for combining perfusion and stenosis information included only 29 patients. This small sample size, along with low disease prevalence, suggests that our reported performance measures have high error bars. Additional research with a larger set of patients is needed. Among the 493 total segments analyzed, PET only identified 18 as definite ischemic (3.7% prevalence), distributed across 7 patients. Our best performing stress $S_{FFR\text{-}CTPA}$ method only missed one ischemic region, but overcalled many regions leading to a high

sensitivity (94%) but low specificity and accuracy (64% and 66%, respectively). Additionally, our DCE-CT derived MBF were generally overestimated, partially attributing to the high false positive rate. While quantitative PET is the gold standard in MBF measurements and ischemia detection, it remains a noisy modality which greatly affects our ground truth values. Additionally, we assumed that each of the 17-segments were independent, but there is likely intra-patient correlations of these measures that were not accounted for in the detectability analysis.

1.3 Segmentation of Porous Implantable Polymeric Scaffolds for µCT Monitoring

To assess the safety and efficacy of implantable biomedical devices, longitudinal radiological monitoring is necessary for risk evaluation. However, polymeric devices are poorly visualized with clinical imaging, hampering efforts to use diagnostic imaging to predict failure and enable intervention. Combining contrast agents with these biomedical devices, either through coating methods or direct mixing with the polymer, offers a potential solution to poor image quality. Direct mixing is more favorable for degradation studies, but the effect of the contrast agents may alter the device's mechanical properties. Here, we describe nanoparticle-doped biomedical devices (phantoms), created from 0–40 wt% tantalum oxide (TaOx) nanoparticles in polycaprolactone and poly(lactide-co-glycolide) 85:15 and 50:50, representing non, slow, and fast degrading systems, respectively. We run a degradation study of 20 weeks in length in multiple simulated physiological environments: healthy tissue (pH 7.4), inflammation (pH 6.5), and lysosomal conditions (pH 5.5), while mass and gross volume loss are monitored. We show that an optimal range of 5–20 wt% TaOx nanoparticles balances radiopacity requirements with implant properties, facilitating next-generation biomedical devices.

1.3.1 Introduction

Polymers are commonly used for biomedical devices due to several advantageous properties. Namely, they offer high biocompatibility, tunable mechanical properties, and are

generally easy to manufacture [47]. This has led to proliferation of implantable biomedical devices in research and clinical scenarios. Despite their frequent use in the clinic, implants made from polymers fail for a number of reasons such as wear, tearing, migration, and infection [48]. With a growing concern for the complications due to device failure, there exists an increased need for a clinical methodology for in situ monitoring of device status following implantation [48]. However, most polymeric devices offer no radiological contrast mechanism for clinical diagnostic imaging, and therefore radiologists cannot monitor the integrity of the device prior to catastrophic failure. Incorporating contrast agents for radiological monitoring of biomedical devices would be a significant step in prevention of emergency device failures.

The widespread use and availability of computed tomography (CT) makes it an excellent modality for device monitoring. While CT has difficulty distinguishing soft tissues compared to magnetic resonance imaging (MRI) and exposes patients to small amounts of radiation, it remains favorable due to its low cost and high signal-to-noise ratio [49], [50]. To tailor polymeric devices for CT monitoring, we must modify or incorporate polymers with contrast agents specific for CT [51]. In other words, we must make them radiopaque while keeping in mind the possibility of releasing cytotoxic elements during degradation [52], [53]. Tantalum oxide (TaOx) nanoparticles, in particular, are biocompatible in vivo with superior CT contrast over traditional iodinated compounds, and can further be incorporated into polymeric matrices for use as biomaterials [54]–[57]. In previous studies, it was shown that TaOx integrated polymer phantoms were easier to identify than phantoms without TaOx [50]. However, more research needs to be done to evaluate the impact of TaOx on the mechanical properties of the polymer. Namely, the nanoparticle should not affect the mechanical stability or material properties of the device while making it radiopaque.

1.3.2 Materials and Methods

The study utilized three types of biocompatible polymers: PCL, PLGA 50:50, and PLGA 85:15. PCL (Sigma Aldrich) had a molecular weight average of 80 kDa. PLGA 50:50 (Lactel/Evonik B6010-4) and PLGA 85:15 (Expansorb DLG 85–7E, Merck) were both ester terminated and had a weight average molecular weight between 80 and 90 kDa, to minimize the effects of polymer chain length on the degradation rate [58]. The polymers were solubilized in suspensions of TaOx (spherical, 3–9 nm in diameter) in dichloromethane (DCM, Sigma-Aldrich). A degradation study was conducted in vitro for 20 weeks and in vivo for 5 weeks.

Phantom Manufacture

The detailed protocol for polymer preparation and phantom manufacture can be found here: https://doi.org/10.1002/adhm.202203167. Briefly, PCL or PLGA were solubilized in TaOx nanoparticles in DCM at 8 and 12 wt%, respectively. Proportions were calculated so that the final scaffold will be tunable 0-40 wt% TaOx. Sucrose (Meijer) was added to the suspension, calculated to be 70 vol% of the polymer + nanoparticle mass in solution, followed by NaCl (Jade Scientific) at 60 vol% of the total polymer + nanoparticle volume. The suspension was vortexed for 10 min and pressed into a silicon mold that was 4.7 mm diameter, and 2 mm high. After air drying, phantoms were removed, trimmed of excess polymer, and then washed for 2 h in distilled water, changing the water every 30 min to remove sucrose and NaCl. Washed phantoms were air-dried overnight and stored in a desiccator prior to use. This process yielded micro-porous (<100 μm) scaffolds that mimics tissue properties, allowing for nutrient diffusion and cell and tissue infiltration.

Micro-Computed Tomography

All tomography images were obtained using a Perkin-Elmer Quantum GX. At every time point, groups of three phantoms were imaged at 90 keV, 88 μA, with a 25 mm field of view at a

50 µm resolution. After the acquisition, individual phantoms were sub-reconstructed using the Quantum GX software to 12–18 µm resolution. Phantoms used for serial monitoring were imaged on day 0 prior to hydration for pore size analysis (Supporting Information) and imaged again 24 h after hydration with buffer. Throughout the remainder of the study, all groups were imaged every week after changing the buffer media.

In vivo µCT on mice was performed at 90 keV, 88 µA. At each time point, two scans were taken of the phantoms, 1) 72 mm field of view (14 min total scan time) at 90 µm resolution and 2) 36 mm field of view (4 min total scan time) at 20–50 µm resolution. In the subcutaneous implantation, both phantoms could not be captured in a single higher-resolution scan, so two scans were taken, one centered on each implant. During acquisition, mice were anesthetized using an inhalant anesthetic of 1–3% Isoflurane in 1 L min^{-1} oxygen. Mice were scanned immediately post-implantation, on day 1 post-implantation, and at day 7 and week 5 post-implantation. Total cumulative radiation dosage was 14–19 Gy over 5 weeks.

Tomography Image Analysis

From the tomography scans of phantoms, several parameters were quantified. Analysis of the polymer matrix component of phantoms with 20 and 40 wt% TaO_x was performed using custom software developed with MATLAB (vR2021b, Mathworks, Natick, MA) on µCT sub-reconstructions. Properties such as scaffold thickness, diameter, porosity, average pore diameter, average pore volume, and mean attenuation were analyzed. From this, the percent porosity of the phantoms was calculated as the percentage of the gross volume not occupied by the matrix. From the diameter and thickness, a "gross volume" was defined as the volume occupied by a solid cylinder with the corresponding thickness and diameter. Subsequently, "scaffold volume" is the

gross volume subtracted by the total pore volume. Phantoms with 5 wt% TaO_x could not be radiographically distinguished from the background.

Before segmenting the polymer from the background, the image was preprocessed by using an adaptive histogram equalization technique to enhance contrast [59]. After, Otsu's binary segmentation method was used to create a rough mask of the volume [60]. An adaptive thresholding method was then employed to segment the polymer within the rough mask from the background [61]. The resulting volume was cleaned up using erosion and dilation operations.

Adaptive histogram equalization works by applying the normal histogram equalization algorithm on local, non-overlapping regions of an image. Histogram equalization can enhance the contrast of images by redistributing the pixel intensities more evenly. Let us first assume an $i\ x\ j$ image f of pixels ranging from 0 to $L - 1$. Let us denote p as the normalized histogram of f with a bin for each pixel value (often 2^8 or 2^{16}). So,

$$p_n = \frac{number\ of\ pixels\ with\ intensity\ n}{total\ number\ of\ pixels} \quad where\ n = 0 \ldots L - 1 \tag{2}$$

Then, the histogram equalized image g can be defined by

$$g_{i,j} = floor((L-1)\sum_{n=0}^{f_{i,j}} p_n \tag{3}$$

Meaning for every pixel bin the cumulative pixel intensity are multiplied by the pixel value, then rounded down. This intuitively makes sense; when we normalize g, bins with few pixels will be weighted higher and bins with many pixels will be weighted lower. To apply this in adaptive fashion, we first split the image into non-overlapping regions and simply apply the algorithm to each region individually before reconstructing the image.

Similarly, adaptive thresholding works by applying a thresholding algorithm (often Otsu) on a local level. Otsu binary thresholding works by minimizing the intra-class variance, defined as a weighted sum of variances of the two classes, foreground and background. The algorithm goes through every possible threshold value *t* (0-255 for an 8-bit image), and calculates the class probability using:

$$\mu_{background} = n\frac{\sum_{n=0}^{t-1} p_n}{\sum_{i=0}^{t-1} p_n} \quad \textbf{(4)}$$

$$\mu_{foreground} = n\frac{\sum_{i=t}^{L-1} p_n}{\sum_{i=t}^{L-1} p_n} \quad \textbf{(5)}$$

Intuitively, the threshold *t* that maximizes the difference of mean pixel intensity of the foreground and background is chosen. To make this algorithm adaptive, we apply it on local regions just like the adaptive histogram method.

1.3.3 Results and Discussion

Shown in Figure 7 is the result of the segmentation algorithm. Top row are y and z slices of the µCT volume and the bottom row is the scaffold mask. Results of analyzing the masks are as follows: All phantoms had an interconnected porosity, with a mean pore size between 350 and 400 µm. Pore walls show generally even dispersion of the TaOx (Figure 7, top row), and the homogeneous dispersion ensured that no regions of the polymer matrices had significantly different material properties or X-ray attenuation. As expected, the mean attenuation of the scaffold is dependent on the TaOx wt%. We use this mask to compute the gross volume of the scaffold over time.

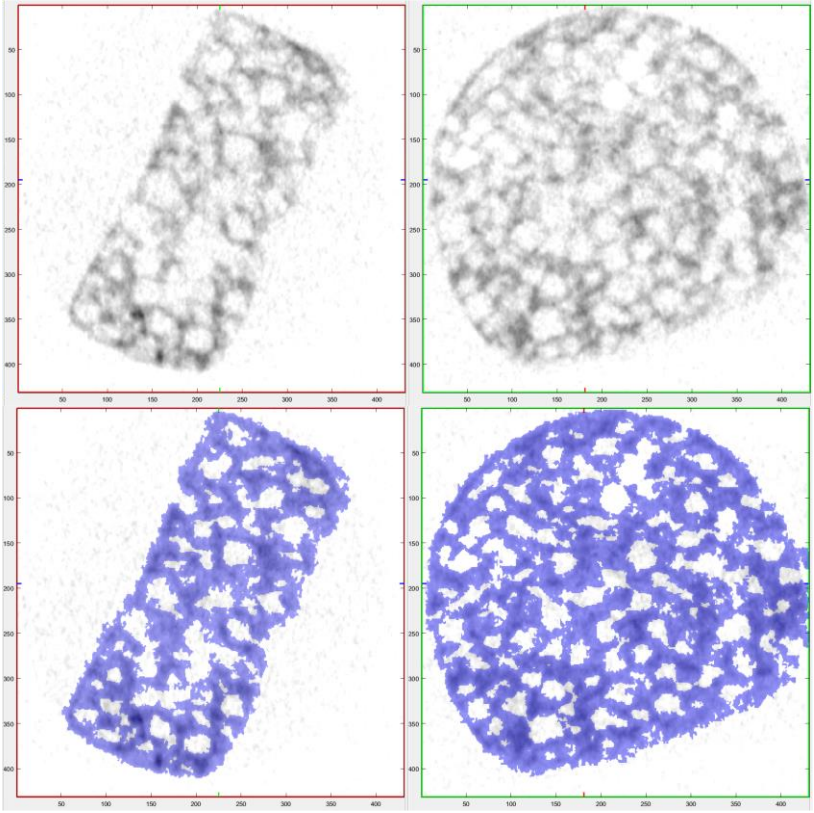

Figure 7. Example of generated scaffold masks from uCT images. An increase in TaOx incorporation increased mean attenuation of the phantom. Scaffolds of TaOx 10-40 wt% for PLGA and PCL could be segmented easily. The attenuation 5 wt% was too low to be differentiated from the background.

Plotting the gross volume alone shows a very clear trend in phantom volume changes (Figure 8e1-f1). We also see from the mean attenuation that radiopacity lasts for at least 20 weeks (Figure 8e3-f3). Due to mechanical properties of PLGA, it degrades rapidly compared to PCL, and even more so in acidic environments [58], [62], [63].

Designing implantable biomedical devices to be radiopaque is an important property to consider. The radiopacity allows physicians and researchers to evaluate and monitor in real time the structural integrity of implantable devices and therefore predict device failure. Here, a novel radiopaque TaOx nanoparticles combined PCL or PLGA scaffold is proposed. We demonstrate that the addition of TaOx nanoparticles enables in situ monitoring of gross phantom features (overall volume, location) using μCT. Importantly, within the range of 5–20 wt% TaOx, the

radiopacity of phantoms was maintained over 20 weeks. Monitoring size and attenuation properties enabled in vivo assessment of the environmental impact on the scaffold. We show that lower pH environments and high nanoparticle content (>20 wt% TaOx) increased degradation rate and decreased structural integrity and mechanical stability. This study represents a significant step toward incorporating in situ monitoring into the next generation of implantable devices.

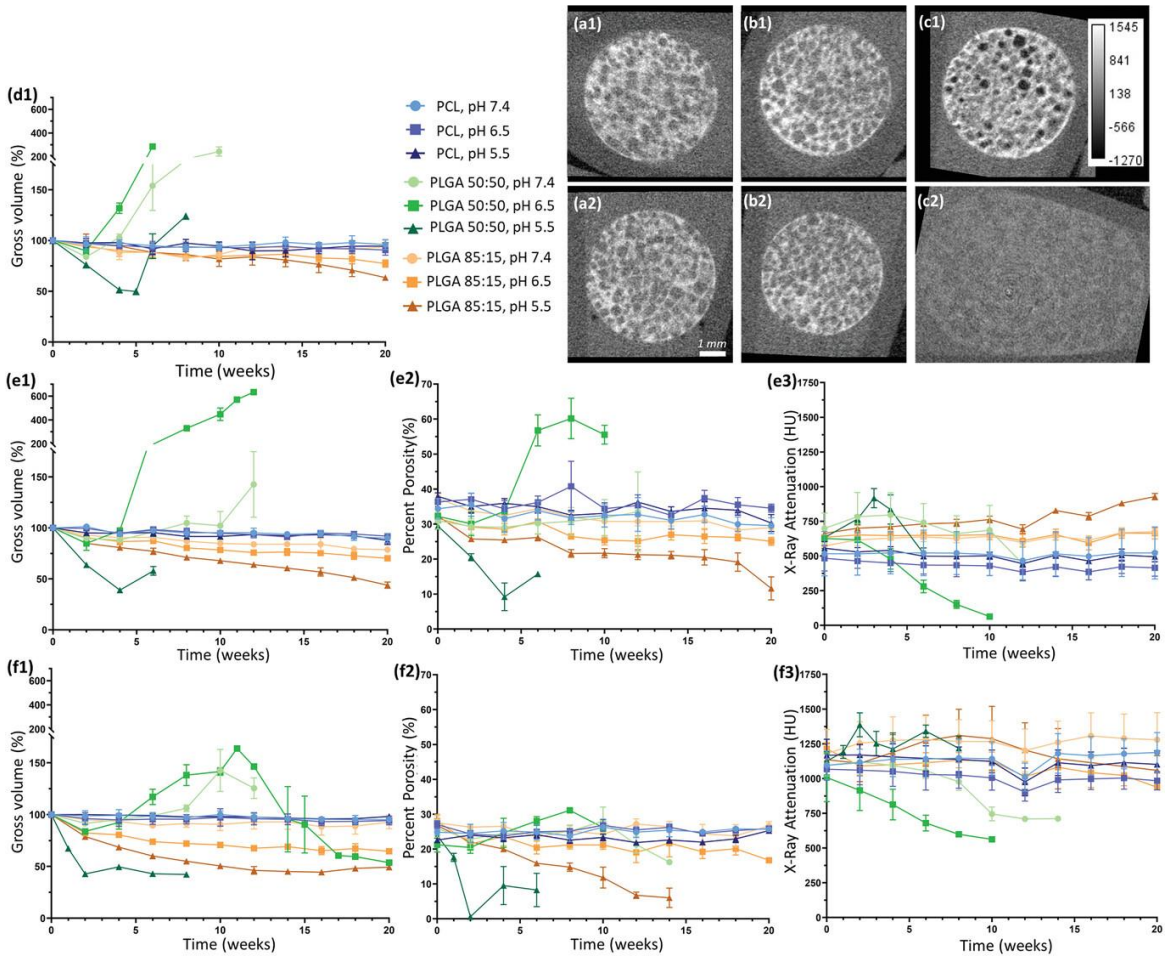

Figure 8. Phantoms with TaOx nanoparticles could be monitored for 20 weeks without loss of radiopacity due to particle leaching. a–c) This allowed for visual monitoring of changes to phantom shape and porosity, as illustrated by CT images from 1) day 1 and 2) 6 weeks: a) PCL+20 wt% TaOx, b) PLGA 85:15 + 20 wt% TaOx, and c) PLGA 50:50 + 20 wt% TaOx. During degradation, significant changes occurred within the phantoms: 1) gross phantom volume, 2) percentage porosity, and 3) X-ray attenuation. TaOx incorporation ranged from d) 5 wt% TaOx, e) 20 wt% TaOx, and f) 40 wt% TaOx. At 5 wt% TaOx, only the gross volume of the phantoms could be quantified, as the matrix could not be segmented from the background. Scale (a–c): 1 mm; HU window is consistent for all images. Data reported as mean ± SEM. Figure courtesy of [2] under CC BY-NC-ND 4.0.

CHAPTER 2: INTRODUCTION

The concept of machine learning and artificial intelligence may sound like a recent development, but it has its roots in the early days of computing. In 1943, neurophysiologists Warren McCulloch and Walter Pitts first described how neurons communicate with each other, laying down the foundation of neural network architecture [64]. In 1949, Donald Hebb developed a model based on the idea of neural plasticity, where connections between neurons can change depending on the feedback it receives [65]. Rosenblatt is credited with building the first perceptron in 1958, a machine designed for image recognition and capable of distinguishing basic patterns [66]. After decades of research, we now have machine learned models nearing human levels of performance in certain tasks. In the medical field, AI has already demonstrated capability in diagnosis, pathology detection, and risk assessment, and is already impacting clinical decision making.

However, machine learned models are far from perfect. The limited use of AI in the medical field is a testament to how difficult certain tasks can be. One major limitation of AI is the need for large quantities of data. The volume of data has a major impact on the performance of the model; in general, higher volume training datasets include greater diversity, enabling better and more generalizable performance. For medical imaging tasks, it is difficult to curate large volumes of data. To partially solve this issue, data augmentation techniques have been proposed. By supplementing our training set with augmented data, we can synthetically add diversity and therefore improve performance. This chapter has four main aims: (1) a high-level overview on the basics of neural networks, (2) how neural networks are tailored to allow for object detection, classification, and segmentation tasks, (3) how we apply these models for rib fracture detection

tasks, (4) data augmentation techniques, and (5) data augmentation applications in the medical imaging field.

2.1 Basics of Neural Networks

A neural network (NN) is a machine learning method that is designed to mimic the human brain. Individual neurons in our brain collect electrochemical signals through dendrites and then pass on the signal through the axon. The axon splits up into thousands of branches with a structure called a synapse at each end. Depending on the input signal, the synapse may release neurotransmitters that inhibit or excite the next neuron. To learn which neuron is correctly inhibited or excited, synapses receive feedback and adjust its behavior accordingly. Based off this understanding, NNs are constructed around a basic unit called a node (Figure 9). Nodes behave similarly to a neuron, where it can activate, propagate a signal, and receive feedback to learn. We structure multiple neurons in groups called layers. An individual node in one layer is connected to every node in the next layer. The connections have weights attached to them that dictate the importance of the input information passed to that connection. The weighted inputs are then added with a bias and processed through an activation function to determine which downstream nodes should be activated. The weights and biases can be modified depending on the feedback it receives. The feedback is evaluated using a loss function. The final layer in our network is called the output layer and may contain a single or multiple nodes depending on the type of output we are expecting [67]–[69]. Here, we will describe more in depth how activation functions and loss functions operate, and how weights are updated using a process called back propagation.

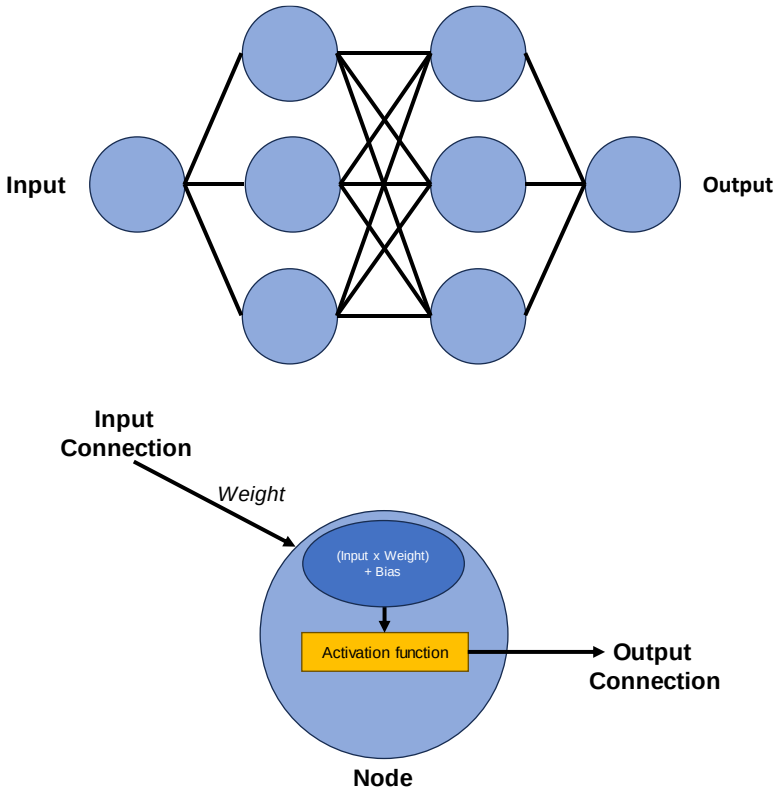

Figure 9. Example of a simple neural network. (Top) A single input node goes through two layers and then an output node. Each connection between nodes has a weight assigned to it. (Bottom) The input, weight, and bias of a node is then used by an activation function to calculate if the next node will be activated.

2.1.1 Activation Functions

At the most basic level, an activation function calculates the output of a node given a set of inputs. It decides whether a downstream node is activated or not. The most commonly used activation functions are non-linear activation functions. If a NN uses a linear activation function, it is equivalent to a regular linear regression model [70]. While a simple linear regression model is easy to solve, it often lacks the complexity necessary to model real world data. The non-linearity of activation functions allows NNs to model complex relationship between nodes. Table 5 highlights several types of non-linear activation functions, each with its own pros and cons.

Table 5. Common activation functions used in neural networks.

Function	Value Range	Pros	Cons
Sigmoid	0,1	• Gives you a smooth gradient while converging. • Gives a clear prediction (classification) with 1 & 0.	• Prone to vanishing gradient problem. • Not a zero-centric function. • Computationally expensive function (exponential in nature)
Tanh	-1, 1	• Zero-centric • It is a smooth gradient converging function.	• Prone to Vanishing Gradient function. • Computationally expensive function (exponential in nature)
ReLu	0, ∞	• Can deal with vanishing gradient problem. • Computationally inexpensive function (linear in nature).	• Not a zero-centric function. • Gives zero value as inactive in the negative axis.
Leaky ReLu	-∞	• Same as ReLu, except it gives some small value instead of 0 in the negative axis.	• Same as ReLu
Binary Step	0,1	• Gives a clear prediction (classification) with 1 & 0. • Zero-centric	• Only supports binary classification

Choosing the correct activation function is vital to the success of the model. Currently, the ReLU (Rectified Linear Unit, [71]) and leaky ReLU activation functions [72] are most commonly used for the hidden layers in deep learning models as it avoids the vanishing gradient problem that sigmoid and tanh functions have, and it converges approximately 6 times faster [73]. Choosing the right function for the output layer, is slightly more complicated. Generally, in a regression problem, we use the linear (identity) activation function with one node. In a binary classifier, we use the sigmoid activation function with one node. In a multiclass classification problem, we use the softmax activation function with one node per class. In a multilabel classification problem, we use the sigmoid activation function with one node per class [74]. Generative adversarial networks (GANs), and other image generating networks generally uses Tanh. After choosing an appropriate activation function for a neural network, the next important consideration is selecting an appropriate loss function.

2.1.2 Backpropagation and Loss Functions

Neural networks "learn" by adjusting the weights and biases of the connections between nodes. To know how much we need to adjust the weights and biases by, we need some way to determine how well our model is doing. To do this we use a loss function. Also called a cost function, a loss function compares your model's current prediction with the actual value. Our aim is to minimize the loss function, that is to minimize the difference between our predictions and ground truth values. We then use a method called backpropagation to adjust the weights and biases based on the results of our loss function.

First, let us define some common loss functions. There are several types of loss functions used in neural networks, each with its own strengths and weaknesses. If we are trying to perform a regression task where the predicted output is a continuous scalar, we would use a mean squared error (MSE) loss. The MSE measures the average squared difference between the predicted value and the actual value (Equation 6). The main advantage of MSE is that it is smooth and easy to optimize using gradient descent. However, it can be sensitive to outliers since the squared term will magnify the error of a very poor prediction [69].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad \textbf{(6)}$$

Where n is the number of data points, Y_i is the true values, and $\hat{Y}_i$ is the predicted values. Another commonly used loss function is the binary cross entropy loss, which is used for binary classification problems. For these types of problems, we only have two possible outputs: yes or no (i.e. 0 or 1). The binary cross-entropy can be expressed as (Equation 7) [75].

$$H_p(q) = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log\left(p(Y_i)\right) + (1 - Y_i) \cdot \log\left(1 - p(Y_i)\right) \quad \textbf{(7)}$$

Where Y_i is the true label of the *i*-th sample, and *p* represents the predicted probability that the sample belongs to the positive class. The main advantage of binary cross-entropy is that it is a simple and efficient measure of the uncertainty of the model's predictions and can be easily optimized using gradient descent [76]. The main idea behind this loss function is to penalize the model more heavily when it makes a wrong prediction with high confidence (i.e., when the predicted probability is close to 0 or 1), and less heavily when it makes a wrong prediction with low confidence (i.e., when the predicted probability is close to 0.5) [77].

Categorical Cross-Entropy: When our classification task has more than 2 possible classes, we would use a categorical cross-entropy loss. It uses the same equation as the binary cross entropy, and simply calculates the cross entropy for each class label per observation.

Huber Loss: This is a loss function used for regression problems, similar to the MSE. The main difference is that it is less sensitive to outliers and can handle them better (Equation 8) [78]. It achieves this by combining MSE with mean absolute error. That is, it uses a different function for large errors (absolute value) and small errors (squared value).

$$L_\delta(Y_i - \hat{Y}_i) = \begin{cases} \frac{1}{2}(Y_i - \hat{Y}_i)^2 & for\ |Y_i - \hat{Y}_i| \le \delta \\ \delta|Y_i - \hat{Y}_i| - \frac{1}{2}\delta^2 & otherwise \end{cases} \quad \textbf{(8)}$$

Simply put, we use MSE when loss values are less than a parameter δ and we use MAE when it is greater than δ. Now that we understand a few ways to measure model predictions, we can adjust the weights and biases of the network using a process called backpropagation. Backpropagation was first described by Werbos is 1974 [79], it has since been improved for more complex systems [80]–[82]. At its core, backpropagation is an iterative process to calculate the derivatives of our loss function with respect to our weights and biases [83]. The backpropagation algorithm consists of two phases: the forward pass and the backward pass. During the forward

pass, the input is fed into the network, and the output is computed using the current weights. We then use the loss function to calculate the error between our predicted output and the true output. During the backward pass, the error is propagated backwards through the network, starting from the output layer and moving towards the input layer. The weights of each connection are adjusted in the opposite direction of the gradient of the error with respect to that weight. In other words, if a weight is causing the error to increase, the weight is decreased, and if the weight is causing the error to decrease, the weight is increased. This adjustment is performed using an optimization algorithm such as gradient descent. This operation terminates when the error is minimized.

Now, with a basic understanding of how neural networks are constructed and how they learn, we can begin to talk about how they can be tailored for different tasks such as classification, detection, segmentation, and prediction.

2.2 Machine Learning and Medical Imaging Applications

Classification, detection, and segmentation are all tasks commonly performed in machine learning and computer vision. Classification is the task of assigning an input to one of several predefined categories. For example, given an image of an animal, a classification algorithm would determine which animal it represents (Figure 10a). Detection is the task of identifying the presence and location of specific objects within an image. Object detection algorithms typically produce a bounding box around each detected object, along with a confidence score indicating the likelihood that the object is present (Figure 10b). Segmentation is the task of dividing an image into multiple segments, where each segment corresponds to a distinct object or region within the image. For example, in an image of multiple animals, segmentation might be used to identify and distinguish pixel-level labels for each animal (Figure 10c).

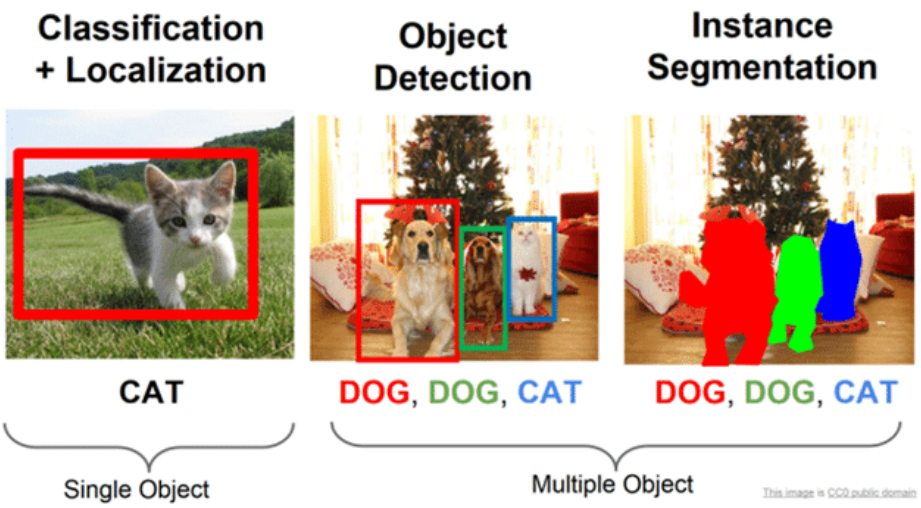

Figure 10. Computer vision tasks. Classification generates a label that best describes the image. Object detection produces labels of any found object within the image and its location. Instance segmentation gives a pixel-level label of objects found within the image. Image courtesy of [84] under CC-BY 4.0.

Using automated computer analysis for medical imaging applications has been around since the 1970s [85]. Early on, hardware limitations prevented complex tasks such as pathology detection and patient outcome prediction. However, researchers at the time were still able to successfully perform low level pixel segmentation [86]–[89], image enhancement [90]–[94], and basic classifiers [30]–[33]. Breakthroughs in both hardware and neural network architecture have led to NNs promising human levels of performance. Here, we will highlight some of the modern types of NNs and recent medical imaging applications.

2.2.1 Convolutional Neural Networks

One of the biggest breakthroughs in NNs is the use of convolution filters. Convolutional neural networks (CNNs) were first described by Fukushima and is designed to capture spatial patterns within images by using small convolutional filters [99]. The convolution layer is the first layer that is used to extract the various features from the input images. After this convolution layer, the data is then passed on to an activation and pooling layer, and then fully connected layers (Figure

11). The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification [100]. The goal of this convolution operation is to obtain all the high-level features of the image and at the same time reduce dimensionality. By reducing the dimensionality, we decrease the require computational power for processing the data and increase the rate of training [101].

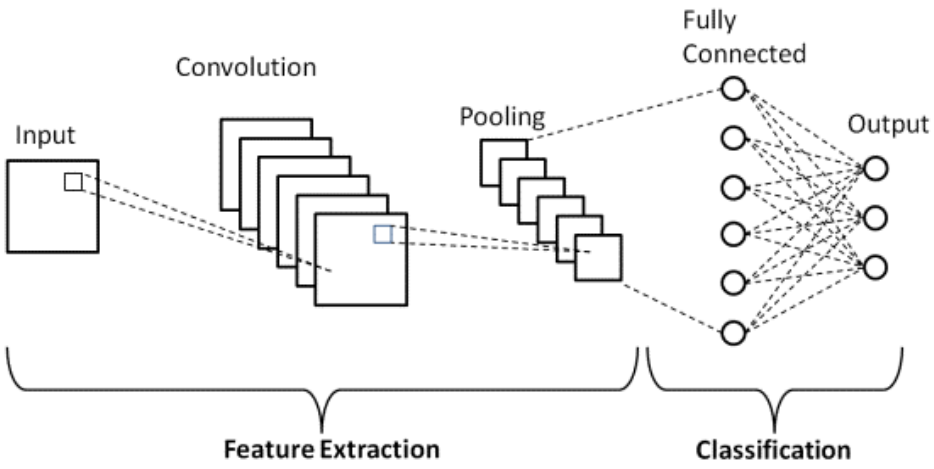

Figure 11. A simple convolutional neural network. Feature extraction of our input is done through convolution layers and pooling layers. The extracted features then are fully connected to our desired final output. Image courtesy of [102] under CC-BY-NC-ND 4.0.

To understand the convolution operation, first let us consider a 7x7 matrix and a 3x3 convolution kernel (Figure 12). An element-wise product between the kernel and the matrix is computed starting from the top right-hand corner of the matrix. The sum of the products for each cell is added to a new matrix called the feature map. The kernel is then shifted one cell the right and the process is repeated. The operation stops when the kernel reaches the bottom left of the matrix.

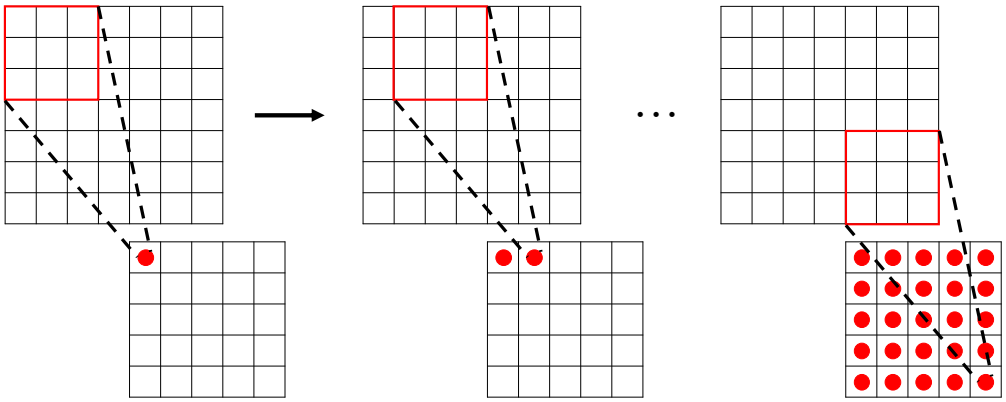

Figure 12. Visualization of the convolution operation. The 3x3 kernel (red box) will march along the matrix (7x7 top row), computing an element-wise product along the way. For each operation, the sum of the products will be added to a new matrix (5x5, bottom row). This newly generated matrix is called the feature map.

The convolution operation described above does not allow the center of each kernel to overlap the outermost element of the input tensor and reduces the height and width of the output feature map compared to the input tensor. Padding, typically zero padding, is a technique to address this issue, where rows and columns of zeros are added on each side of the input tensor, to fit the center of a kernel on the outermost element and keep the same in-plane dimension through the convolution operation (Figure 13). Modern CNN architectures usually employ zero padding to retain in-plane dimensions in order to apply more layers. Without zero padding, each successive feature map would get smaller after the convolution operation.

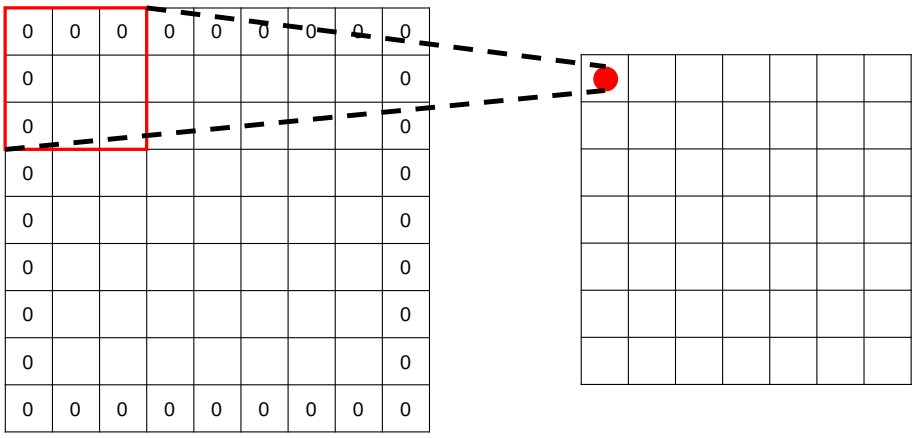

Figure 13. Visualization of zero padding. To retain the same 7x7 dimension of our original matrix, we must add a layer of zeros around it. The zero padded 9x9 matrix when convolved with a 3x3 kernel will yield a 7x7 matrix.

As mentioned previously, once the convolution process is complete, it is fed into an activation function and then a pooling layer. Generally, ReLU activation functions are used. A pooling layer is downsampling operation where we reduce our data, and subsequently decrease the number of subsequent learnable parameters [100]. This operation reduces the computational costs and therefore speeds up training. Pooling also allows for the extraction of features at different spatial scales. Two commonly used pooling operations are max pooling and average pooling (Figure 14). The operation creates "pools" of non-overlapping regions in the data and then represents each pool with a single number. For average pooling, we simply average all the data together in a pool. For max pooling, we take the maximum value of the pool. Max pooling is more commonly used as it captures the strongest activation in the feature map, which can help to retain information about the edges and other key features of an object [103].

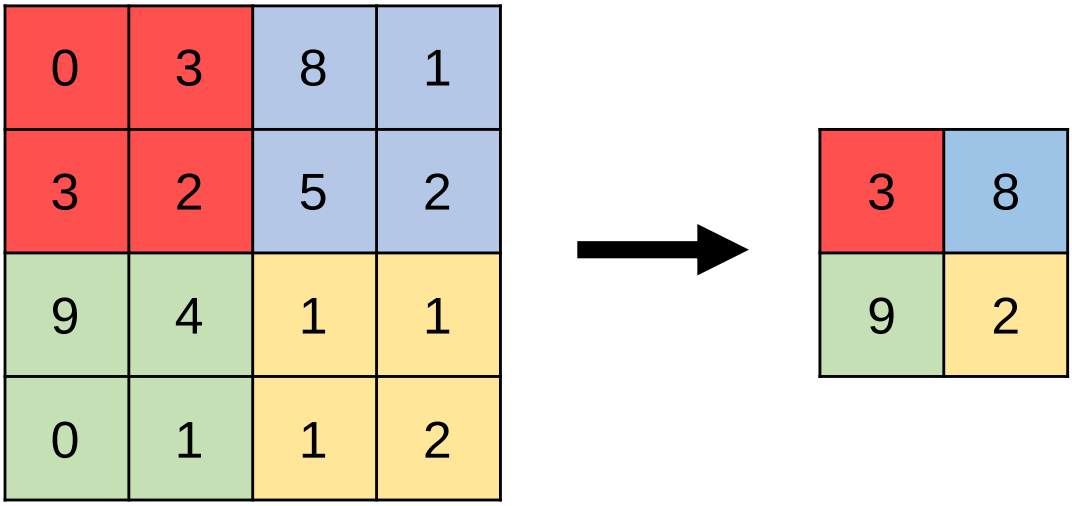

Figure 14. Visualization of max pooling. Here we use a 2x2 filter to run over out input 4x4 matrix. We'll use a stride of 2 (skipping every other index) so that regions won't overlap. For each region highlighted by the filter, we take the MAX value of the region and then map it to a new matrix.

Finally, the pooling layer is flattened (transformed into a 1-Dimensional vector) and input into a fully connected layer. A fully connected layer simply means that every element in the input vector is connected to every output node. The fully connected layer is eventually mapped to our final output layer. For classification tasks, it is normal to use one hot encoding, where each class is encoded by one output node. An output node simply returns a probability between 0-1 that the input data belongs to a particular class, therefore the final layer would return a vector of length equal to the number of possible classes.

2.2.2 U-Net

The U-net architecture was first proposed by Ronneberger et al. in 2015 and is a type of CNN [104]. The architecture consists of two phases, a downsampling phase and an upsampling phase, and was originally designed for image segmentation tasks (Figure 15). The downsampling phase is identical to a normal CNN as described in the previous section. The upsampling phase consists of up-convolutions with a large number of feature channels which allow the network to propagate context information to higher resolution layers [105]. These layers also have

concatenations with features from the contracting path. These concatenations are necessary to recover spatial information lost during downsampling [106]. The final layer uses a 1x1 convolution to map the component feature vector to the desired number of classes [104]. Many object detection networks such as EfficientNet, DenseNet, Yolo, and RetinaNet all use elements that are common to the U-net architecture.

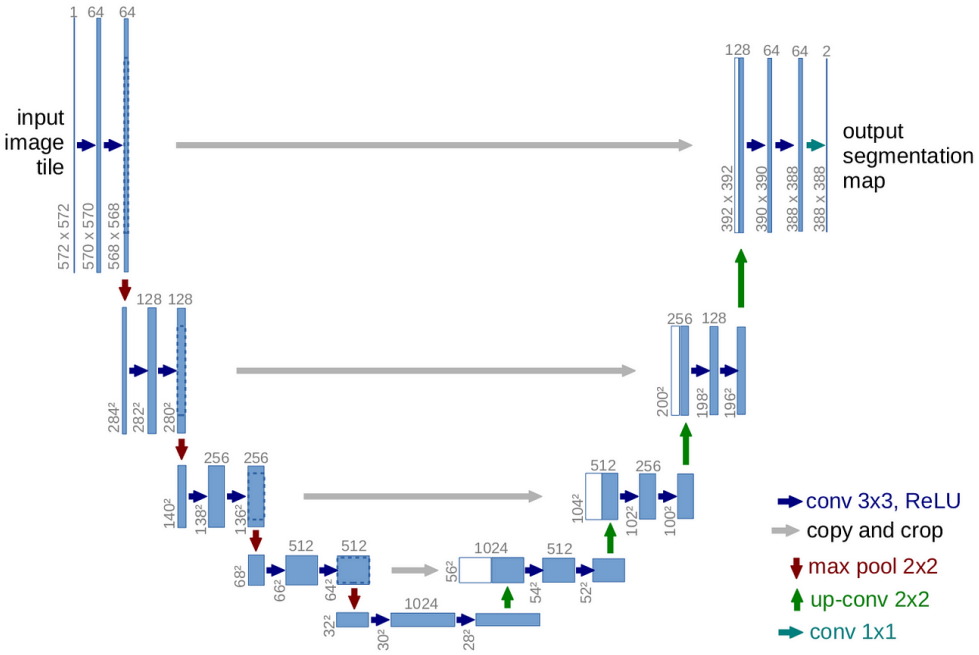

Figure 15. Original U-net architecture design from [104]. The input matrix is reduced through convolution and max pooling operations. Then, the extracted features are upsampled using transpose convolution operations. Image license CC-BY-NC-ND 4.0.

2.2.3 Residual Blocks

Residual blocks are a building block used in deep neural networks, particularly in the architecture known as ResNet (short for Residual Network). Residual blocks address the problem of vanishing gradients in very deep neural networks. As the network gets deeper, it becomes harder for the gradients to propagate backwards through the layers during training, leading to increasingly large errors. Residual blocks aim to address this problem by introducing a skip-connection, which

maps to activations earlier in the network (Figure 16). They are designed in such a way that the output of a layer is taken and added to another layer deeper in the block [107]. By learning these residual mappings rather than the underlying mappings, ResNet was able to significantly reduce the difficulty of training, which resulted in great performance boosts in terms of both training and generalization error [108].

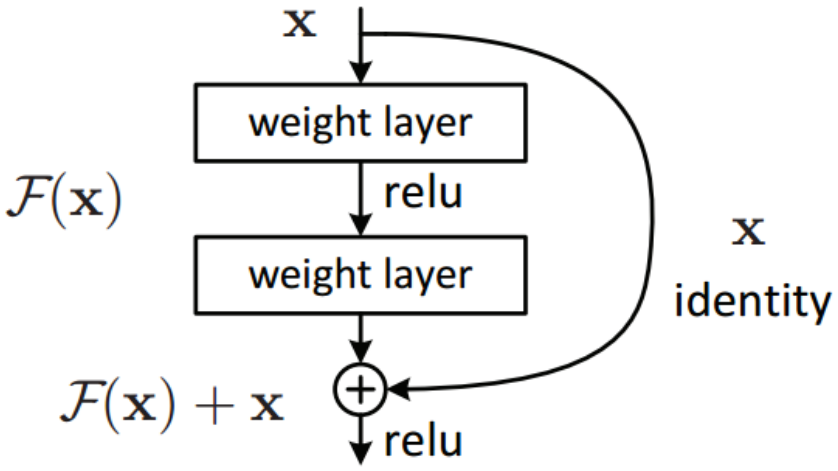

Figure 16. The original design of a single residual block from [107]. Here, the output from one layer directly feeds into the next layer *and* a layer 2 or 3 connections away. We can concatenate many of these blocks together to form deep neural networks. Image license under CC-BY-NC-ND 4.0.

2.2.4 Deep Learning for Rib Fracture Detection

Rib fractures are a common type of injury, present in approximately 4-12% of trauma admissions and is a major source of chronic pain [109], [110], making early detection and diagnosis critical. Therefore, with the rapid development of deep learning techniques in recent years, there has been a growing interest in the use of deep neural networks or deep learning (DNN and DL, respectively) for rib fracture detection. Several recent studies have shown that deep neural networks can outperform human radiologists in detecting rib fractures. Table 6 highlights articles published between 2020 and the present using state-of-the-art neural networks.

Table 6. Recently published studies localizing, detecting, segmenting, and classifying rib fractures.

Data and Model Description	Model Used (Type)	Results	Reference
3644 chest CT images were used to train a single shot deep neural network based off DenseNet. The performances of rib fracture detection by the network and two medical interns and two radiologists were compared.	DNN (Modified DenseNet)	The model outperformed the interns, achieving a sensitivity, positive predictive value, and F1-score of 0.645, 0.793, and 0.711, respectively, while the interns achieved mean scores of 0.285, 0.797, and 0.649. However, the model did not perform as well as the radiologists, who achieved a sensitivity of 0.860.	[111]
865 fractures on 713 ribs from 198 CT images were used to train a CNN object detector. Total fractures detected and total reading time by two radiologists (R1, R2) either assisted or unassisted by NN was recorded.	CNN (Faster R-CNN)	DL detected 687 (79.4%) of the 865 true fractures with 0.43 FPS. Sensitivity of radiologists assisted by DL significantly increased; 82.8% to 88.9% for R1, and 83.9% to 88.7% for R2.	[112]
8529 CT images containing 861 rib fractures were used to train a CNN. Precision, recall, F1-score, and diagnostic time of two junior radiologists with and without the deep learning model were computed.	CNN (VRB-Net)	CNN informed radiologists' precision, recall, and F1-score increased to 0.943, 0.978, and 0.960, respectively, from 0.812, 0.885, and 0.845.	[113]
1697 CT scans divided into 65:20:15 training, validation, and testing sets (594 fracture present cases) were used to train a 3D DNN. Their model was compared to ResNet, DenseNet, R(2+1)D, and CSN classifiers.	DNN (SGANet)	SGANet outperformed all other established networks in precision, sensitivity, and F1-score (68.97, 90.91, and 0.7843, respectively), and had the second-best specificity (78.05 vs CSN's 78.66)	[114]
1707 chest CTs split into 1507:100:100 training, validation, and testing were used to train a custom 3-step segmentation and detection model. First and second stage consisted of a U-net segmenting bone and then detecting ribs. The final stage used a 3D DenseNet to propose fracture location and classification. Radiologists were evaluated on precision, recall, F1-score, negative predictive value with and without the aid of the model.	DNN (Modified DenseNet)	Radiologists improved F1-score, precision, recall, and NPV with the use of the model (0.842 to 0.948, 0.773 to 0.946, 0.932 to 0.949, and 0.979 to 0.989, respectively). On average, the diagnosis time of radiologist assisted with this detection system was reduced by 65.3s. The model alone achieved F1-score, precision, recall, and NPV of 0.890, 0.869, 0.913 and 0.969, respectively.	[115]
511 whole body CT scans (fracture absent, n = 159, fracture present, n = 352) were used to train a 2-stage deep neural network. The first stage was a 3D ResNet model used to propose a region(s) that the Fast-Region CNN second stage would then filter out poor predictions.	DNN (Modified ResNet)	The model's sensitivity, specificity, positive predictive value, negative predictive value, accuracy, and F1-score was 87.4%, 91.5%, 82.3%, 94.1%, 90.2%, 0.85, respectively. Their model's sensitivity is approximately the same as others in literature, however their PPV is higher than average.	[116]

Table 6 (cont'd)

12208 emergency room (ER) trauma patients and an external dataset of 1613 ER trauma patients taking chest CT scans were used to train a cascaded deep neural network. The model consisted of two cascading U-net models that first segmented ribs and then detected fractures.	DNN (RB-Net)	In general, results showed that 6 attending radiologists tended to miss more rib fractures than the deep models; however, they generally reported fewer false positives on their external dataset, the model beat radiologists in patient-level diagnoses, with a sensitivity of 86.2% compared to 70.5%.	[117]
20646 annotated axial CT scans were used to transfer learn a variety of deep neural networks. The paper assessed the speed-accuracy trade-offs by using only the first -n blocks of each pretrained network.	DNN (InceptionV3, ResNet50, MobileNetV2, and VGG16)	Generally, the reduction of a single block reduced accuracy 1-1.5% but decreased the inference time by 10-25%. The best performance-to-speed model was the InceptionV3 network with 7 blocks, with an accuracy and sensitivity of 96.00% and 94.0%, respectively.	[118]
7473 annotated CT images from 900 patients were used to train a 3-step fracture segmentation model. The model is based on a 3D U-Net structure and consists of a preprocessing step, a sliding window prediction step, and a post-processing step. Sensitivity and false positives of the detection performance were compared between the model and expert radiologists.	3D U-Net (FracNet)	The model achieved a sensitivity of 92.9% with 5.27 false positives per scan where radiologists achieved a much lower false positives per scan (1.13), while underperforming the deep neural networks in terms of detection sensitivities (77.5%).	[119]
10943 CT scans were used to train an ensemble 3D U-Net + 2D RCNN network. The U-Net was used to segment the ribcage while the RCNN was used to detect fractures. Precision, sensitivity, and F1 score were used as metrics to assess model vs radiologist rib fracture detection performance.	Ensemble 3D U-Net and 2D Fast RCNN	The model achieved a precision of 82.2% and sensitivity of 84.9%; compared to three radiologists with a precision of 90.6% and sensitivity of 79.7%. With the help of the model, radiologists achieved a higher sensitivity (89.2%) but a lower precision (88.4%).	[120]
The MICCAI 2020 RibFrac challenge consists of 660 CT scans with ~5000 fractures split into 420 training, 80 validation, and 160 testing set. This dataset was used to train a two-stage detector with a nnU-Net segmentation network and a DenseNet classification network. This nnU-net model was compared to two other nnU-net versions and assessed based on Dice coefficient, intersection over union (IOU), and average symmetric surface distance (ASSD).	3D nnU-Net and DenseNet	A higher score is better for both Dice and IOU, and a lower score is better for ASSD. The nnU-net model achieved a Dice, IOU, and ASSD score of 62.80, 48.81, and 11.40, respectively. The model outperformed its 2D and 3D cascaded versions in all three metrics.	[121]

Table 6 (cont'd)

1080 radiographs were randomly divided into the training set (918 radiographs) and the testing set (162 radiographs) and used to train an off-the-shelf object detector (YOLOv3). Receiver operating characteristic (ROC) and free-response ROC (FROC) were used to evaluate the model's diagnostic performance against radiologists.	CNN (YOLOv3)	The sensitivity and precision of the detection by the CNN model, senior radiologist, and junior radiologist were 87.3%, 80.3%, and 80.3%, respectively, and 82.4%, 73.4%, and 81.7%, respectively. The sensitivity of detection was significantly higher in the CNN model than among the junior radiologist ($P = 0.01$), however no significant difference existed between the CNN and senior radiologist ($P > 0.05$)	[122]
4366 chest X-rays (3411 fracture absent and 955 fracture present) were used to train a two-stage object detector. A U-net model first took the image and segmented the left and right lungs. Then an EfficientNet model was used to classify the ROIs into fracture or no fracture. The model was evaluated using area under the receiver operator characteristic curve (AUROC) and accuracy.	U-net and EfficientNet	The model achieved an AUROC of 0.965 and an accuracy of 0.916. This article did not compare the model to other networks or radiologists.	[123]
1020 CT images and patient clinical information was used to train two models: Faster RCNN and a fusion ResNet101+RCNN. The ResNet101 network was used as a feature extraction step, where then clinical information was concatenated to the output, and then used as input to the RCNN classifier. The diagnostic performance of both models and radiologists were assessed based on precision, recall (sensitivity), and F1-score.	ResNet101 and Faster RCNN	The fusion model outperformed the regular Faster RCNN model in all metrics; precision 0.799 to 0.629, recall 0.973 to 0.945, and F1-score 0.877 to 0.755. The fusion model also had significantly higher sensitivity (0.95 vs 0.77) but significantly lower precision (0.80 vs 0.87) compared to radiologists.	[124]

Overall, these papers show promising results for machine learned rib fracture detection and suggest that it could be used to inform medical decision making. However, there are also several challenges that need to be addressed before deep neural networks can be widely used in clinical practice. One major challenge is biases in the data. Most of the above studies are heavily biased for white adult males. This means that potentially, the performance of the object detectors would be poor when presented with an image of a child, woman, or person of color. Additionally, only 4 of the above papers are multicenter, which means most object detectors may have limited generalizability. Another challenge is the interpretability of deep neural networks. We often describe NNs as "black boxes" since we do not know *why* the network is making its decision. This

is a problem because physicians need to be able to provide an explanation for their diagnosis and treatment plan. The black box issue is also why the first step in integrating NNs with clinical decision making is to have them inform or aid doctors, not replace them.

The first issue of biased data and challenges with poor generalizability could potentially be partially solved using data augmentation. In general, when the diversity of the training set is higher, the performance of the model also improves. With limited availability of such expertly labeled medical images, researchers use data augmentation to generate synthetic medical images, either by simple morphological operations or through more complex techniques. We will talk more about data augmentation in the following section.

2.3 Data Augmentation Techniques

For a ML model to learn effectively, the observed data must be a diverse, accurate representation of the true distribution. Therefore, to properly estimate the true distribution, extremely large datasets become necessary [125]. However, in healthcare, datasets of sufficient size may be rare or absent, thus hindering direct training of ML models. Large amounts of medical imaging data are hard to acquire, as lack of standardization, lengthy curation process, releasing HIPPA compliant images, and need for expert labeling hinder the availability of training data [126]. Additionally, medical imaging data acquisition can be affected by the prevalence of the disease in question as well as the cost of the imaging modality. One of the ways of dealing with this problem is data augmentation, where we supplement our datasets with slightly modified copies of already existing data or newly created synthetic data based on existing data. Early methods of data augmentation included simple morphological operations such as shrinking, rotations, blurs, flips, and noise addition [127]. Recently, sophisticated data augmentation methods are based on a class of neural networks called Generative Adversarial Networks (GANs), which generate new images of high perceptual quality that combine the content of a base image with the appearance of

another one [128]. GANs have also been used widely for image-to-image translation, where we transform an image from one domain to another (i.e. image of a horse to image of a zebra).

GANs, were a huge breakthrough for data augmentation. It utilized two neural networks which were trained simultaneously to output realistic fake images by learning the probability distribution of a set of images (Figure 17) [129]. More specifically, a generator network creates fake images, and a discriminator is fed real and fake images and determines which is real. As both networks learn and improve, we reach a stop condition where hopefully our generator outputs fake images indistinguishable from the real ones by the discriminator. GANs have been adapted for image-to-image translation tasks, where a neural network learns to create a mapping from images of one domain to another domain. This technique has been used to transform PET to CT image [130], create lesions on non-lesioned dermatological samples [131], create tumors on healthy brain MR images [132], and transform T1 to T2-weighted MR images [133]. Here, we will review GANs and its many variants, common evaluation metrics, and its medical imaging applications.

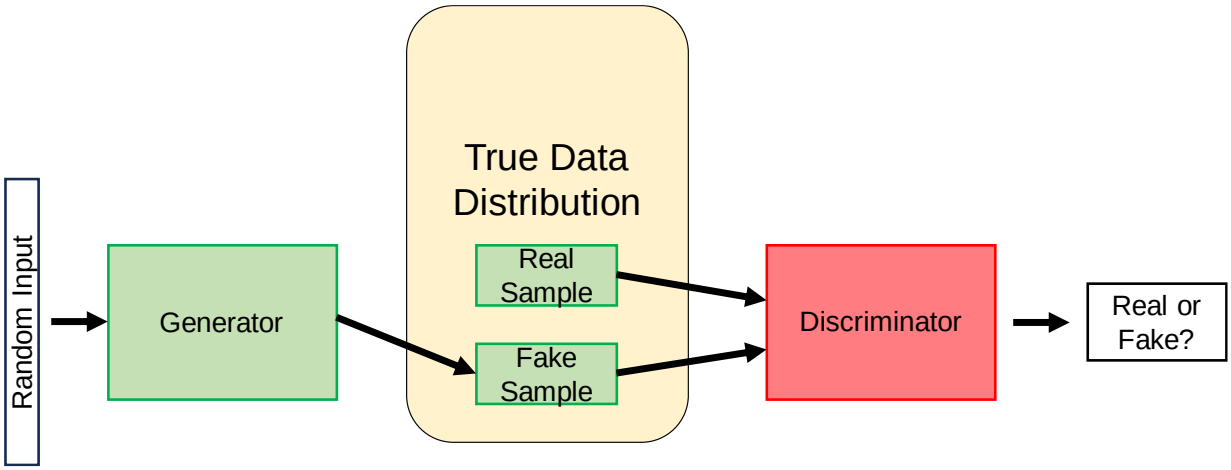

Figure 17. Generative Adversarial Network architecture. We have two neural networks "competing" against each other. The generator's goal is to fool the discriminator by outputting convincing fake samples. The discriminator's goal is to tell whether it's being given a real or a fake sample.

2.3.1 Generative Adversarial Nets

The original GAN as described by Goodfellow et al. is non-conditional, meaning that it takes a random latent vector and maps it to the sample distribution [129]. The GAN simply needs to generate images like those in the dataset given a random vector. The generator architecture is designed similarly to the upsampling phase of the U-net architecture. It goes through a project and reshape operation, and then consecutive up-convolution layers until we reach the desired resolution (Figure 18). The discriminator is a normal CNN classifier and outputs a probability between 0-1.

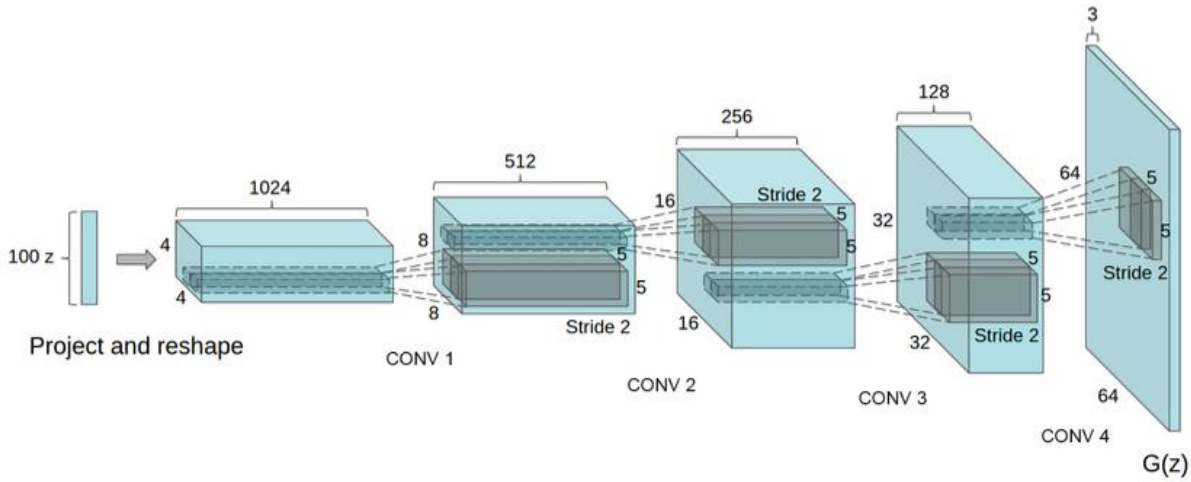

Figure 18. Generator architecture of a of a non-conditional GAN. A random latent vector is upsampled through multiple transpose convolution layers until we get to the desired resolution. Image courtesy of [134] under CC-BY-NC-ND 4.0.

Let us define a generator G that is trying to learn a distribution p_g from data *x* and produces a mapping $G(z;\ \theta_g)$ from a vector $p_z(z)$. Here, θ_g are learnable parameters. Now, let us define a discriminator $D(x;\ \theta_d)$ is trained to maximize the probability of assigning the correct label to both training examples and samples from the generator. Here, $D(x)$ is the probability that *x* is from our training data and not from $G(z)$. Simultaneously, the generator is trained to minimize:

$$\log\left(1 - D\left(G(z)\right)\right) \quad \textbf{(9)}$$

And both networks play the following two-player minimax game with the loss function:

$$\min_G \min_D V(\mathrm{D}, \mathrm{G}) = \mathbb{E}_{\mathrm{x}\sim \mathrm{p}_{data}(x)}[\log \mathrm{D}(x)] + \mathbb{E}_{\mathrm{z}\sim \mathrm{p}_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad \textbf{(10)}$$

Once trained sufficiently and assuming Equation 5 converges, it will reach a point where neither D or G can improve because $p_g = p_{data}$ The discriminator is unable to differentiate between the two distributions, i.e. $D(x) = 1/2$.

There are many challenges when training the original GAN, such as mode collapse, vanishing gradients, and non-convergence. Mode collapse is when the generator can only create a small set of convincing outputs. These outputs, while realistic, represent only a portion of the sample distribution. Therefore, it easily fools the discriminator and hinders learning. For example, Figure 19, bottom row, shows mode collapse when training on the MNIST digits dataset. This dataset contains handwritten digits from 0-9 and ideally the GAN must learn to produce each class. However, over time, it learns to only produce a 6, which fools the discriminator every time.

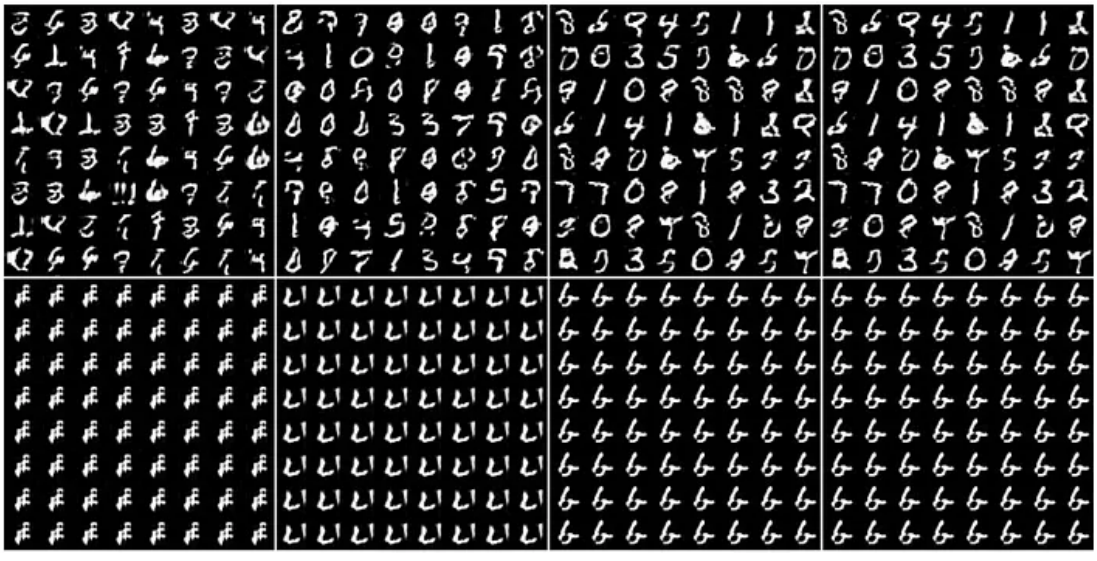

Figure 19. Example of successful GAN training (top row) and mode collapse (bottom row). Image courtesy of [135] under CC-BY-NC-ND 4.0.

The issue of non-convergence arises due to the non-convex nature of the loss function. When the loss function is non-convex, it is more difficult to minimize using gradient descent. Intuitively, D or G always counters the actions of the other in the next iteration, making large swings in the learning curve (Figure 20). Arjovsky and Bottou dives deeper into the nature of this phenomenon, showing how the norm of the gradient grows drastically as the discriminator trains longer [136]. In all cases, using this to update the generator leads to a notorious decrease in sample quality. Additionally, the large swings in the learning curve show that the variance of the gradients is increasing, which is known to delve into slower convergence and more unstable behavior in the optimization [137].

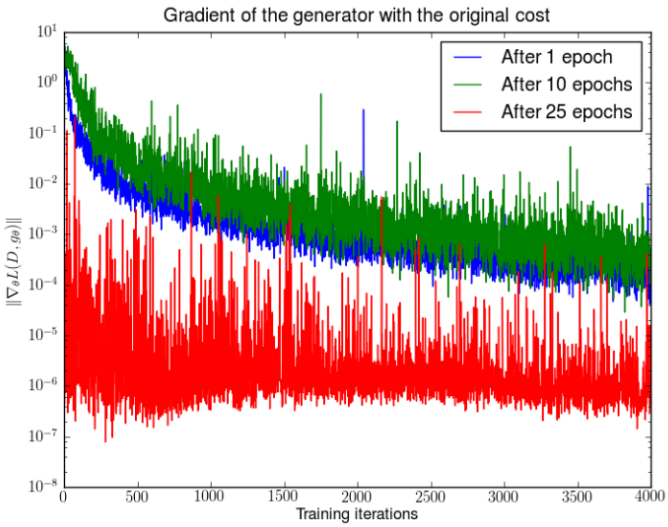

Figure 20. Example of unstable generator training. In this study, the generator network is fixed while only the discriminator trains. The gradient norms quickly decay with wild swings from iteration to iteration. This demonstrates that as the discriminator improves, the generator's gradient vanishes. Image courtesy of [136] under CC-BY-NC-ND 4.0.

2.3.2 GAN Variants

To solve these issues, multiple papers have suggested alternative loss functions [138]–[145]. Among these, the most popular and robust is Wasserstein distance GAN (WGAN) and WGAN with gradient penalty (WGAN-GP). The Wasserstein distance alone is informally defined

as the minimum cost of transporting mass in order to transform the distribution q into the distribution p (where the cost is mass times transport distance) [145] and is represented by the following equation:

$$min_G max_{D\in \mathrm{D}}\ \mathbb{E}_{\mathrm{x}\sim \mathrm{p}_r}[\mathrm{D}(x)] - \mathbb{E}_{\tilde{x}\sim \mathrm{p}_g}[D(\tilde{x})] \quad \textbf{(11)}$$

Intuitively, we can explain WGAN as simply minimizing the distance between the distribution of the generator's output and the true distribution (modeled by your training data). WGAN uses a technique called weight clipping, which enforces a 1-Lipschitz constraint on the discriminator. A Lipschitz constraint limits how fast a function changes by putting bounds on the function's first derivative. This means that the weights of the discriminator are forced to lie within a compact space defined by a Lipschitz function. For example, a sin function, the absolute value of its derivative is always bounded by 1 and therefore it is 1-Lipschitz constrained. Intuitively, Lipschitz continuity bounds the gradients and is beneficial in mitigating gradient explosions in deep learning. Instead of weight clipping WGAN-GP enforces the 1-Lipschitz constraint by adding an additional loss term to the Wasserstein distance (Equation 12).

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}. \quad \textbf{(12)}$$

The penalty term is based on the norm of the gradient of the discriminator's output with respect to the input data. Specifically, the penalty term is defined as the difference between the norm of the gradient and a constant value of 1, squared and multiplied by a hyperparameter λ. Intuitively, the gradient penalty satisfies the Lipschitz constraint by encouraging the discriminator to have a gradient with a norm close to 1, and penalizes it when the norm deviates from 1. While demonstrably better than the alternatives, it does not guarantee convergence [146].

In addition to loss function changes, there were numerous architectural improvements made. We highlight the most popular types in Table 7:

Table 7. Common GAN variants.

Variant	Novelty/Description	Reference
cGAN	Adds conditional information, specifically class label information to the basic GAN architecture. This allows the generator to selectively generate an imbalanced class through label input.	[147]
DA-GAN	Introduces a "Deep Attention" mechanism that is a compound loss of instance-level and set-level translation task. Essentially, for each pair of images, not only is the goal to egions of each image to each	[148]
Table 7 (cont'd)		
DCGAN	Applied specific architectural constraints to allow for stable training in a deep network. These constraints include using a batchnorm layer in both the generator and the discriminator, using ReLU activation in generator for all layers except for the output, which uses Tanh, and using LeakyReLU activation in the discriminator for all layers.	[134]
PGGAN	The key idea is to grow both the generator and discriminator progressively: starting from a low resolution, new layers are added so that the model produces increasingly fine details as training progresses. This both speeds the training up and greatly stabilizes it, allowing us to produce images of unprecedented quality.	[149]
StyleGAN	Instead of starting from a random vector input, the generator starts from a learned input. StyleGAN's generator, therefore, consists of two separate networks; one for mapping the latent space input to an intermediate space and one for synthesis. This mapped input is now used in the synthesis network and only added to specific layers that correspond to a "style", i.e. glasses, male, facing left, etc.	[150]
AC-GAN	Adds an auxiliary class decoder network to discriminator to reconstruct class labels. By forcing a model to perform additional tasks is known to improve performance on the original task. This was the first model to measure discriminability using the Inception network, which is now standard in assessing the quality of synthetic images for GANs.	[151]
Pix2pix	Pix2pix introduced the embedding of whole images as an input to the generator instead of random noise allowing a paired, image-to-image translation.	[152]
StarGAN	StarGAN is a novel and scalable approach to perform image-to-image translation among multiple domains using a single model. It has a unified modeling architecture that allows simultaneous training of multiple datasets and different domains within a single network.	[153]
cycleGAN	CycleGANs introduced cycle consistency and identity losses allowing for unpaired training of pix2pix architecture.	[154]

2.3.3 Image-to-Image Translation GANs

Image-to-Image translation is the transformation of one image domain to another while preserving content representations [155]. For example, we can convert a natural black and white image of Marilyn Monroe to a green one (Figure 21). Notice how the intrinsic source content (background, location of features, etc) are preserved while the extrinsic target style is transferred (grayscale to green). We can use a type of conditional GAN, namely pix2pix, UNIT, StarGAN,

and cycleGAN for this task. Unlike the original GAN which is unsupervised learning (trying to discover the underlying distribution), conditional GANs are a type of supervised learning (we define the expected outcome).

Figure 21. Unpaired and Paired Images. Unpaired images do not require the structural constraints that paired images have. In other words, the position of the apples do not have to match the position of the apples. However, with paired images, we can speed up training by applying this constraint.

The chief innovation for pix2pix is the use of paired images, using a U-net architecture for the generator, and a patchGAN architecture for the discriminator [152]. Paired images (Figure 21) consist of two sets of spatially identical but texturally different images. They are paired in the sense that an image from set A only maps to the corresponding image in set B and are used together as inputs to the pix2pix discriminator. The use of paired images has shown to be more efficient in training and is more likely to converge as it makes the translation task more constrained [156]. The use of a U-net shaped generator with skip connections allows for better preservation of underlying structures. GANs are known to produce blurry images, since the discriminator tends to model low-frequency content better [157]. To model high frequencies, pix2pix uses a discriminator that classifies localized image patches. It has been shown that by querying random samples from images models are better able to preserve local structure [158]. The average probability of all patches is then used as the final output of the discriminator. While results produce sharp, high

resolution, and generally realistic images, the need for a paired dataset limits pix2pix to very specific image translation applications.

CycleGAN was developed specifically to use unpaired training data and utilizes two generators and two discriminators in its network with a specialized cyclic loss [154]. One generator-discriminator pair aims to translate images from domain A to domain B while the other performs the opposite operation and translates images from B to A. The generator network and discriminator network are nearly identical to pix2pix, other than having two sets (Figure 22).

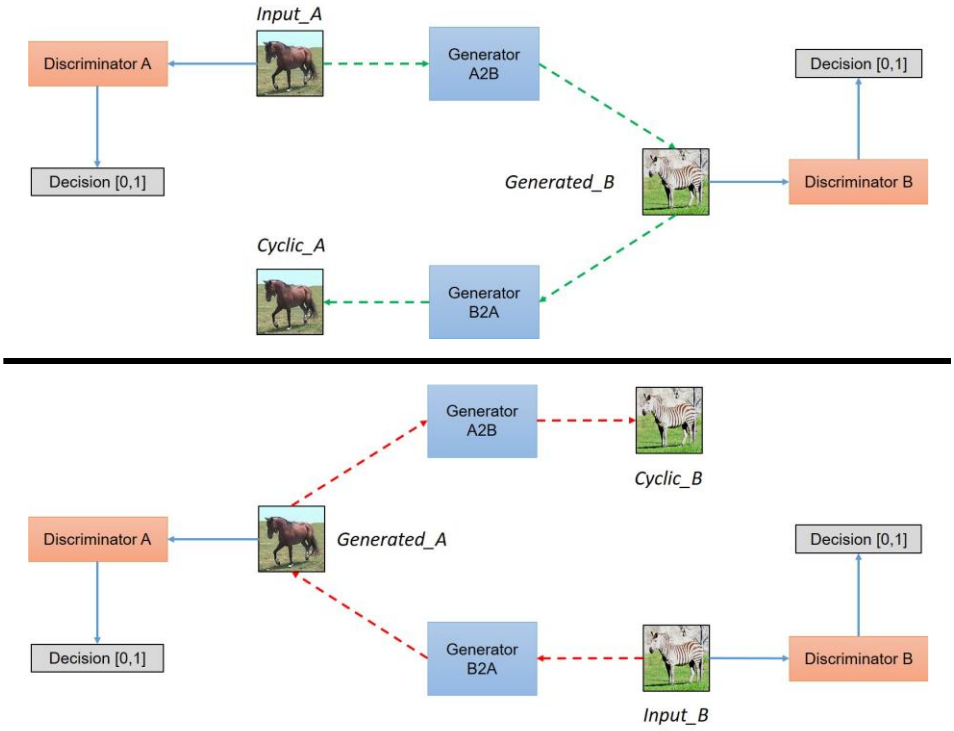

Figure 22. CycleGAN architecture. Two pairs of generators and discriminators are trained simultaneously. One pair translates from domain A to domain B, and the other from domain B to domain A. A cyclic loss is computed to ensure that an image, when translated through both generators, remains the same. Image courtesy of [159] under BSD license.

Images from each domain are fed into a generator to transform them to the opposing domain (G_A : X→Y and G_B : Y→X). Then, a real image and the transformed image are given to the respective discriminator to judge which is real and which is fake. D_A is therefore trained given samples $\{x_i\}_{i=1}^{n}, x \in X$, with distributions $x \sim p_X(x)$ and D_B is trained given the samples

$\{y_j\}_{j=1}^{n}, y \in Y$, with distribution $y \sim p_Y(y)$, where X and Y are the domains of the unpaired images. Discriminator B's loss function is therefore defined as

$$L_{GAN}(G_A, D_B, X, Y) = \mathbb{E}_{y \sim p_Y(y)}[\log D_B(y)] + \mathbb{E}_{x \sim p_X(x)}\left[\log(1 - D_B(G_A(x))\right] \quad \textbf{(13)}$$

and discriminator A's loss function is

$$L_{GAN}(G_B, D_A, Y, X) = \mathbb{E}_{y \sim p_Y(y)}\left[\log(1 - D_A(G_B(x))\right] + \mathbb{E}_{x \sim p_X(x)}[\log D_A(y)] \quad \textbf{(14)}$$

The use of the cyclic loss function is inspired by an intuitive idea in natural languages. When translating from one language to another (English → Spanish: Hello to Hola), we should expect the original input when translating back (Spanish → English: Hola to Hello). Therefore, we can add an additional loss term, originally called the cycle consistency loss, to the discriminator. To ensure cycle consistency, the transformed image is fed to the opposing generator, and then compared to the original input image. In other words, $G_A(G_B(X)) = X$ and $G_B(G_A(Y)) = Y$. Therefore, the cycle consistency loss function is given by

$$L_{Cyc}(G_A, G_B) = \mathbb{E}_{x \sim p_X(x)}\left[\left\|F(G(x)) - x\right\|_1\right] + \mathbb{E}_{y \sim p_Y(y)}\left[\left\|G(F(y)) - y\right\|_1\right] \quad \textbf{(15)}$$

Combined, the full loss function is

$$L_{tot}(G_A, G_B, D_A, D_B) = L_{GAN}(G_A, D_B, X, Y) + L_{GAN}(G_B, D_A, Y, X) + \lambda L_{Cyc}(G, F) \quad \textbf{(16)}$$

where λ controls the relative importance of the cycle consistency loss.

CycleGAN shows similar performance as pix2pix, but again uses unpaired images. This allows for easier data curation, and therefore can be applied to more image translation tasks. This is particularly useful for medical image translation, where data is notoriously difficult to acquire in high quantities. However, unpaired images makes the translation more unconstrained, and therefore is less likely to converge and is susceptible to exploding gradients [156].

2.3.4 Common quantitative evaluation metrics

There are 4 common ways to evaluate GANs: (1) MAE/MSE, (2) Human Observer Study, (3) Inception Score/Frechet Inception Distance, and (4) Downstream Task Performance [160], [161]. If we have ground truth images, such as in pix2pix or cGAN, we can measure the mean squared error between the generator prediction and those images. We can also use peak signal to noise ratio (PSNR), Dice similarity coefficient, Jaccard similarity index (JI), and structural similarity index measure (SSIM) to evaluate image similarity. These metrics are often used in segmentation GANs. Ideally, we would have expert readers assess the quality, realism, and diversity of the synthetic images, but this is generally the costliest option. The inception score (IS) and Frechet Inception Distance (FID) both use the pretrained InceptionV3 network [162] to measure average conditional probability of our generated samples [163]. The main difference is that IS measures the difference between the predicted class probabilities of the generated images and the training set and FID measures the distance between the distributions of the last layer of the InceptionV3 network for synthetic and real images [164], [165]. The FID is sensitive to class mode dropping, with distances increasing with greater class discrepancies between the two distributions as the average set of features begins to differ [166]. Both IS and FID have been empirically shown to have high correlation with the quality and diversity of generated images and is consistent with human judgments [167]–[169]. Finally, we can indirectly measure the quality of the generated images by using them for a downstream task such as object detection. If the generator is able to produce realistic and diverse results, then the use of synthetic images in the training set should improve the performance of the object detector. Using downstream metrics is the most practical evaluation of the quality of synthetic images since it is the goal of data augmentation.

2.4 Review of GANs applied to Medical Imaging Applications

This section is organized by GAN task and then image modality. Due to the sheer number of articles published in this field, we have limited the scope of this review to papers published after 2019.

2.4.1 Image Reconstruction

Over the years, reconstruction methods for medical images have changed from analytical to iterative to machine learned. There are many factors that affect the quality of the reconstructed image for various modalities, from concentration of contrast administered, amount of radiation administered, level of noise and artifacts, resolution, and sampling [170]. Many articles use GANs described in Table 8, or close variations. For example, Waheed et al present a method that architecturally is identical to an ACGAN, the only changes made is the size of the input layer of the generator [171]. Therefore, it would be classified in the ACGAN family. However, while Kamran et al proposed a model whose architecture sounds similar to cycleGAN; two discriminators and two generators, is would not be considered in the cycleGAN family because each pair of networks are not performing the opposite translation task [172]. If the GAN is not closely related to any of the ones mentioned in Table 8, then the name used by the authors is simply given. Here, we highlight in Table 8 GANs that mitigate these factors and are effective in compromised reconstruction. We focus on the way GANs are tailored for the reconstruction task and the novelty of the application.

Table 8. Image reconstruction applications using GANs. Generally, authors choose to either reconstruct an image from raw data or from a low-sample rate/noisy image. For reconstruction tasks, PSNR, SSIM, and MSE are commonly used to evaluate the image quality, but is left out of this table for simplicity.

Modality	Reference	GAN Variant or Family	Data Description and Network Architecture Novelty
MRI	[173]	cGAN	A conditional GAN was trained to reconstruct fully sampled, multi-contrast MRI using undersampled acquisitions. This deviates from a traditional cGAN where the input is a latent vector with a class label. The proposed approach can successfully recover pathologies that are either missing in the source contrast or are not clearly visible in the undersampled acquisitions of the target contrast. Additionally, this reconstruction method was 50x faster than traditional reconstruction techniques. This network was trained and tested using publicly available datasets: MIDAS, IXI, and BRATS and found to have reasonable peak signal to noise ratio (PSNR), structural similarity index (SSIM), and root mean squared error (RMSE) values.
	[174]	cycleGAN	Due to lack of fully sampled, high spatio-temporal resolution ground truth images for time-resolved MR angiography (tMRA), a cycleGAN was trained on sparsely sampled, aliased images and unpaired high-resolution MRI images. Unlike the traditional cycleGAN, which uses two pairs of generator-discriminators, the author's "Optimal Transport-cycleGAN" only uses one pair and replaces one generator with a deterministic Fourier transform. Twelve sets of in vivo 3D DCE MRI data was acquired, which corresponds to 78,740 slices of data, 33,890 which were used for training. They demonstrated that undersampled images with reduced view sharing can be properly reconstructed using a cycleGAN.
	[175]	DCGAN	Knee MR scans were obtained from 19 patients. Each volume consists of 320 2D slices that were divided into training, validation, and test examples with a 70/15/15 split. This data was used to train multiple variable autoencoders (generator) of varying sizes (1-4 residual blocks). They demonstrated that multiple recurrent blocks decrease uncertainty, which suggests an effective way of promoting robustness.
	[176]	DA-GAN	The original DA-GAN was improved by replacing a generator loss term with the Wasserstein distance and adding a perceptual loss term. The authors used the MICCAI 2013 grand challenge diencephalon dataset using 3 different under sampling masks. They demonstrated that their version of the DA-GAN was superior at reconstruction when compared to other GANs such as Pixel-GAN and regular DA-GAN, and a non-GAN neural network called ADMM-Net.
	[177]	cycleGAN	The general architecture presented in [174] remains the same with minor adjustments. Namely, the authors used a Wasserstein GAN loss in the generator. They validated their network with two publicly available datasets: Human Connectome Project and the FastMRI dataset. The authors show with both datasets that the addition of the Wasserstein loss function improves the PSNR and SSIM of the reconstruction.

Table 8 (cont'd)

	[178]	DCGAN	A novel two-generator GAN was proposed for sparsely sampled reconstruction. The first generator maps the sparse data to a fully sampled k-space. The second generator maps the IFT of the fully sampled k-space to a denoised and anti-aliased image. The proposed method was applied to three essentially different brain MRI datasets: IXI, the 2015 Longitudinal MS Lesion Segmentation Challenge, and a private DCE-MRI dataset of stroke and brain-tumor patients acquired at the Soroka University Medical Center. This method of reconstruction was compared to conventional Compressed Sensing MRI (CS-MRI) and a different deep-learning approach called ADMM-Net and was shown to be superior to both.
CT	[179]	RED-GAN [180]	A GAN to denoise low dose CTs (LDCT) was developed. The novelty comes with the proposed "Noise Aware Loss" in the discriminator network. A patch-wise mean squared error (MSE) loss was used to mitigate gradient vanishing. Since CT images contains air in a significant region; after a few epochs of training, the MSE will be very small for all those image regions; subsequently, gradient update during backpropagation will be insignificant, as the total loss is small. So, after a few epochs, effective training will stop implicitly. The generator network was borrowed from [180].They tested this network on the publicly available 2016 NIH-AAPM-Mayo Clinic LDCT dataset. It was shown to outperform other state of the art NN reconstruction methods in terms of PSNR and SSIM.
	[181]	RED-GAN	Unlike standard GAN discriminators, the proposed DU-GAN (dual U-net GAN) utilizes a U-Net-based discriminator for LDCT denoising, therefore providing per-pixel feedback and the global structural difference to the denoising model. Additionally, there are two discriminators, one for both the image and gradient domain. The RED-GAN generator was also used. The network was trained and tested using 2016 NIHAAPM-Mayo Clinic LDCT dataset.
	[182]	SA-GAN	The proposed self-attention GAN (SA-GAN) generator is composed of multiple layers of self-attention blocks sandwiched in between 3D conv layers. Each block computes the correlation matrix that represents spatial dependencies between any two positions within the input feature maps. Each position is calculated and updated by the weighted sum of all other positions. This is done both depth wise (between slices) and plane wise (within slices). The network was trained and tested using the 2016 NIHAAPM-Mayo Clinic LDCT dataset.
	[183]	WGAN	The generator is U-net structure and has 9 layers total, with 2 downsampling blocks, 5 residual blocks, and 2 upsampling blocks. The generator loss has three parts: the WGAN loss, an SSIM loss, and a L2 loss (MSE). The discriminator consists of six convolutional layers and three fully-connected layers. The network was trained and tested using the 2016 NIHAAPM-Mayo Clinic LDCT dataset.
	[184]	cycleGAN, IdentityGAN, and GAN-CIRCLE	The authors compared three previously described GANs for the use of unpaired translation of LDCT to FDCT. Among CycleGAN, IdentityGAN, and GAN-CIRCLE, the latter achieves the best denoising performance (Lowest PSNR and SSIM) with the shortest computation time. Subsequently, GAN-CIRCLE is used to demonstrate that the increasing number of training patches and of training patients can improve denoising performance. The network was trained and tested using the 2016 NIHAAPM-Mayo Clinic LDCT dataset.

Table 8 (cont'd)

	[185]	WGAN	The generator and discriminator networks are similar to previously described WGANs. The generator is U-net shaped and the discriminator is borrowed from [186]. The authors proposed an additional perceptual loss in addition to the WGAN and L2 losses and is simply the feature vector given by VGG-19. The performance of the proposed DR-WGAN is compared to other previously described GANs and was better in terms of PSNR and SSIM. All GANs were trained and tested using the publicly available LUNA16 dataset.
	[187]	cGAN	Unlike other deep residual generators with a perception loss, the proposed network DRL is a conditional GAN. The input is a LDCT image and uses a Sobel filtered LDCT image as the class label. The authors show that the edge detection layer improves the performance of the network, and that the overall network outperforms other reconstruction techniques. The GAN was trained and tested on a simulated dataset from The Cancer Imaging Archive [188], as well as curated deceased piglet and thoracic CT datasets.
PET	[189]	LSGAN [139]	The proposed model is an improvement of the LSGAN, which uses a least squares loss function for the discriminator. The novelty comes with the addition of a self-attention layer in the generator, as well as more residual blocks to better preserve structural details and edges. Whole body scans were used, where the ground truth images were produced after 150s scanning as noise-free HC data, and the input images were produced after scanning for 75 s as low-count input data with noise. In terms of PSNR, the proposed model is only slightly better than other neural networks (CNN3D) and traditional noise reduction reconstruction algorithms such as non-local means (NLM). However, it significantly outperforms all other techniques when measured by SSIM.
	[190]	Transformer-GAN	The proposed generator network comprises three components: (1) a CNN-based encoder, (2) a transformer network used to model the long-range dependencies between the input sequences learned, and (3) a CNN-based decoder. Residual blocks that are normally found in a DCGAN are replaced by a transformer, which is useful in capturing slice-to-slice information. This was tested on a clinical dataset which includes eight normal control (NC) subjects and eight mild cognitive impairment (MCI) subjects, from which 729 large patches of size $64 \times 64 \times 64$ are sampled.
	[191]	DCGAN	The proposed network is described as a 2.5D encoder-decoder since it is normally designed to take in a 3-channel image, but instead takes 3 single channel slices as inputs. A feature matching layer was also applied to reduce noise and to correct pathological features. This model outperformed [192] in terms of PSNR and SSIM, and overall image quality is rated decent by radiologists. The model was trained on forty PET datasets from 39 participants where ground truth samples were reconstructed as the standard-dose and 1% low-dose PET scans were reconstructed using randomly undersampled data.
	[193]	SA-GAN	The network is a modified self-attention GAN where the input consists of 5 consecutive slices of 128x128 PET images. Similar to the non-local means filter, models trained with a self-attention module implicitly learn to suppress irrelevant regions in an input image while highlighting salient features. Both simulated and real patient data was used to train and test this model. Compared to other SA-GAN skews the proposed method leads to higher contrast in tumors and have sharper boundaries.

Table 8 (cont'd)

	[194]	cycleGAN	Full count PET images and low count PET sinograms were the two target domains for image-to-image translation. The cycleGAN generator architecture remains largely the same from the original paper except for the input and output layer size (now larger) and the number of residual blocks (5 instead of 3). A total of 30 clinical PET volumes (310 slices per volume) were used to train and test the model. Evaluation of PSNR and MSE revealed that the proposed method was better than other reconstruction methods (expectation-maximation, NLM denoising, and a vanilla GAN).
	[195]	Task-GAN	Here, a novel task-specific network is used in addition to the generator and discriminator. It aims to help regularize the training of the generator and complement the adversarial loss to ensure the output images better approximate the ground truth images. The task-network learns to refine the output of the generator to match the label of the image. For example, if the reconstructed image is supposed to have pathology present, the task-network would refine the generator output to make pathologies "clearer". 40 ultra-low-dose 1% PET images were reconstructed after random undersampling. Each PET volume consists of 89 2.78 mm-thick slices with 256 × 256 pixels.

2.4.2 Image Synthesis

As previously mentioned, GAN's most promising application is in data augmentation, where we can create diverse synthetic images to solve the problem of low volume labeled imaging datasets. Presented in Table 9 are GANs used for medical image synthesis for improving downstream segmentation, classification, and detection tasks. If the articles use privately curated datasets, a brief description is provided; otherwise if the articles use publicly available datasets, only the name is provided. For simplicity, novelty of network architecture is not described, and the results are focused on the promising performance of GANs.

Table 9. Image synthesis applications using GANs.

Modality	Ref	Generation and Downstream Task	Data	GAN Variant or Family	Results
MRI	[196]	Cardiac MR images for segmentation	33 short-axis cardiac MR image sequences, each 20 frames with 8 to 15 slices, for a total of 10,022 images.	DT-GAN	Proposed method achieves a mean Hausdorff distance (HD) of 2.98 mm ± 0.43 mm and a Dice score of 0.79 mm ± 0.10 mm for myocardium segmentation, which is superior to a previously described 23-layer U-net (HD = 3.04 mm ± 0.27 mm, Dice = 0.74 ± 0.04.

Table 9 (cont'd)

	[197]	Knee MRI for detection	OAI dataset + 25 heterogeneous MRIs locally collected through clinical routines	cGAN	Training a cGAN on the OAI dataset led to poor performance on the clinical dataset (Dice = 0.519, HD = 6.23). However, using this pretrained model and transfer learning to the clinical dataset improved the mean Dice score to 0.819 and the mean HD to 1.46.
	[198]	Brain MRI with tumors for detection	BRATS 2013	FixedGAN	The downstream ROC response of an object detector (ResNet 50) is evaluated given images synthesized by Fixed-Point GAN, Star-GAN, and CAM. Fixed-Point GAN achieved an AUC of 0.98 and a sensitivity of 84.5% at 1 false positive per image, outperforming StarGAN who had an AUC of 0.46 and a sensitivity levels of 13.6%. CAM, however, outperformed both with an AUC of 0.99 and a sensitivity of 60% at 0.037 false positives per image.
	[132]	Brain MRI with tumors for detection	BRATS 2016	PGGAN, SIMGAN, and UNIT	The Visual Turing test of the all three GAN generated images revealed that 75% of images contained realistic texture and tumor appearance. The downstream ROC response of an object detector (ResNet 50) was also evaluated given augmented data. Without synthetic images, ResNet 50 achieved an accuracy, sensitivity, and specificity of 93.14, 90.91, and 95.85, respectively. The addition of augmented data, whether it be through classical transformations or with GANs, all improved object detector performance. The addition of classical DA and UNIT images resulted in the highest accuracy and sensitivity (96.7 and 97.48, respectively).
	[199]	Brain MRI with tumors for classification	BRATS 2016	PGGAN	This model was an improvement on the PGGAN described in [132]. All evaluation methods remained the same. Without synthetic images, ResNet 50 achieved an accuracy, sensitivity, and specificity of 90.06, 85.27, and 97.04, respectively. This improved to 91.08, 86.60, and 97.60 with the addition of PGGAN synthetic data.

Table 9 (cont'd)

	[200]	3D Brain MRI for detection	ADNI and BRATS 2018	3D α-GAN	To quantitatively evaluate synthetic 3D MRI images, the maximum mean discrepancy (MMD) and multi-scale SSIM metrics were used. Compared to other 3D GANs such as 3D-VAE-GAN and 3D-WGAN-GP, the proposed model achieved the lowest MMD of 0.072 and the second closest MS-SSIM to real images (0.829).
	[201]	Brain MRI with Parkinson's for classification	PPMI Dataset	Modified DC GAN	The pre-trained Le-Net-5 network was used as a classifier to detect Parkinson's from brain MRIs. Synthetic GAN data were added to the training set which yielded an accuracy, specificity, and sensitivity of 88, 87.14, and 87.92, respectively, compared to a baseline of 84.67, 83.76, and 84.13 without synthetic data.
	[202]	Diffusion MRI for classification	Human Connectome Project	cycleGAN	Using a cycleGAN to translate between structural and diffusion MRI images, the authors were able to achieve reasonable MSSIM values (0.839 ± 0.014 and 0.937 ± 0.008 for synthetic fractional anisotropy and mean diffusivity images, respectively). The authors did not provide a baseline to compare to.
Retinal Fundus	[172]	Vessel segmentation map of retinal fundus images	DRIVE, CHASE-DB1, and STARE	RV-GAN	RV-GAN outperforms other segmentation neural networks and GANs for all 3 datasets. RV-GAN beats traditional U-net, DenseNet, IterNet, and SUD-GAN in all metrics, with F1, specificity, accuracy, AUC, mean IOU, and SSIM of 0.869, 0.996, 0.979, 0.988, 0.976, and 0.9237, respectively. Surprisingly, RV-GAN only loses to M-GAN in terms of sensitivity (0.793 vs 0.835).
	[203]	Vessel segmentation map of retinal fundus images	DRIVE, STARE	DRPAN	DRPAN showed to match or barely exceed other leading segmentation algorithms. All CNN performance in terms of accuracy, sensitivity, and specificity were within 0.01, and were not statistically significant.
	[204]	Vessel segmentation map of retinal fundus images	DRIVE, STARE	RetinaGAN	RetinaGAN achieves statistically significant improvement in AUROC, precision, and recall, surpassing the current state-of-the-art method by 0.2 − 1.0% in ROC and 0.8 − 1.2% in precision and 0.5 − 0.7% in recall.

Table 9 (cont'd)

	[205]	SLO images with fundus disease for classification	4590 scanning laser ophthalmoscopy (SLO) images of size 2600 x 2048 were captured from teenage (< 18yo) patients.	AMD-GAN	Compared to ResNet 50, the proposed AMD-GAN classifier resulted in higher accuracy, precision, recall and F1 (ResNet 50 = 77.13, 68.90, 68.42, 68.65, 87.10, AMD-GAN = 84.75, 79.15, 82.15, 80.41, 97.25, respectively).
	[206]	Vessel segmentation map of retinal fundus images for segmentation	DRIVE and DRISHTI-GS	Pix2pix, cycleGAN	Two types of image-translation models were compared in generating vessel segmentations from retinal fundus images. Multiple skews for the generators were also explored (U-net, ResNet6, and ResNet9). The best performing method for PSNR was the pix2pix with ResNet9 generator (25.36) with the worst performing model cycleGAN with the ResNet9 generator (21.9). The best performing method for SSIM was pix2pix with U-Net generator (0.911) and the worst performing was the cycleGAN with ResNet9 generator (0.877).
	[207]	Retinal fundus images with different grades of diabetic retinopathy for classification	EyePACS	DR-GAN	Here, a model to generate fundus images with controllable diabetic retinopathy severity levels, which can be used to augment images and improve the performance of the DR grading models by mitigating class imbalance. VGG16, ResNet 50, and InceptionV3 were used to assess the classification performance with/without the addition of synthetic images. Overall, DR-GAN seems to solve the class imbalance problem and the addition of synthetic images significantly improves the accuracy of all three classifiers.
	[208]	Vessel segmentation map of retinal fundus images	Retinal color fundus dataset with 6,432 retinal images was collected from local hospitals. DRIVE was also used for segmentation tasks.	SkrGAN	SkrGAN achieves a MS-SSIM of 0.614, and FID of 27.59, which are all better than DCGAN, ACGAN, WGAN, and PGGAN. Additionally, a U-net segmentation model performed better with SkrGAN generated images (sensitivity = 0.846, accuracy = 0.951) than without (sensitivity = 0.778, accuracy = 0.948).

Table 9 (cont'd)

Dermoscopy	[209]	Optical photos of skin cancer lesions for segmentation	ISIC Skin Lesion Challenge Dataset	DAGAN	The authors compared the proposed DAGAN segmentation network with U-net and its many variations. Overall, DAGAN had the highest Dice coefficient (0.859) where the next highest was a U-net with skip and dense connections (0.832). DAGAN also highest accuracy and specificity, but only the second highest sensitivity.
	[210]	Optical photos of skin cancer lesions for classification	ISIC Skin Lesion Challenge Dataset	cGAN	In this paper, a CNN was trained with/without synthetic data to classify skin cancer into benign or malignant. Without GAN images, the CNN had an accuracy, sensitivity, specificity, and F1-score of 53%, 0.51, 0.57, and 0.5, respectively. With GAN images, the CNN had an accuracy, sensitivity, specificity, and F1-score of 71%, 0.68, 0.74, and 0.7, respectively.
	[131]	Optical photos of skin cancer lesions for segmentation	SMARTSKINS, Dermofit image library, ISIC Challenge dataset	cycleGAN	The cycleGAN generated segmentation maps was compared to simple adaptive thresholding, and other neural networks (Gossip and Reduce Mobile Deeplab). Both cycleGAN Dice coefficient and Jaccard Index (JI) were superior to other segmentation techniques (92.74 and 86.7, respectively).
X-ray	[211]	X-ray images with bone lesions for detection	514 adult X-ray images of tibia, humerus, and femur bone lesions	cycleGAN	A trained CNN classifier had a baseline lesion detection sensitivity, specificity, and AUC of 0.9, 0.776, and 0.876, respectively. The CNN trained with real and cycleGAN generated data yielded values of 0.84, 0.842, and 0.924, respectively.
	[166]	X-ray images with various diseases for classification	ChestX-ray8	PGGAN	FID score indicated generally high realism and quality for most disease classes. Overall synthetic images had an FID of 8.02, where closer to 0 is ideal. Images that are supposed to contain edema or pneumonia were found to be of less quality (FID 59.4 and 32.05, respectively). Real images were identified as such by radiologists 73% (95% CI 63, 82) of the time, while generates were identified as real 61% (95% CI 51, 70) of the time, with both groups more likely than chance to be identified as real.

Table 9 (cont'd)

	[171]	Chest X-ray with Covid for detection	IEEE Covid Chest X-ray dataset, COVID-19 Radiography Database, and COVID-19 Chest X-ray Dataset Initiative	CovidGAN	A variant of the ACGAN developed to generate synthetic, Covid-19 positive chest x-rays. A VGG16 detector was transfer learned with/without CovidGAN images. Accuracy, sensitivity, and specificity of the VGG16 classifier improved from 0.85, 0.69, and 0.95 to 0.95, 0.90, 0.97, respectively, when adding the synthetic chest radiographs.
	[212]	Mammography images of varying levels of breast density for classification and segmentation	INbreast dataset	cGAN	The best performing model segment the dense regions well with an accuracy, Dice coefficient, JI of 98%, 88%, and 78%, respectively. Compared to other NN segmentation techniques, the cGAN resulted in the highest precision, sensitivity, and specificity of 97.85%, 97.85%, and 99.28%, respectively, for breast density classification. FCN-8 and VGG16 yielded 0.748, 0.997, 0.69 and 0.832, 0.996, 0.66, respectively.
	[213]	Mammogram mass images for segmentation	INbreast dataset, and a private dataset of 549 mammograms containing 376 mass regions.	cGAN	The U-net segmentation model performed the best when using a combination of INbreast, privately curated data, and cGAN generated images with a JI, Dice score, and accuracy of 79.35, 88.2, and 88.8, respectively. Without the additional synthetic images, the segmentation performance fell to 77.23, 86.77,and 87.29, respectively.
	[214]	Rib suppressed chest X-ray for disease detection	LIDC-IDRI, TianChi AI competition 2017, 2019	RSGAN	The important metric to note is Weber Contrast (WB), which provides an estimation of rib-suppression performance on the boundaries for chest x-ray images by calculating the contrast gap between the rib-suppressed region and the background, where a lower contrast gap is more valuable. RSGAN yielded in a WB of 1.96 while other NNs yielded a WB of 3.49 (U-net) and 2.36 (ResNet).
	[215]	Chest x-ray for disease classification	CheXpert	cGAN, DenseNet	The DenseNet-121 pretrained network is used to transfer learn disease classification. A cGAN was developed to augment chest x-rays with either lung lesions, pleural effusion, or fractures. Overall, the addition of synthetic images only improves performance in low-volume scenarios (<10% of real dataset). The authors found that training with only their full dataset is better than training with augmented data.

Table 9 (cont'd)

	[216]	Chest X-ray for lung segmentation	JSRT and Montgomery County Datasets	cGAN	The proposed auxiliary U-net GAN with multiple residual blocksresulted in a Dice score of 0.979 when training on the JSRT dataset, where a normal U-net yielded 0.946.
	[217]	Rib suppressed chest X-ray for disease detection	JSRT dataset	SFRM-GAN	The proposed model is a mix between WGAN-GP and pix2pix. The proposed model resulted in a mean PSNR of 43.588, mean RMSE of 0.00025, and mean SSIM of 0.989. Compared to pix2pix which resulted in 41.37, 0.0004, and 0.982, respectively.
CT	[218]	High resolution CT image reconstruction	29 post-registered ankle CT scans of low- and high resolution which resulted in 14,000 matching pairs of low- and high-resolution patches of size 64×64	GAN-CIRCLE	The goal of this study was to reconstruct high resolution CT images to assess trabecular bone microstructures. To evaluate the GAN, they chose to use concordance correlation coefficient (CCC) to measure the agreement in the microstructure. Evaluating real low-resolution images, the Tb thickness and Tb volume CCC scores were 0.66 and 0.83, respectively. This increased to 0.95 and 0.88 when evaluating synthetic high-resolution images.
	[219]	CT scans for organ segmentation	NIH Pancreas-CT dataset	cycleGAN	When the kidney model was trained with CycleGAN augmentation techniques, performance increased dramatically (from a Dice score of 0.09 to 0.66, $p < 0.001$). Improvements for the liver and spleen were smaller, from 0.86 to 0.89 and 0.65 to 0.69, respectively.
	[220]	CT scans for nodule segmentation	LIDC-IDRI	AUGAN	This study compared its AUGAN to other segmentation networks (FCN, U-Net, and U-net GAN). For Dice coefficient, AUGAN yielded the highest score of 0.849 while the U-net GAN yielded a the second highest score of 0.835. Similar performance is seen when evaluating JI, with AUGAN at 0.750 and U-net at 0.733.
	[221]	CT scans for nodule segmentation	LIDC-IDRI	3D cGAN	Only qualitative approaches were used to assess image quality and nodule realism. Generally, the cGAN generated images do appear realistic with a diverse set of nodule presentations. The nodules also appear to be tunable in size.
PET	[222]	Low dose to Standard dose PET for detection (LPET to SPET)	Phantom Brain Dataset and Real Human Brain Dataset	AR-GAN	The proposed AR-GAN outperforms other state-of-the-art GANs in PSNR and SSIM (28.106 and 0.891, respectively). Stack-GAN, GDL-GAN, and Ea-GAN resulted in PSNR, 26.77, 27.07, and 26.39, and SSIM values of 0.884, 0.886, and 0.882, respectively.

Table 9 (cont'd)

	[223]	PET images for Alzheimer's classification	ADNI1	DCGAN	The authors provided no other neural network as a point of comparison. The mean PSNR of the DCGAN generated images was 32.83 and the SSIM was 77.48.
Ultrasound	[224]	High resolution US images for segmentation	Privately curated set of B-mode US images contains 6054 of chest, 1231 of hip joints, and 3261 of ovaries	PGGAN (named spGAN)	Standard interpolation was used as the baseline method for generating high-resolution US. For the chest and ovary datasets, SpGAN had better FID SSIM, and LPIPS scores (chest = 36.36, 0.751, 0.168, ovary = 47.11, 0.497, 0.795) compared to standard interpolation (chest = 65.41, 0.7428, 0.210, ovary = 63.15, 0.4332, 0.8901). For the hip joint dataset, the interpolation technique only surpassed spGAN in SSIM.
	[225]	Breast US for cancer segmentation	DBUI, SPDBUI, ADBUI	ASS-GAN	The ASS-GAN was evaluated on IoU, accuracy, and Dice and compared to other NN segmentation methods (U-net, DeepLabV3, AttenU-Net). The proposed method outperformed all other NNs with an IoU, acc, and Dice scores of 0.7683, 0.9760, 0.8690, respectively, for the DBUI dataset. The performance was approximately the same for other dataset, and ASS-GAN was still the best for each.
	[226]	Breast US for radiomics and cancer classification	Privately curated dataset includes 1447 tumor-present images from 357 female patients.	TripleGAN	Breast mass classification accuracy reached 90.41%, sensitivity 87.94%, specificity 85.86% with TripleGAN synthetic data + real data. Compared to other state-of-the-art methods such as GAN, DCGAN, and InfoGAN, the proposed method had significantly higher metrics.
	[227]	Thyroid US for nodule classification	TDID	Res-GAN	Thyroid nodules were classified by a either ResNet18 or Res-GAN into malignant or benign. Res-GAN had a classification accuracy, specificity of 92.2, 86.5, and 95, respectively, which is significantly higher than ResNet18 (82.2, 66.2, 89.8, respectively).
	[228]	US images for bone segmentation	Privately curated dataset containing 1235 in vivo B-mode US images containing either radius, femur, spine, or tibia.	patchGAN	Pix2pix, DCGAN, and WGAN segmentation performance were compared to the proposed patchGAN network. The IoU of patchGAN generated segmentation maps was 0.9357 with a Dice score of 0.9640. This was significantly better than other models. WGAN performed the worst with an IoU of 0.8726 and a Dice of 0.9158.

It is important to note that while most studies have shown that the addition of augmented data will improve the performance of NNs, to date, most gains are only moderate. Most commonly, synthetic data results in 1-4% increase in sensitivity, specificity, accuracy, or respective metric. Additionally, while more complex GANs correlate to better downstream performance, this also means that the networks become less generalizable as they are heavily tailored to a singular application.

2.4.3 Cross-Modality Translation

Cross-modality translation refers to the generation of an image of one medical imaging modality from that of another, for example, CT to MR. This is beneficial for the patient as it may decrease their number of scans, decrease the risks from imaging (contrast reactions, radiation dose, etc), and decrease healthcare costs. Healthcare providers may also benefit from cross-modality translation as it will increase patient throughput and decrease scanning turnaround time, along with allowing for less radiotracer production runs [229]. Additionally, registration mismatch between modalities will be eliminated if translation models are used. Highlighted in Table 10 are recent developments in cross-modality translation. Again, for simplicity, the network architecture of the GANs is not included.

Table 10. Cross-modality image translation applications using GANs. Interestingly, while MR and PET are the more expensive and less available modality, most studies are trying to derive CT images from either MR or PET.

Modalities	Ref	Data	GAN Variant or Family	Results
MR and PET	[230]	ADNI	BiGAN	Given MR brain images, BiGAN outputs the corresponding PET image. BiGAN outperforms (cycleGAN) with the highest PSNR (27.36) and SSIM (0.88), indicating that the quality of the synthetic images derived from the proposed method is closest to the real PET images. CycleGAN generated PET images resulted in a PSNR of 24.68 and a SSIM of 0.78.

Table 10 (cont'd)

	[231]	ADNI	GLA-GAN	The GLA-GAN produces FDG PET images given structural MR images for the downstream classification of Alzheimer's disease. A comparison between U-net, cycleGAN, and GLA-GAN is made. GLA-GAN is significantly better than other methods in terms of SSIM, PSNR, and MAE (96.88, 29.32, and 0.014, respectively). U-net performed better than CycleGAN in terms of SSIM and PSNR but had a higher MAE.
	[232]	ADNI	E-GAN	E-GAN transforms FDG-PET images to T1 weighted MRI images. DCGAN, WGAN, and pix2pix were also trained to perform this task. PSNR, SSIM, and MAE are used to evaluate all 4 GANs. E-GAN performs the best with scores of 28.16, 0.75, and 105, respectively. Pix2pix performed the second best in all metrics, with PSNR, SSIM, and MAE scores of 24.76, 0.61, 295, respectively.
	[233]	ADNI	TPA-GAN	The goal of TPA-GAN is to use 3D MRI to generate corresponding FDG PET volume and then use both volumes to classify brain diseases. TPA-GAN PET generation was compared to vanilla GAN, AttentionGAN, cycleGAN, and PA-GAN. TPA-GAN images results in a mean SSIM, PSNR, and MSE of 0.915, 29.0, and 184, respectively. The next best performing model (PAGAN) resulted in scores of 0.913, 28.5, and 204, respectively.
	[234]	ADNI	GANBERT	BERT was incorporated into a standard U-net GAN as a secondary discriminator. Two variations of GANBERT were trained; one with a CNN and BERT discriminator, and one with only BERT as the discriminator. Both models were compared to a standard pix2pix network. GANBERT + CNN yielded a PSNR, SSI, and RMSE of 57.58. 0.27. and 0.80, respectively. GANBERT-only yielded 56.53, 0.31, and 0.91, and pix2pix resulted in scores of 50.41, 0.0, and 1.85, respectively.
CT and MR	[235]	Privately curated 50 3D contrast-enhanced thoracic-abdominal CT and abdominal MRI images.	cycleGAN	This two-stage organ detection method uses cycleGAN to translate images from MR to CT, which were then used to train a separate NN (Yolov5) to detect organs. Mean average precision of Yolov5 decreased from 8.66mm to 7.95mm when adding synthetic data.
	[236]	SpineWeb library	cycleGAN	Another cycleGAN was used to translate 3D MR to 3D CT volumes. The cycleGAN generated CT images resulted in an average Dice coefficient of 0.83 and a mean landmark error of 2.2mm. The authors provided baseline comparison for these scores. However, they did qualitatively compare another GAN [237] single-slice model provide a comparison

Table 10 (cont'd)

	[238]	Privately curated, same-day MR and CT images were acquired.	cycleGAN	The proposed modified cycleGAN generated CT images were compared to a normal cycleGAN, a cGAN, and a DCNN. For MAE and PSNR, the modified cycleGAN was superior with scores of 0.0416 and 36.11, respectively. The next best model was the normal cycleGAN with an MAE of 0.0465 and a PSNR of 37.10.
	[239]	Atlas project	cGAN	The proposed cGAN borrows its generator architecture from pix2pix architecture but has 6 or 9 residual blocks. SSIM, MAE, PSNR, and MSE are compared between the cGAN, pix2pix, and a U-net. The pix2pix model is worse than both the 6, and 9 block cGAN. The 9 residual block cGAN produces the highest quality images, with PSNR, SSIM, MAE, and MSE of 21.4, 0.823, .0.3, and 0.01, respectively.
PET and CT	[240]	Privately curated images from 169 patients with established coronary artery disease undergoing hybrid coronary 18F-NaF PET and contrast CT angiography.	cGAN	A cGAN was trained to synthesize CT images from PET. Target-to-background (TBR) and standardized update values (SUV) were used to assess the quality of the CT image alignment given the ground truth CT. The correlation of TBR for the cGAN was 0.31 and the SUV was 0.26 to human registration, indicating excellent correlation of observer and the proposed method. Additionally, the generation time of the GAN was only 27.5 seconds on average, which is 33 times faster than humans.
	[241]	A privately curated dataset contained 60 CT and PET was constrained to slices in the region of the liver.	FCN + cGAN	MAE and PSNR was used to evaluate the performance of FCN, cGAN, and a fusion of both in generating PET from CT images. On average, the combined network achieved the lowest MAE (0.72) and highest PSNR (30.22), with FCN at a close second (MAE = 0.74 and PSNR = 30.05).
	[242]	Privately curated dataset containing 1935 brain PET and CT scans were taken.	cycleMEDGAN	Here, authors expanded on their previous MEDGAN [130] by adding a cycle consistency loss as well as a cycleGAN network structure. Here, cycleMEDGAN aims to translate from PET to CT. It also outperformed normal cycleGAN and UNIT models in terms of SSIM and PSNR (0.911 and 24.08, respectively). The normal cycleGAN achieved the second best performance with a SSIM of 0.896 and a PSNR of 23.35.

In conclusion, despite being a recent innovation, GANs have been widely applied to numerous medical imaging applications. GANs and their variants have had great success in augmenting data in multiple modalities and have been used for downstream reconstruction, segmentation, classification, and detection tasks, among others. Consistently, studies have shown

that the addition of synthetically generated images to the training set of other NNs improves performance. However, there are still challenges that need to be addressed, such as the limited generalizability of models. The above mentioned GANs are highly specialized and often do not translate well to other image generation tasks. Also, very few studies have verified the image quality and realism of the generated medical images using a) human observers or b) a task-based assessment against human performance due to the challenge of performing these comparisons. Finally, only a few studies reported whether a certain ratio of synthetic to real images yielded optimal performance. Nonetheless, the progress made in the field of GANs in medical imaging is promising and holds significant potential for improving the accuracy and efficiency of medical diagnosis and treatment.

CHAPTER 3: NEAR-PAIR PATCH GENERATIVE ADVERSARIAL NETWORK

These efforts were partially presented at the IEEE Nuclear Science Symposium, Medical Imaging Conference 2022, and has been submitted to the SPIE Journal of Medical Imaging.

Data scarcity in machine learning medical imaging applications is a major issue. Deep learning-based methods require a large volume of training data, which may be difficult to acquire due to several reasons, such as lack of standardization, lengthy curation process, accessing HIPAA compliant images, and the need for expert labeling. Data augmentation is a common solution to mitigate this issue. Recently, sophisticated data augmentation methods are based on a class of NNs called Generative Adversarial Networks (GANs), which generate new images of high perceptual quality. Here, we present a method to support distant supervision of object detectors using generated synthetic pathology-present labeled images. Our proposed method, named near-pair patch cycleGAN (NPP-cycleGAN), employs the previously proposed cyclic generative adversarial network (cycleGAN) with two primary innovations: 1) use of "near-pair" pathology-present regions and similar pathology-absent regions for training and 2) the addition of a realism metric (Fréchet Inception Distance) to the generator loss term. The NPP-cycleGAN is then used to augment data by synthetically generating pathology on pathology-absent images, which can then be used to train object detectors. We train and test the method with 2800 fracture-present image patches from 1109 unique pediatric chest radiographs. In a blinded observer study, we presented four expert pediatric radiologists with either a real fracture absent image, a real fracture present image, or a synthetic fracture present image and asked them to score 1-5 the likelihood of a fracture (1 = Definitely not a fracture, 5 = Definitely a fracture). Results showed that real fracture absent images scored 1.71 ± 0.99, real fracture present images 4.14 ± 1.23, and synthetic fracture present a 2.51 ± 1.24. These results suggest that the proposed GAN can generate high quality fracture-

present pediatric chest radiographs.

3.1 Introduction

Rib fractures in pediatric patients are a sentinel injury for non-accidental trauma, making accurate and timely detection of these injuries crucial in protecting the well-being of children. Unfortunately, between 80-100% of rib fracture cases in young children are a result of child abuse [243], [244]. This statistic is particularly concerning because over two-thirds of rib fractures can be missed during first reads by radiologists [245], and in our recently published study expert radiologists achieved a reader-to-reader F2 score of only 0.73 [246]. While the importance of detecting rib fractures is high, it is a particularly difficult task even for experienced radiologists. The difficulty of detection is partially due to the diverse presentation of fractures. While certain fractures are easy to detect, with obvious signs of bony displacement and/or healing (including subperiosteal new bone formation, callus bridging, and medullary sclerosis) (Figure 23), challenging fractures are much less conspicuous and more difficult to diagnose, showing little to no signs of bony displacement or healing.

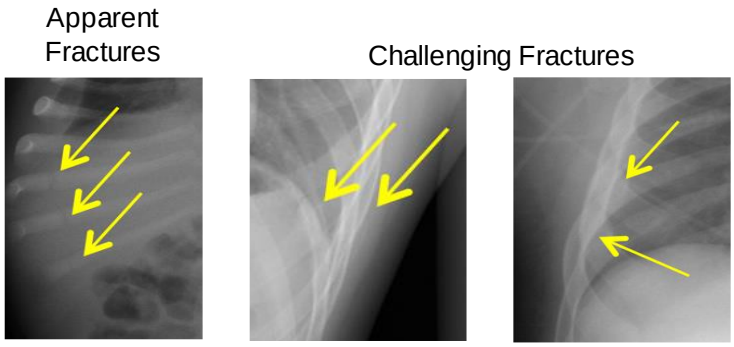

Figure 23. Examples of apparent and challenging fractures. Transverse fractures are readily apparent vs. challenging fractures that show no sign of displacement.

Recent studies have shown machine learning models can match human performance in the detection of rib fractures in children. One study trained a deep convolutional neural network on a dataset of 845 CT scans from children achieved a sensitivity of 43% and a specificity of 88% in

their test set [247]. Another study used similar techniques but was trained on a dataset of 300 radiographs and achieved a sensitivity of 91.3% and a specificity of 90% [248]. Our previous study proposed a method entitled "avalanche decision" was motivated by the reality that pediatric patients commonly present with multiple clustered fractures [246]. We improved two leading single stage detectors, RetinaNet and YOLOv5, with this decision scheme. The networks were then trained on 1109 radiographs and yielded RetinaNet and RetinaNet+Avalanche F2 scores of 0.55 and 0.65, respectively. F2 scores of base YOLOv5 and YOLOv5+Avalanche were 0.58 and 0.65, respectively. One underlying issue among all prior studies is that they were trained with a relatively small volume dataset.

Small training datasets are commonly augmented with a variety of methods. More sophisticated NN based augmentation methods have recently become popular. Both cycleGAN and pix2pix, belonging to a class of NN called Generative Adversarial Networks, have been used in a variety of medical imaging applications, such as image segmentation, lesion detection, and retinal image analysis [127], [131], [172], [211], [235] (see tables 4-6 in chapter 1).

We propose a novel GAN approach, where near-pair image patches are used as inputs to a cycleGAN, to translate image patches of rib fracture absent radiographs to rib fracture present radiographs. We hypothesize that the "near-pair" aspect of our training data will allow for more constrained training and ultimately more successful translation without the use of true 1-to-1 paired data. While our method is specifically tested with rib fracture radiographs, this methodology is potentially generalizable for data augmentation of any image dataset that seeks to detect focal pathology.

3.2 Methods

Rib Fracture Dataset

Our dataset was collected through an IRB-approved study at Seattle Children's Hospital.

The dataset contains 1109 unique patients, of which 624 are fracture present and 485 are fracture absent. There are 241(34.2%) female and 463(65.8%) male patients. The average age of patients is 268.76 ± 784.93 days (range 0 − 6935, median 84, IQR 196). After removing outliers (missing, age = 0, or age ≥ Q3 + 1.5IQR), the average age of patients is 128.11 ± 111.43 days (range 1 − 476, median 84, IQR 140). The images are chest radiographs in an anterior-posterior perspective, provided in DICOM file format. Ground truth annotations of rib fracture locations were provided by eight board-certified pediatric radiologists. When grading the fracture present section of the dataset the radiologists were given prior knowledge that at least one fracture was present in each image, thus there is a slight bias towards the labeling performance of the radiologists.

3.2.1 Near-Pair Patch cycleGAN

Training a model on localized image patches instead of the whole image poses many benefits: 1) computationally cheaper, 2) faster training, and 3) multiple training pairs can be extracted from one radiograph. Finally, the generation of localized patches will greatly benefit the training of object detectors because we know the exact bounding box locations surrounding the pathology.

Near-Pair Generation

Our dataset contains 2800 labeled fractures from 515 patients. The average bounding box size was 78 x 70 pixels. To give our model contextual information surrounding the fracture, we standardized the size of our patches to 2.5 cm x 2.5 cm (128x128 pixels), where the center of our patches is the same as the center of the original bounding box. The near-pair (fracture-absent) patch was manually selected from the same radiograph. The general rules for selecting a near pair was to first select a patch on the contralateral rib and horizontally flip the image; If that patch happened to contain a labeled fracture, then select a fracture absent patch with similar orientation, most commonly a couple ribs above or below the target patch. Examples of near pairs can be seen

in Figure 24. These near-pair patches are then normalized at the patch level using a robust standard scalar method using a 98% interquartile range (IQR).

$$Scaled\ image = \frac{Original\ Image - Median\ pixel\ value}{Original\ Image\ IQR} \quad \textbf{(17)}$$

Normalization of the patch allows our pixel values to be in the same range as our activation functions, usually between 0 and 1. This allows for less frequent non-zero gradients during training, and therefore the neurons in our network will learn faster.

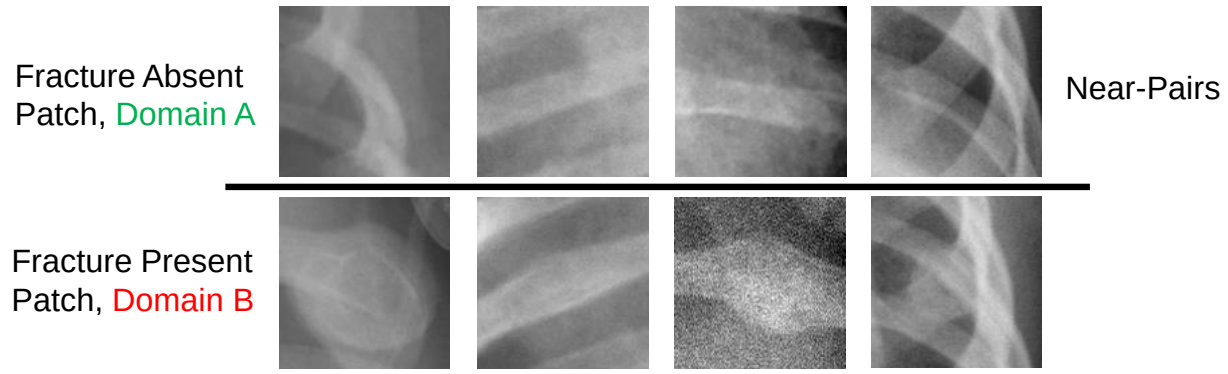

Figure 24. Examples of Near-Pair Patches. Each patch is manually selected from the same radiograph. If no closely resembling patch within the same radiograph is fracture absent, then we select from an age, sex, and chest volume matched image.

Training Details

The 2800 near-pair patches were then used to train a cycleGAN, with the overall and generator architecture presented in Figures 25 & 26. We trained the cycleGAN using all 2800 fracture-present and 2800 fracture-absent near-pair image patches in the training set. We used a batch size of 4, learn rate of 0.0002, a lambda of 10, and training was optimized by Adam. The network converged in approximately 90 epochs using an NVIDIA V100S GPU.

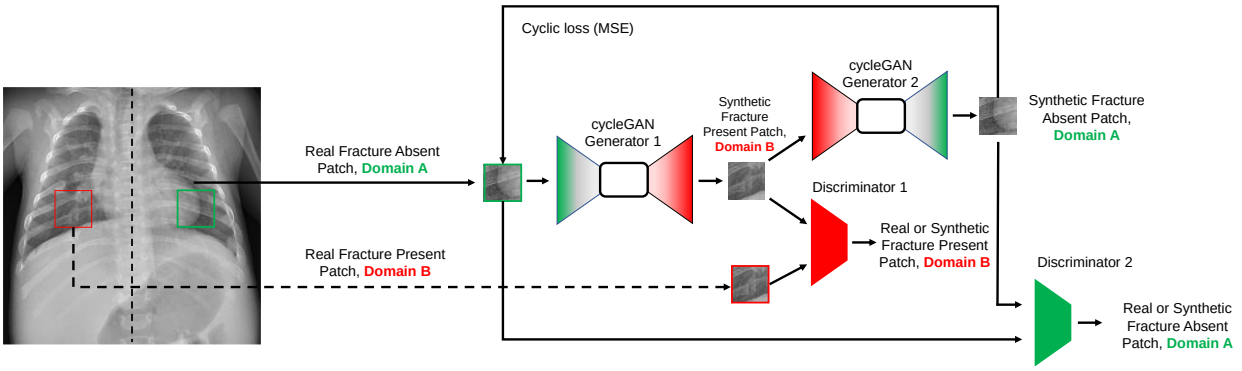

Figure 25. NPP-GAN training flowchart. Two sets of generator/discriminator pairs are trained simultaneously using near pair patches. Generator 1 converts real fracture-absent patches to fracture-present patches. Discriminator 1 distinguishes between synthetic fracture-present and real fracture-present patches. Generator 2 removes pathology to create synthetic fracture-absent patches. Discriminator 2 distinguishes between synthetic and real fracture-absent patches. The novelty of this work is the use of near-pair real patches derived from similar regions of the image from the same base image.

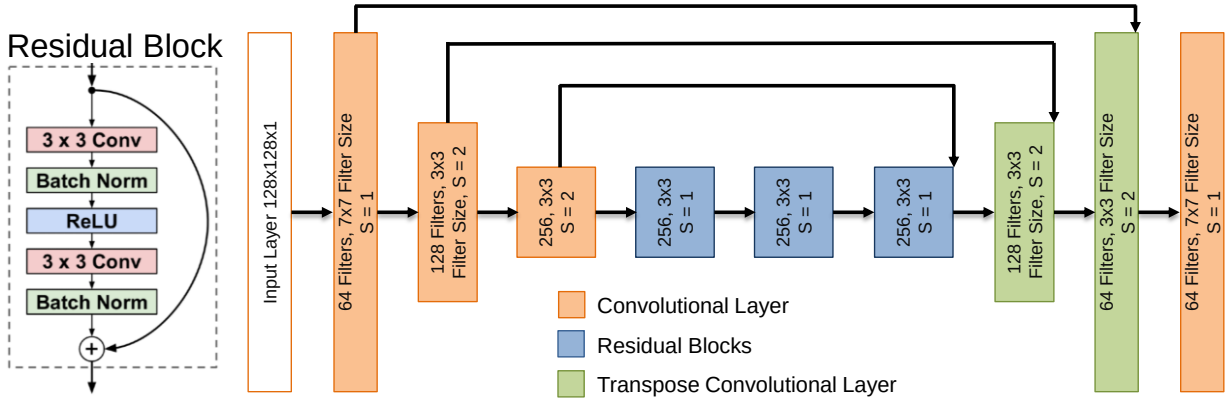

Figure 26. NPP-GAN Generator architecture. An input layer of size 128x128x1 goes through an initial 2D convolution layer with 64 filters and a filter size of 7x7. Then it goes through 2 downsampling blocks, 3 residual blocks, 2 upsampling blocks, and then a final convolution layer. An individual residual block's architecture is presented on the left. Each block is followed by an instance normalization and a ReLU layer, but these have been omitted for simplicity.

3.2.2 Unpaired cycleGAN

To assess the benefit of using near-pair patches, we trained another conventional cycleGAN with unpaired data to compare with our NPP cycleGAN. To create our unpaired fracture-absent patches, we randomly selected 128x128 regions of known fracture-absent

radiographs. Training parameters and architecture of the unpaired cycleGAN were identical to our NPP cycleGAN, although it converged in approximately 100 epochs.

3.2.3 Fréchet Inception Distance Near-Pair Patch cycleGAN

As mentioned in Section 1.3.4, a common method to evaluate realism of the synthetically generated radiographs is the Fréchet Inception Distance (FID) [169]. The Fréchet Inception Distance functions by embedding a set of real and synthetic images in the final average pooling layer of an Inception Net [162] pre-trained on ImageNet [249]. The two sets are assumed to be multivariate Gaussian distributions with the average and covariance of each utilized to calculate the Fréchet distance, also known as the Wasserstein-2 distance. This distance reflects the difference in the average features extracted from each image set based on the learned kernels of the Inceptionv3 Net model. The distance has been demonstrated to be consistent with human judgement of visual quality and more resistant to noise than prior approaches for natural images [168].

We used the FID as a metric to evaluate the quality of the generated pathology and as an innovation in the proposed generator 1 by creating a new generator loss term using the sum of FID and the conventional cycleGAN loss (Equation 13). Therefore, our new loss function is now:

$$\mathrm{L_{tot}}(\mathrm{G_A}, \mathrm{G_B}, \mathrm{D_A}, \mathrm{D_B}) = \mathrm{L_{GAN}}(\mathrm{G_A}, \mathrm{D_B}, \mathrm{X}, \mathrm{Y}) + \mathrm{L_{GAN}}(\mathrm{G_B}, \mathrm{D_A}, \mathrm{Y}, \mathrm{X}) + \lambda \mathrm{L_{Cyc}}(\mathrm{G}, \mathrm{F}) + \beta \mathrm{L}_{FID}(\mathrm{G}, \mathrm{F}) \quad \textbf{(18)}$$

Where β controls the weight of the FID loss term. The intuition is that the FID loss term will enforce the transformed image from the first generator (fracture-present) to "look realistic" compared to a random real fracture-present patch. The training set images, training parameters, and generator architecture are identical to our NPP cycleGAN, but converged in approximately 120 epochs.

3.2.4 Blinded Observer Study

A total of 90 images (30 Real Fracture Present, 30 Real Fracture Absent, 30 Synthetic Fracture Present) were randomly presented to four pediatric radiologists. The synthetic fracture present radiographs were generated using the FID-NPP model. Each image contains a 200x200 pixel bounding box highlighting the portion of the radiograph we want the radiologists to focus on. The radiologists were asked on a 1-5 scale if there is a fracture within the box (1 = Definitely not a fracture, 2 = Unlikely a fracture, 3 = May be a fracture, 4 = Likely a fracture, 5 = Definitely a fracture).

3.2.5 Full Radiograph Generation

Cumulative Informed Fractures

To generate full radiographs, we first select a fracture absent radiograph and then randomly sample 2.5 cm x 2.5 cm patches from the rib cage. The patch sample selection was guided by a heatmap of common fracture locations from our dataset (Figure 27). Approximately two-thirds of fractures were along the oblique ribs, and our generated radiographs reflect this distribution. Each radiograph had between 1-6 synthetic fractures added to follow typical fracture frequency in this patient population.

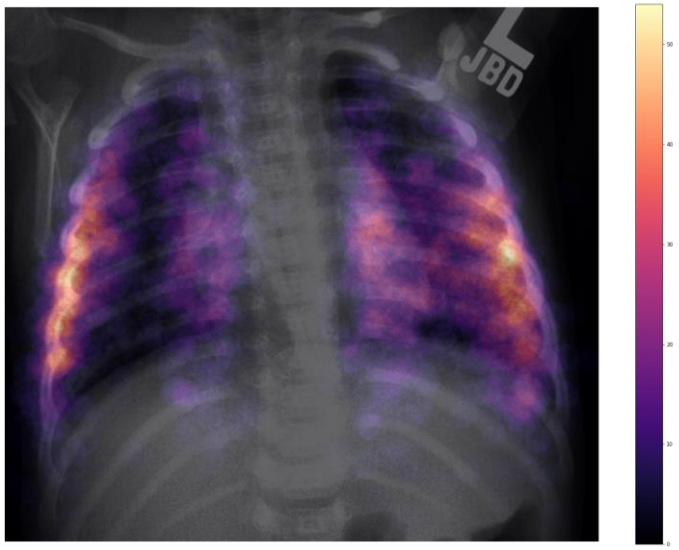

Figure 27. Cumulative distribution of the common locations of fractures on a reference radiograph. Approximately 2/3rds of fractures appear on the oblique rib cage. Generated from fracture-present radiographs where each fracture was mapped to a common atlas space. The sum of all fractures in the common space are presented on a representative chest radiograph.

Poisson Inpainting

After patch selection, generator 1 was then used to convert the selected fracture absent patch to a synthetic fracture present patch. The patch was reinserted into its original radiograph using a method we developed based on a Poisson blending technique (Figure 28) [250]. The intuition behind Poisson blending, also known as inpainting, is that to color match two different domains, the gradient of the images is more important than the intensity. Therefore, the method tries to replace the gradients of the target image with the gradients of the source image, while overall intensity is matched to the target image [251].

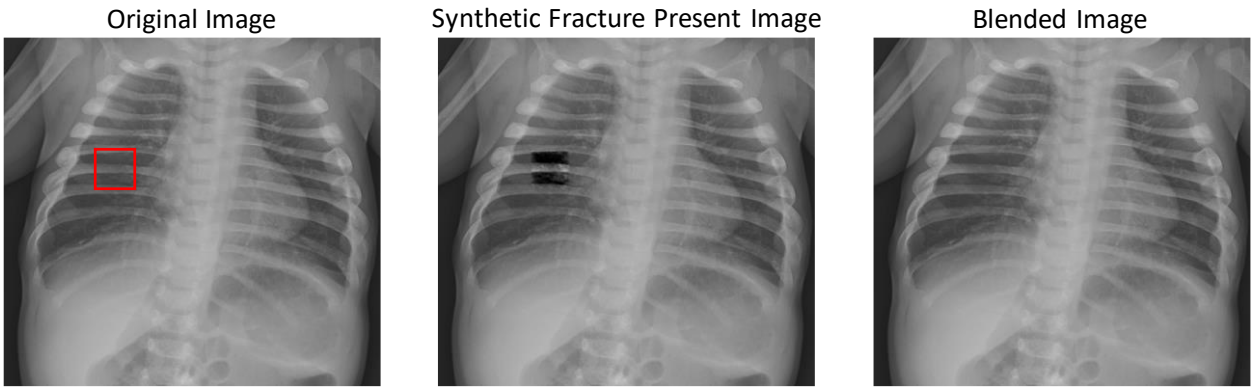

Figure 28. We use generator 1 to synthetically add a fracture to a selected fracture absent patch. The patch is then reinserted back into the original location and then blended using Poisson Inpainting techniques.

In Poisson blending, we try to fill in missing regions of an image. Therefore, we first define a boundary using a mask. Pixels outside of the mask are known, pixels inside the mask are missing. Then, we apply a Laplacian operator to measure the local variations in an image (like edges and texture). The operator propagates image textures into missing regions by solving a boundary constrained optimization problem. The inpainting process often involves an iterative optimization procedure. At each iteration, the estimated pixel values are refined based on the computed Laplacian and the boundary conditions. The process continues until all pixels within the mask are updated.

Once the missing regions are filled in, post-processing techniques can be applied to smooth out any remaining inconsistencies or artifacts. Here, we chose to use a gradient filter that preserves the edge of our original patch and the center of our synthetic patch (Figure 29).

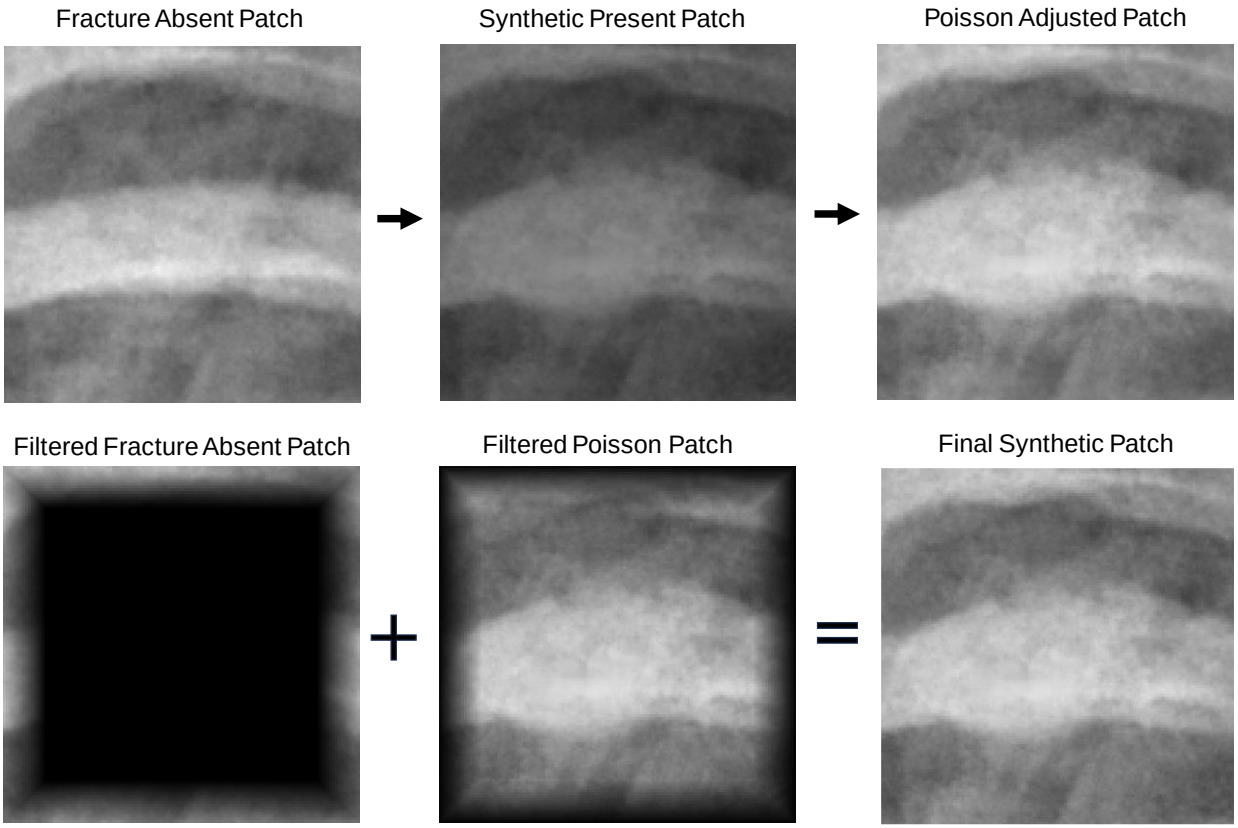

Figure 29. Patch blending process. After a fracture absent patch is translated into a fracture present patch, there remains some contrast issues. The pixel values are adjusted using Poisson inpainting. Then, we use a gradient filter to blend the edges of our patch to match the original surroundings.

3.3 Results and Discussion

Figure 30 shows example synthetic fractures for all three trained models given healthy patches as inputs. Qualitatively, each model can generate convincing apparent fractures with signs of subperiosteal new bone formation, callus bridging, and medullary sclerosis. Our two proposed GANs, NPP and FID-NPP, generally produces sharper images compared to the normal cycleGAN. However, visual inspection suggests that convincing fractures are only generated approximately one-third of the time. Examples of failed generation of fractures are shown in Figure 31, where no new structure seems to have been generated and with negligible changes to visual attenuation properties.

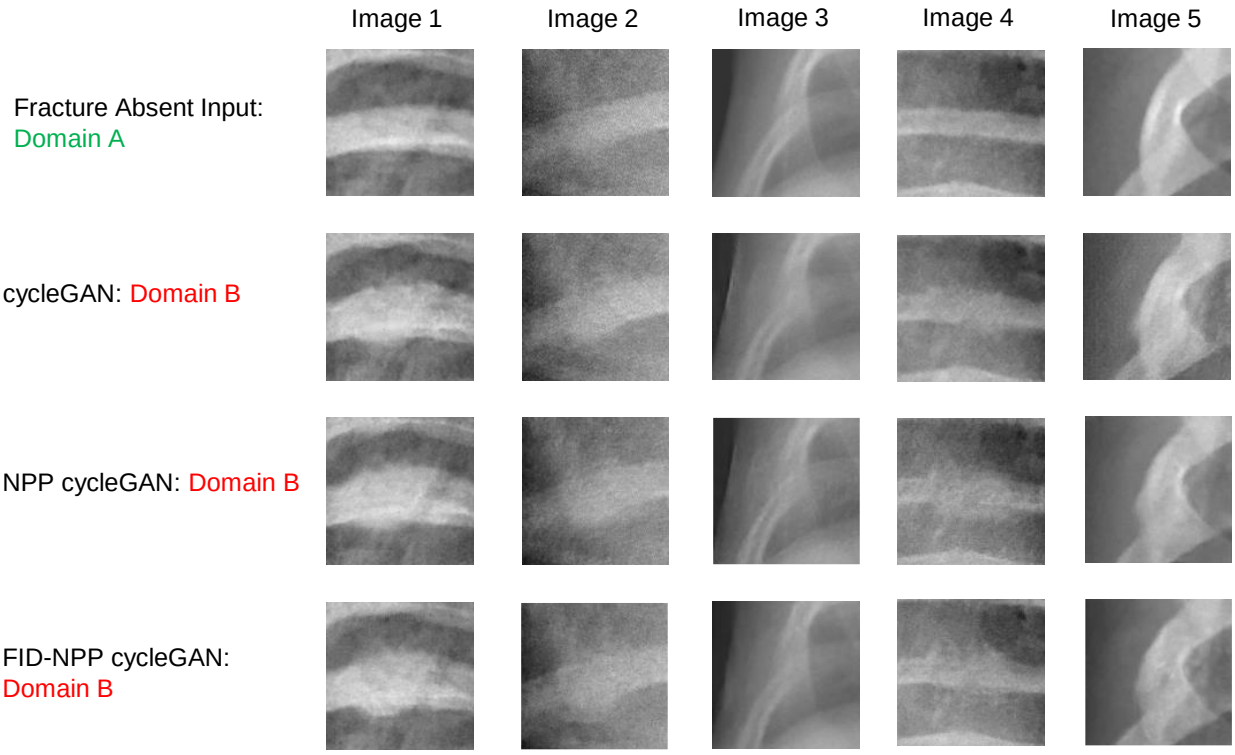

Figure 30. Examples of synthetically generated fractures given fracture absent image patch inputs (Top Row). Each variation of cycleGAN is capable of generating new structure, however, visually the NPP and FID-NPP versions are less blurry and appear more realistic.

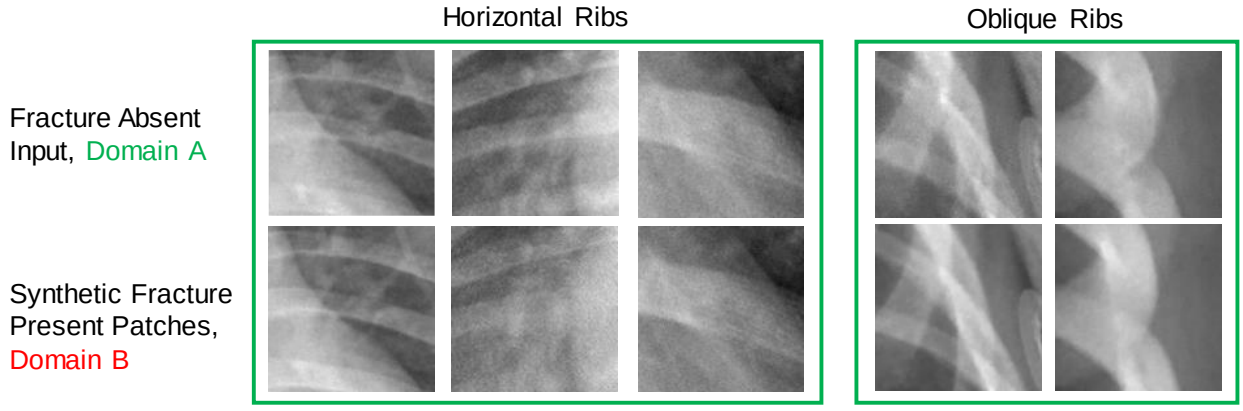

Figure 31. Examples of NPP-GAN synthetically generated fractures (Bottom Row) given fracture absent image patch inputs (Top Row) with little to no evidence of fracture formation.

Table 11 shows the average FID scores for all synthetically generated and real fracture present patches. A lower FID is favorable and indicates that the InceptionV3 network feature vector of the synthetic fracture present patches are closer to the feature vectors from real fracture present patches. Both the proposed NPP and FID-NPP models produces images with lower FID scores than the normal cycleGAN, suggesting they produce more realistic fractures. Furthermore,

the combined FID-NPP method yielded the lowest, best, score. Note that the FID score of a set of patches of real fractures is not zero considering it is being compared to a different set of reference patches of real fractures.

Table 11. Fréchet Inception Distance (FID) score for each cycleGAN variant. A set of 100 randomly selected patches were used to calculate this score. The mean of all FID scores were significantly different from each other at a p-value <0.05.

# of Fracture Absent patches	normal cycleGAN	NPP GAN	FID-NPP	Real Frac Present patches
100	26.3 ± 0.99	24.0 ± 1.08	23.5 ± 1.07	20.6±1.02

The blinded observer study results are displayed in Table 12, which suggests that the synthetic fractures are indeed realistic, and the blending process produces convincing full radiographs. When presented with real fracture absent, real fracture present, and synthetic fracture present images, 4 expert pediatric radiologists scored 2.03 ± 0.84, 4.13 ± 1.23, and 2.73 ± 1.18, respectively. Overall, 10 of 30 synthetic fracture present images were scored at least 3 or higher by 3 radiologists and in 15 images at least one radiologist scored a 4 or higher.

Table 12. Scores from the blinded observer study grading the full radiograph with no fractures, real fractures, with synthetic patch inserted into image. *On average across all readers and images, visual appearance of fracture for synthetic fracture present is significantly higher than fracture absent (p=0.01).

	Real Fracture Absent	Real Fracture Present	Synthetic Fracture Present (FID-NPP)	All Images
Number of Images x Number of Readers	30 x 4	30 x 4	30 x 4	90 x 4
Likelihood of Fracture	2.03 ± 0.84	4.13 ± 1.23	2.73 ± 1.18*	2.79 ± 1.55
Intraclass Correlation Coefficient	0.604	0.781	0.731	0.881

Overall, the evaluation of the FID scores and blinded observer study suggests that our proposed FID-NPP cycleGAN model can generate realistic fractures for many input patches. It particularly excels at creating fractures with calluses of a variety of shapes and sizes. Visually, the generator appears to fulfill our goal of improving the diversity of our object detector training set.

However, we acknowledge that visual inspection suggests that fractures are only generated approximately 1/3rd of the time. This could indicate a failure of our generator, or it could indicate that our generator is creating challenging fractures like in Figure 23C and we are unable to visually discern them. The former statement is more likely and is supported by our blind observer study, where exactly 10/30 synthetic fracture present images were rated at least a 3 by three radiologists.

CHAPTER 4: DATA AUGMENTED NEURAL NETWORK PEDIATRIC RIB FRACTURE DETECTION

The ultimate goal of data augmentation, in general, is to increase a dataset's size and diversity so that a machine learned model that trains on the augmented dataset improves performance. To assess the downstream performance of a rib fracture detector we adapted the YOLOv5 network from Ultralytics [252]. Rather than training an object detector from scratch, we opted to utilize the YOLOv5|6 model for transfer learning with their pretrained weights. Compared to the popular YOLOv3 which operated with a Darknet backbone architecture [253], the YOLOv5 architecture uses a CSPNet backbone based on DenseNet [254]. It additionally integrated a novel mosaic data augmentation method that aimed to improve detection performance of small objects in images.

4.1 Introduction

YOLOv5 is an object detection algorithm from the popular You Only Look Once (YOLO) series of real-time object detection models. YOLOv5 follows a one-stage object detection architecture. It divides an input image into a grid and predicts bounding boxes and class probabilities for objects within each grid cell. It uses a single CNN based off CSPNet to make these predictions. YOLOv5 employs a detection head on top of the backbone network. The detection head consists of additional convolutional layers that process the extracted features and predict bounding box coordinates and class probabilities for each object detected. One interesting innovation of YOLOv5 is the use of anchor boxes. Anchor boxes are predefined bounding boxes of various sizes and aspect ratios, to handle objects of different shapes and sizes. These anchor boxes are associated with each grid cell, and YOLOv5 adjusts them to match the objects' shapes during training.

The YOLO family of networks is constantly evolving, with the current model at v8. We

chose to use YOLOv5 because it is commonly used as a baseline detector in previous studies (See Section 1.4). Overall, YOLOv5 is a powerful object detection algorithm that offers state-of-the-art performance, real-time inference capabilities, and a user-friendly framework for training and deployment.

4.2 Materials and Methods

The real image dataset used for training and evaluation of the YOLOv5 architecture contains 1,109 unique real images, of which 624 are fracture present and 485 are fracture absent. For full demographics, see Section 3.2.

Evaluation Methods and Metrics:

This study aimed to evaluate the performance gains from varying the amount of augmented data in our training set. Specifically, does augmented data provide higher performance gains in low volume scenarios. Table 13 shows the different training conditions. We evaluated and compared training stability, precision, recall, F2, and receiver operating characteristic for each of the conditions. We define these performance metrics as follows:

$$\text{Precision} = \frac{True\ Positive}{Prectricted\ Positive} \qquad \text{Recall} = \frac{True\ Positive}{Condition\ Positive} \qquad F_\beta = (1+\beta)\frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

Table 13. Combinations of real images with GAN generated synthetic fracture images used for training the YOLOv5 object detector.

Training Condition	# Real Images	# Synthetic Fracture Images
1	0	500
2	50	0
3	50	500
4	250	0
5	250	500
6	500	0
7	500	500

While an F1 score is commonly used as an evaluation metric in classification and detection tasks, we opt to use the F2 score for a couple of reasons. Since the goal of this detection task is to

aid radiologists by flagging suspicious regions, we set our β term to weight recall more heavily than precision. We would rather have false positives than false negatives. Therefore, we evaluate all models by F2 score and placing twice as much weight on recall as precision.

Training Details

To avoid bias, any fracture-absent radiographs that were used as the base image during our synthetic radiograph generation were split so that they could only appear in the training set and not the testing set. The remaining real fracture-absent images were then used to generate the testing and validation sets. The test set contains 120 images evenly split with 60 fracture present and 60 fracture absent images.

Each YOLOv5 model was fine-tuned on our data for a maximum of 300 epochs with a batch size of 8 on NVIDIA V100S GPUs. An early stopping protocol was used to end training if performance on the validation set did not improve within 100 epochs. Test set performance was evaluated using weights with the highest validation metric on each respective training set. Error bars for each metric were calculated using a stratified bootstrapping method. In each of the 5,000 iterations, the subsets of 60 fracture present and fracture absent images were randomly sampled with replacement, maintaining 60 images in each set for a total of 120 images to match the original test set size.

4.3 Results and Discussion

Table 14. Performance of object detector trained with different volumes of real and FID-NPP GAN synthetic images. Bold values represent the highest score for each training set size.

Training Dataset	Precision	Recall	F2 Score
0 Real 500 Synthetic	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
50 Real 0 Synthetic	0.764 ± 0.051	**0.327 ± 0.048**	**0.369 ± 0.051**
50 Real 500 Synthetic	**0.981 ± 0.019**	0.201 ± 0.031	0.239 ± 0.036
250 Real 0 Synthetic	0.883 ± 0.031	**0.434 ± 0.042**	**0.482 ± 0.042**
250 Real 500 Synthetic	**0.923 ± 0.027**	0.408 ± 0.049	0.458 ± 0.050
500 Real 0 Synthetic	**0.991 ± 0.009**	0.412 ± 0.041	0.466 ± 0.042
500 Real 500 Synthetic	**0.855 ± 0.034**	**0.488 ± 0.041**	**0.534 ± 0.040**

Tables 14 show the performance of the YOLOv5 object detector trained on different sets of training data augmented with the FID-NPP GAN generated radiographs. Training solely with synthetically generated fractures led to abysmal object detector performance with the detector unable to identify any real fractures. The use of the augmented data with different volumes of real training samples resulted in varying levels of performance gains with increased precision for the low volume conditions (50 Real+500 Synthetic; 250 Real+500 Synthetic) and increased recall and F2 Score (14.6% increase from 0.466 ± 0.042 to 0.534 ± 0.040) for the relatively high-volume condition (500 Real+500 Synthetic). Combined, these results suggest that, in this current application, synthetic data alone is not sufficient for training object detectors and the relative performance gains will be a function of the data augmentation mix (real+synthetic data).

Our modest object detector improvement in performance lines up with similar studies. Section 1.4 showed that generally downstream NN only has a 1-4% increase in performance metric. In best case scenarios, Hammami et al. showed an 8% increase in mean average precision (mAP) for multi-organ detection when adding cycleGAN augmented CT images [235]. Kanayama et al. showed a 6% increase in mAP for gastric cancer detection using a conditional GAN [255], and a 10% increase in sensitivity was seen in brain metastases detection by adding PGGAN generated MR images by Han et al. [256]. The addition of 500 synthetic images to 500 real images using our best performing GAN shows an 18.5% increase in recall and a 14.6% increase in F2 score, which is a promising indication that our NPP-FID method is better than no data augmentation.

The variable behavior in object detector performance relative to synthetic images in the overall training data is also seen with other studies. The three mentioned previously also saw that either adding too little or too many synthetic images produce worse results than only real images.

Additionally, an increase in synthetic image realism does not necessarily improve a detection performance score [256]. The reason behind this behavior is not definitively clear, however the leading theory is that learning from data produced by other models causes model collapse – a degenerative process whereby, over time, models forget the true underlying data distribution, even in the absence of a shift in the distribution over time [257]. Therefore, further studies need to be made to determine the best ratio of synthetic to real images for object detector training, as well as methods to minimize model collapse.

In summary, we proposed a new technique that utilizes near-pair image patches along with an FID loss function to train a cycleGAN model. Our results show that this approach can generate realistic-looking pediatric chest radiographs containing rib fractures. The augmented fracture data led to a moderate improvement in the performance of a rib fracture detection model. This technique could potentially be generalizable to other medical imaging tasks where the goal is to synthesize realistic pathology.

4.4 Perspective on the Future of Image Generation

The field of generative AI is a fast-evolving field. GANs are less than 10 years old and new improvements are still being made. It has been used for a wide range of medical imaging applications, indicating that the technology is versatile and adaptable. In the future, the proposed FID-NPP cycleGAN can potentially be generalized to any problem that seeks to localize pathology. The pipeline for this generalization is as simple as generating near pairs from a target dataset and using the images to train the FID-NPP network (available here: https://github.com/tuethan/Pediatric-Chest-Radiograph-Data-Augmentation). Similarly, we can then generate synthetic images using the same process as in section 3.2.5 and then train an object detector using the augmented data. There are two general improvements that can be made to the proposed method; improving fracture labels and changing the generator architecture. The size of

the patches was chosen based on the median size of our labeled bounding boxes. Large bounding boxes tend to contain multiple fractures, since physicians simply highlight an area if there are a cluster of fractures. Therefore, some fracture-present patches may contain off-centered fractures or partial fractures, which hinders our cycleGAN training. Additionally, the object detector assumes that each bounding box has one fracture present, so training with a few samples that contain multiple fracture bounding boxes does not benefit the model. By relabeling our images and standardizing the size of our bounding boxes, we may achieve better results.

The second major improvement is changing the design of our generator network. This change can be as simple as tuning hyper parameters (number of filters, filter size, convolution padding style, upsampling method, etc.), or as complex as changing the architecture (adding residual blocks, adding batch normalization layers, adding an auxiliary network, etc.). We may even choose to replace GANs altogether with a new type of image generation network which has recently dominated the field. Diffusion models such as DALL-E [258], Midjourney [259], and Stable Diffusion [260] have gained notable attention due to their remarkably high-resolution outputs from a prompt in natural language (Figure 32). These models are trained using hundreds of millions of images and have the flexibility of generating diverse content. The underlying network is fundamentally different from GANs. Rather than a generator learning based off a discriminator's ability to classify real from fake, diffusion models learn to map noise to an image in a progressive manner. Despite numerous copyright lawsuits claiming misuse of artists' images, diffusion models are still advancing (Midjourney v5.2 just released in June 2023). This all begs the question; if diffusion models can generate images that are as high quality as GANs and can generate more diverse content than GANs, do we even need GANs anymore?

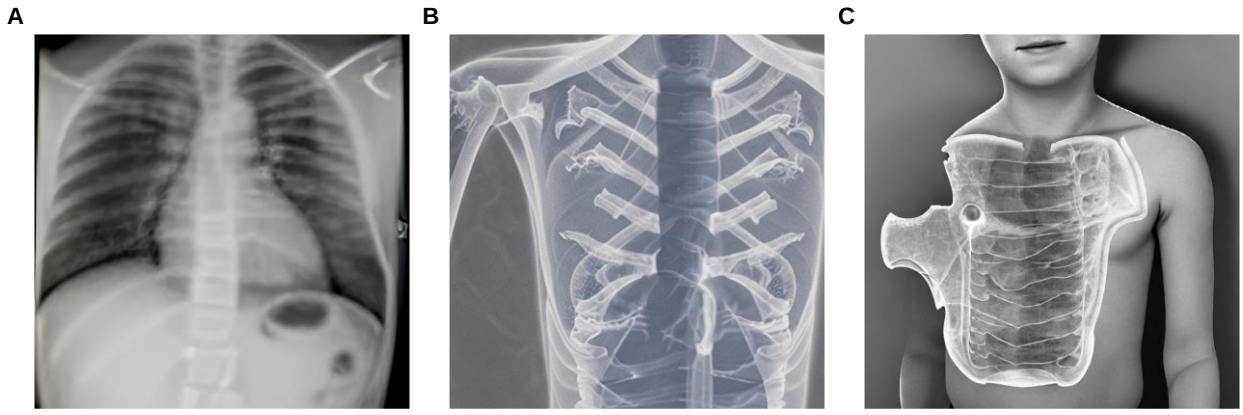

Figure 32. Images generated using DALL-E (A), Midjourney (B), and Stable Diffusion (C) with the text prompt "pediatric chest radiograph with rib fractures".

I believe that GANs be obsolete, soon. Currently, there are two main constraints that limit the use of diffusion models for medical imaging research. The first is computational resources and second is data. Diffusion models are notoriously known to be computationally expensive and slow to train while requiring huge volumes of data [261]. GANs are still valuable in research settings simply because they are cheaper and more efficient. They still have immense potential in the medical imaging field where we have highly specialized tasks. However, this may soon change as a prospective study has produced a model combining diffusion models and GANs [262], and a few prospective studies have shown that diffusion models are superior to GANs for specialized image generation [263], [264], including medical imaging tasks [265], [266].

In conclusion, GANs and Diffusion Models represent two branches of generative AI stemming from a large tree of neural networks. This field is ever growing and always exciting, it will be interesting to see what the new buds will bring.

4.5 Supplemental Information

The following Tables A1 and A2 are the object detector performance results using normal cycleGAN and NPP cycleGAN augmented data. These tables are not included in Section 4.3 because they are not a fair comparison to each other. The set of real radiographs used for Table 14 is different from Table A1 and A2, and therefore the baseline real-only evaluations do not match. Additionally, the base radiographs as well as selected patches used for synthetic image generation do not match from set to set. For future studies we would like to use matching images for both real and synthetic portions to offer a fair comparison between different models.

Table 15. Performance of object detector trained with different volumes of real and normal cycleGAN synthetic images. Bold values represent the highest score for each training set size.

Training Dataset	Precision	Recall	F2 Score
0 Real 500 Synthetic	0.125 ± 0.128	0.004 ± 0.004	0.005 ± 0.005
50 Real 0 Synthetic	0.735 ± 0.070	**0.214 ± 0.043**	**0.249 ± 0.048**
50 Real 500 Synthetic	**0.815 ± 0.091**	0.169 ± 0.044	0.200 ± 0.050
250 Real 0 Synthetic	**0.820 ± 0.035**	0.452 ± 0.048	0.496 ± 0.047
250 Real 500 Synthetic	0.816 ± 0.039	**0.456 ± 0.053**	**0.499 ± 0.052**
500 Real 0 Synthetic	**0.901 ± 0.030**	0.448 ± 0.051	0.498 ± 0.051
500 Real 500 Synthetic	0.860 ± 0.033	**0.493 ± 0.048**	**0.539 ± 0.047**

Table 16. Performance of object detector trained with different volumes of real and FID-NPP GAN synthetic images. Bold values represent the highest score for each training set size.

Training Dataset	Precision	Recall	F2 Score
0 Real 500 Synthetic	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
50 Real 0 Synthetic	0.779 ± 0.059	**0.254 ± 0.035**	0.**294** ± 0.038
50 Real 500 Synthetic	**0.922 ± 0.046**	0.179 ± 0.035	0.213 ± 0.040
250 Real 0 Synthetic	0.894 ± 0.047	**0.468 ± 0.048**	**0.517 ± 0.048**
250 Real 500 Synthetic	**0.909 ± 0.033**	0.449 ± 0.050	0.499 ± 0.050
500 Real 0 Synthetic	0.825 ± 0.035	**0.536 ± 0.044**	**0.576 ± 0.042**
500 Real 500 Synthetic	**0.874 ± 0.036**	0.524 ± 0.047	0.569 ± 0.046

REFERENCES

[1] E. Tu, M. Azmat, K. Branch, and A. Alessio, "Machine learned approach for estimating myocardial blood flow from dynamic CT and coronary artery disease risk factors," in *Medical Imaging 2021: Biomedical Applications in Molecular, Structural, and Functional Imaging*, SPIE, Feb. 2021, pp. 370–375. doi: 10.1117/12.2581703.

[2] K. M. Pawelec *et al.*, "Incorporating Tantalum Oxide Nanoparticles into Implantable Polymeric Biomedical Devices for Radiological Monitoring," *Adv. Healthc. Mater.*, vol. 12, no. 18, p. 2203167, 2023, doi: 10.1002/adhm.202203167.

[3] R. Nakazato, D. S. Berman, E. Alexanderson, and P. Slomka, "Myocardial perfusion imaging with PET," *Imaging Med.*, vol. 5, no. 1, pp. 35–46, Feb. 2013, doi: 10.2217/iim.13.1.

[4] R. S. Driessen, P. G. Raijmakers, W. J. Stuijfzand, and P. Knaapen, "Myocardial perfusion imaging with PET," *Int. J. Cardiovasc. Imaging*, vol. 33, no. 7, pp. 1021–1031, 2017, doi: 10.1007/s10554-017-1084-4.

[5] R. Sciagrà *et al.*, "Clinical use of quantitative cardiac perfusion PET: rationale, modalities and possible indications. Position paper of the Cardiovascular Committee of the European Association of Nuclear Medicine (EANM)," *Eur. J. Nucl. Med. Mol. Imaging*, vol. 43, no. 8, pp. 1530–1545, Jul. 2016, doi: 10.1007/s00259-016-3317-5.

[6] O. R. Coelho-Filho, C. Rickers, R. Y. Kwong, and M. Jerosch-Herold, "MR myocardial perfusion imaging," *Radiology*, vol. 266, no. 3, pp. 701–715, Mar. 2013, doi: 10.1148/radiol.12110918.

[7] F. Bamberg *et al.*, "Dynamic myocardial CT perfusion imaging for evaluation of myocardial ischemia as determined by MR imaging," *JACC Cardiovasc. Imaging*, vol. 7, no. 3, pp. 267–277, Mar. 2014, doi: 10.1016/j.jcmg.2013.06.008.

[8] M. Jerosch-Herold, "Quantification of myocardial perfusion by cardiovascular magnetic resonance," *J. Cardiovasc. Magn. Reson.*, vol. 12, no. 1, p. 57, Oct. 2010, doi: 10.1186/1532-429X-12-57.

[9] R. Y. Kwong *et al.*, "Cardiac Magnetic Resonance Stress Perfusion Imaging for Evaluation of Patients With Chest Pain," *J. Am. Coll. Cardiol.*, vol. 74, no. 14, pp. 1741–1755, Oct. 2019, doi: 10.1016/j.jacc.2019.07.074.

[10] M. Bindschadler, D. Modgil, K. R. Branch, P. J. La Riviere, and A. M. Alessio, "Comparison of blood flow models and acquisitions for quantitative myocardial perfusion estimation from dynamic CT," *Phys. Med. Biol.*, vol. 59, no. 7, pp. 1533–1556, Apr. 2014, doi: 10.1088/0031-9155/59/7/1533.

[11] M. Bindschadler, K. R. Branch, and A. M. Alessio, "Quantitative myocardial perfusion from static cardiac and dynamic arterial CT," *Phys. Med. Biol.*, vol. 63, no. 10, p. 105020, May 2018, doi: 10.1088/1361-6560/aac0bd.

[12] A. Varga-Szemes, F. G. Meinel, C. N. De Cecco, S. R. Fuller, R. R. Bayer, and U. J. Schoepf, "CT myocardial perfusion imaging," *AJR Am. J. Roentgenol.*, vol. 204, no. 3, pp. 487–497, Mar. 2015, doi: 10.2214/AJR.14.13546.

[13] C. Q. Davidson, C. P. Phenix, T. Tai, N. Khaper, and S. J. Lees, "Searching for novel PET radiotracers: imaging cardiac perfusion, metabolism and inflammation," *Am. J. Nucl. Med. Mol. Imaging*, vol. 8, no. 3, pp. 200–227, Jun. 2018.

[14] T. H. Schindler, H. R. Schelbert, A. Quercioli, and V. Dilsizian, "Cardiac PET Imaging for the Detection and Monitoring of Coronary Artery Disease and Microvascular Health," *JACC Cardiovasc. Imaging*, vol. 3, no. 6, pp. 623–640, Jun. 2010, doi: 10.1016/j.jcmg.2010.04.007.

[15] K. L. Gould *et al.*, "Anatomic versus physiologic assessment of coronary artery disease. Role of coronary flow reserve, fractional flow reserve, and positron emission tomography imaging in revascularization decision-making," *J. Am. Coll. Cardiol.*, vol. 62, no. 18, pp. 1639–1653, Oct. 2013, doi: 10.1016/j.jacc.2013.07.076.

[16] R. A. P. Takx *et al.*, "Diagnostic accuracy of stress myocardial perfusion imaging compared to invasive coronary angiography with fractional flow reserve meta-analysis," *Circ. Cardiovasc. Imaging*, vol. 8, no. 1, p. e002666, Jan. 2015, doi: 10.1161/CIRCIMAGING.114.002666.

[17] C. Jaarsma *et al.*, "Diagnostic performance of noninvasive myocardial perfusion imaging using single-photon emission computed tomography, cardiac magnetic resonance, and positron emission tomography imaging for the detection of obstructive coronary artery disease: a meta-analysis," *J. Am. Coll. Cardiol.*, vol. 59, no. 19, pp. 1719–1728, May 2012, doi: 10.1016/j.jacc.2011.12.040.

[18] B. A. Mc Ardle, T. F. Dowsley, R. A. deKemp, G. A. Wells, and R. S. Beanlands, "Does rubidium-82 PET have superior accuracy to SPECT perfusion imaging for the diagnosis of obstructive coronary disease?: A systematic review and meta-analysis," *J. Am. Coll. Cardiol.*, vol. 60, no. 18, pp. 1828–1837, Oct. 2012, doi: 10.1016/j.jacc.2012.07.038.

[19] C. B. Monti, M. Codari, M. van Assen, C. N. De Cecco, and R. Vliegenthart, "Machine Learning and Deep Neural Networks Applications in Computed Tomography for Coronary Artery Disease and Myocardial Perfusion," *J. Thorac. Imaging*, vol. 35 Suppl 1, pp. S58–S65, May 2020, doi: 10.1097/RTI.0000000000000490.

[20] R. Arsanjani *et al.*, "Prediction of revascularization after myocardial perfusion SPECT by machine learning in a large population," *J. Nucl. Cardiol. Off. Publ. Am. Soc. Nucl. Cardiol.*, vol. 22, no. 5, pp. 877–884, Oct. 2015, doi: 10.1007/s12350-014-0027-x.

[21] Y. Li *et al.*, "Detection of Hemodynamically Significant Coronary Stenosis: CT Myocardial Perfusion versus Machine Learning CT Fractional Flow Reserve," *Radiology*, vol. 293, no. 2, pp. 305–314, Nov. 2019, doi: 10.1148/radiol.2019190098.

[22] J. Betancur *et al.*, "Prognostic Value of Combined Clinical and Myocardial Perfusion Imaging Data Using Machine Learning," *JACC Cardiovasc. Imaging*, vol. 11, no. 7, pp. 1000–1009, Jul. 2018, doi: 10.1016/j.jcmg.2017.07.024.

[23] J. M. Wolterink, T. Leiner, R. A. P. Takx, M. A. Viergever, and I. Isgum, "Automatic Coronary Calcium Scoring in Non-Contrast-Enhanced ECG-Triggered Cardiac CT With Ambiguity Detection," *IEEE Trans. Med. Imaging*, vol. 34, no. 9, pp. 1867–1878, Sep. 2015, doi: 10.1109/TMI.2015.2412651.

[24] A. M. Alessio, M. Bindschadler, J. M. Busey, W. P. Shuman, J. H. Caldwell, and K. R. Branch, "Accuracy of Myocardial Blood Flow Estimation From Dynamic Contrast-Enhanced Cardiac CT Compared With PET," *Circ. Cardiovasc. Imaging*, vol. 12, no. 6, p. e008323, Jun. 2019, doi: 10.1161/CIRCIMAGING.118.008323.

[25] M. D. Cerqueira *et al.*, "Standardized Myocardial Segmentation and Nomenclature for Tomographic Imaging of the Heart," *Circulation*, vol. 105, no. 4, pp. 539–542, Jan. 2002, doi: 10.1161/hc0402.102975.

[26] L. J. Shaw *et al.*, "Impact of ethnicity and gender differences on angiographic coronary artery disease prevalence and in-hospital mortality in the American College of Cardiology-National Cardiovascular Data Registry," *Circulation*, vol. 117, no. 14, pp. 1787–1801, Apr. 2008, doi: 10.1161/CIRCULATIONAHA.107.726562.

[27] F. Cademartiri *et al.*, "Myocardial blood flow quantification for evaluation of coronary artery disease by computed tomography," *Cardiovasc. Diagn. Ther.*, vol. 7, no. 2, pp. 129–150, Apr. 2017, doi: 10.21037/cdt.2017.03.22.

[28] W. B. Meijboom *et al.*, "Comprehensive assessment of coronary artery stenoses: computed tomography coronary angiography versus conventional coronary angiography and correlation with fractional flow reserve in patients with stable angina," *J. Am. Coll. Cardiol.*, vol. 52, no. 8, pp. 636–643, Aug. 2008, doi: 10.1016/j.jacc.2008.05.024.

[29] C. W. White *et al.*, "Does visual interpretation of the coronary arteriogram predict the physiologic importance of a coronary stenosis?," *N. Engl. J. Med.*, vol. 310, no. 13, pp. 819–824, Mar. 1984, doi: 10.1056/NEJM198403293101304.

[30] C. A. Santana *et al.*, "Diagnostic performance of fusion of myocardial perfusion imaging (MPI) and computed tomography coronary angiography," *J. Nucl. Cardiol. Off. Publ. Am. Soc. Nucl. Cardiol.*, vol. 16, no. 2, pp. 201–211, 2009, doi: 10.1007/s12350-008-9019-z.

[31] G. Pontone *et al.*, "Rationale and design of the PERFECTION (comparison between stress cardiac computed tomography PERfusion versus Fractional flow rEserve measured by Computed Tomography angiography In the evaluation of suspected cOroNary artery disease) prospective study," *J. Cardiovasc. Comput. Tomogr.*, vol. 10, no. 4, pp. 330–334, 2016, doi: 10.1016/j.jcct.2016.03.004.

[32] A. Bol *et al.*, “Direct comparison of [13N]ammonia and [15O]water estimates of perfusion with quantification of regional myocardial blood flow by microspheres,” *Circulation*, vol. 87, no. 2, pp. 512–525, Feb. 1993, doi: 10.1161/01.cir.87.2.512.

[33] S. R. Bergmann *et al.*, “Quantification of regional myocardial blood flow in vivo with H215O,” *Circulation*, vol. 70, no. 4, pp. 724–733, Oct. 1984, doi: 10.1161/01.cir.70.4.724.

[34] T. Kero, J. Nordström, H. J. Harms, J. Sörensen, H. Ahlström, and M. Lubberink, “Quantitative myocardial blood flow imaging with integrated time-of-flight PET-MR,” *EJNMMI Phys.*, vol. 4, p. 1, Jan. 2017, doi: 10.1186/s40658-016-0171-2.

[35] J. M. Renaud *et al.*, “Characterization of 3-Dimensional PET Systems for Accurate Quantification of Myocardial Blood Flow,” *J. Nucl. Med. Off. Publ. Soc. Nucl. Med.*, vol. 58, no. 1, pp. 103–109, Jan. 2017, doi: 10.2967/jnumed.116.174565.

[36] H. Schelbert, “Measurement of MBF by PET is ready for prime time as an integral part of clinical reports in diagnosis and risk assessment of patients with known or suspected CAD,” *J. Nucl. Cardiol.*, vol. 25, no. 1, pp. 153–156, Feb. 2018, doi: 10.1007/s12350-016-0423-5.

[37] S. V. Nesterov *et al.*, “Quantification of Myocardial Blood Flow in Absolute Terms u27sing 82Rb PET Imaging: Results of RUBY-10—a multicenter study comparing ten computer analysis programs,” *JACC Cardiovasc. Imaging*, vol. 7, no. 11, pp. 1119–1127, Nov. 2014, doi: 10.1016/j.jcmg.2014.08.003.

[38] M. S. Ambrose, C. Valdiviezo, V. Mehra, A. C. Lardo, J. A. C. Lima, and R. T. George, “CT perfusion: ready for prime time,” *Curr. Cardiol. Rep.*, vol. 13, no. 1, pp. 57–66, Feb. 2011, doi: 10.1007/s11886-010-0152-3.

[39] A. So *et al.*, “Non-invasive assessment of functionally relevant coronary artery stenoses with quantitative CT perfusion: preliminary clinical experiences,” *Eur. Radiol.*, vol. 22, no. 1, pp. 39–50, Jan. 2012, doi: 10.1007/s00330-011-2260-x.

[40] A. So, J. Hsieh, J.-Y. Li, J. Hadway, H.-F. Kong, and T.-Y. Lee, “Quantitative myocardial perfusion measurement using CT perfusion: a validation study in a porcine model of reperfused acute myocardial infarction,” *Int. J. Cardiovasc. Imaging*, vol. 28, no. 5, pp. 1237–1248, Jun. 2012, doi: 10.1007/s10554-011-9927-x.

[41] C. A. Cuenod and D. Balvay, “Perfusion and vascular permeability: basic concepts and measurement in DCE-CT and DCE-MRI,” *Diagn. Interv. Imaging*, vol. 94, no. 12, pp. 1187–1204, Dec. 2013, doi: 10.1016/j.diii.2013.10.010.

[42] G. Pontone *et al.*, “Diagnostic performance of non-invasive imaging for stable coronary artery disease: A meta-analysis,” *Int. J. Cardiol.*, vol. 300, pp. 276–281, Feb. 2020, doi: 10.1016/j.ijcard.2019.10.046.

[43] M. Lu, S. Wang, A. Sirajuddin, A. E. Arai, and S. Zhao, “Dynamic stress computed tomography myocardial perfusion for detecting myocardial ischemia: A systematic review

and meta-analysis," *Int. J. Cardiol.*, vol. 258, pp. 325–331, May 2018, doi: 10.1016/j.ijcard.2018.01.095.

[44] F. M. A. Nous *et al.*, "Dynamic Myocardial Perfusion CT for the Detection of Hemodynamically Significant Coronary Artery Disease," *JACC Cardiovasc. Imaging*, vol. 15, no. 1, pp. 75–87, Jan. 2022, doi: 10.1016/j.jcmg.2021.07.021.

[45] N. P. Johnson and K. L. Gould, "Integrating noninvasive absolute flow, coronary flow reserve, and ischemic thresholds into a comprehensive map of physiological severity," *JACC Cardiovasc. Imaging*, vol. 5, no. 4, pp. 430–440, Apr. 2012, doi: 10.1016/j.jcmg.2011.12.014.

[46] C. R. R. N. Hunter, R. Klein, R. S. Beanlands, and R. A. deKemp, "Patient motion effects on the quantification of regional myocardial blood flow with dynamic PET imaging," *Med. Phys.*, vol. 43, no. 4, pp. 1829–1840, 2016, doi: 10.1118/1.4943565.

[47] Y. Xia, Y. He, F. Zhang, Y. Liu, and J. Leng, "A Review of Shape Memory Polymers and Composites: Mechanisms, Materials, and Applications," *Adv. Mater.*, vol. 33, no. 6, p. 2000713, 2021, doi: 10.1002/adma.202000713.

[48] M. Veletić *et al.*, "Implants with Sensing Capabilities," *Chem. Rev.*, vol. 122, no. 21, pp. 16329–16363, Nov. 2022, doi: 10.1021/acs.chemrev.2c00005.

[49] Y.-H. Shao, K. Tsai, S. Kim, Y.-J. Wu, and K. Demissie, "Exposure to Tomographic Scans and Cancer Risks," *JNCI Cancer Spectr.*, vol. 4, no. 1, p. pkz072, Feb. 2020, doi: 10.1093/jncics/pkz072.

[50] K. M. Pawelec *et al.*, "Design Considerations to Facilitate Clinical Radiological Evaluation of Implantable Biomedical Structures," *ACS Biomater. Sci. Eng.*, vol. 7, no. 2, pp. 718–726, Feb. 2021, doi: 10.1021/acsbiomaterials.0c01439.

[51] J. Wallyn, N. Anton, S. Akram, and T. F. Vandamme, "Biomedical Imaging: Principles, Technologies, Clinical Aspects, Contrast Agents, Limitations and Future Trends in Nanomedicines," *Pharm. Res.*, vol. 36, no. 6, p. 78, Apr. 2019, doi: 10.1007/s11095-019-2608-5.

[52] D. A. Szulc and H.-L. M. Cheng, "One-Step Labeling of Collagen Hydrogels with Polydopamine and Manganese Porphyrin for Non-Invasive Scaffold Tracking on Magnetic Resonance Imaging," *Macromol. Biosci.*, vol. 19, no. 4, p. e1800330, Apr. 2019, doi: 10.1002/mabi.201800330.

[53] A. Erol, D. B. H. Rosberg, B. Hazer, and B. S. Göncü, "Biodegradable and biocompatible radiopaque iodinated poly-3-hydroxy butyrate: synthesis, characterization and in vitro/in vivo X-ray visibility," *Polym. Bull.*, vol. 77, no. 1, pp. 275–289, Jan. 2020, doi: 10.1007/s00289-019-02747-6.

[54] J. M. Crowder, N. Bates, J. Roberts, A. S. Torres, and P. J. Bonitatibus, "Determination of tantalum from tantalum oxide nanoparticle X-ray/CT contrast agents in rat tissues and bodily

fluids by ICP-OES," *J. Anal. At. Spectrom.*, vol. 31, no. 6, pp. 1311–1317, 2016, doi: 10.1039/C5JA00446B.

[55] J. W. Lambert *et al.*, "An Intravascular Tantalum Oxide-based CT Contrast Agent: Preclinical Evaluation Emulating Overweight and Obese Patient Size," *Radiology*, vol. 289, no. 1, pp. 103–110, Oct. 2018, doi: 10.1148/radiol.2018172381.

[56] S. Chakravarty *et al.*, "Tantalum oxide nanoparticles as versatile contrast agents for X-ray computed tomography," *Nanoscale*, vol. 12, no. 14, pp. 7720–7734, Apr. 2020, doi: 10.1039/d0nr01234c.

[57] G. Mohandas, N. Oskolkov, M. T. McMahon, P. Walczak, and M. Janowski, "Porous tantalum and tantalum oxide nanoparticles for regenerative medicine," *Acta Neurobiol. Exp. (Warsz.)*, vol. 74, no. 2, pp. 188–196, 2014.

[58] A. Göpferich, "Mechanisms of polymer degradation and erosion," *Biomaterials*, vol. 17, no. 2, pp. 103–114, Jan. 1996, doi: 10.1016/0142-9612(96)85755-3.

[59] S. M. Pizer *et al.*, "Adaptive histogram equalization and its variations," *Comput. Vis. Graph. Image Process.*, vol. 39, no. 3, pp. 355–368, Sep. 1987, doi: 10.1016/S0734-189X(87)80186-X.

[60] N. Otsu, "A Tlreshold Selection Method from Gray-Level Histograms".

[61] D. Bradley and G. Roth, "Adaptive Thresholding using the Integral Image," *J. Graph. Tools*, vol. 12, no. 2, pp. 13–21, Jan. 2007, doi: 10.1080/2151237X.2007.10129236.

[62] L. Lu *et al.*, "In vitro and in vivo degradation of porous poly(DL-lactic-co-glycolic acid) foams," *Biomaterials*, vol. 21, no. 18, pp. 1837–1845, Sep. 2000, doi: 10.1016/s0142-9612(00)00047-8.

[63] C. E. Rapier, K. J. Shea, and A. P. Lee, "Investigating PLGA microparticle swelling behavior reveals an interplay of expansive intermolecular forces," *Sci. Rep.*, vol. 11, no. 1, p. 14512, Jul. 2021, doi: 10.1038/s41598-021-93785-6.

[64] W. McCulloch and W. Pitts, "A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY," *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[65] D. Hebb, *The Organization of Behavior*. John Wiley & Sons Inc, 1949.

[66] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, pp. 386–408, 1958, doi: 10.1037/h0042519.

[67] P. S. Churchland, "How Do Neurons Know?," *Daedalus*, vol. 133, no. 1, pp. 42–50, 2004.

[68] "How Neural Networks Learn from Experience," *Scientific American*. https://www.scientificamerican.com/article/how-neural-networks-learn-from-expe/ (accessed Apr. 07, 2023).

[69] H. White, “Learning in Artificial Neural Networks: A Statistical Perspective,” *Neural Comput.*, vol. 1, no. 4, pp. 425–464, Dec. 1989, doi: 10.1162/neco.1989.1.4.425.

[70] S. Sharma, S. Sharma, and A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 04, no. 12, pp. 310–316, May 2020, doi: 10.33564/IJEAST.2020.v04i12.054.

[71] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models”.

[72] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines”.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Apr. 07, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[74] R. Pramoditha, “How to Choose the Right Activation Function for Neural Networks,” *Medium*, Jan. 26, 2022. https://towardsdatascience.com/how-to-choose-the-right-activation-function-for-neural-networks-3941ff0e6f9c (accessed Apr. 07, 2023).

[75] D. R. Cox, “The Regression Analysis of Binary Sequences,” *J. R. Stat. Soc. Ser. B Methodol.*, vol. 20, no. 2, pp. 215–242, 1958.

[76] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A Tutorial on the Cross-Entropy Method,” *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, Feb. 2005, doi: 10.1007/s10479-005-5724-z.

[77] R. Y. Rubinstein and D. P. Kroese, *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2004.

[78] P. J. Huber, “Robust Estimation of a Location Parameter,” *Ann. Math. Stat.*, vol. 35, no. 1, pp. 73–101, Mar. 1964, doi: 10.1214/aoms/1177703732.

[79] P. Werbos, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph. D.). Appl. Math. Harvard University,” 1974.

[80] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, Art. no. 6088, Oct. 1986, doi: 10.1038/323533a0.

[81] “Learning Internal Representations by Error Propagation.” Accessed: Apr. 07, 2023. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA164453

[82] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990, doi: 10.1109/5.58337.

[83] Kohonen, Barna, and Chrisley, "Statistical pattern recognition with neural networks: benchmarking studies," in *IEEE 1988 International Conference on Neural Networks*, Jul. 1988, pp. 61–68 vol.1. doi: 10.1109/ICNN.1988.23829.

[84] F. F. Li, J. Johnson, and S. Yeung, "Detection and Segmentation," May 10, 2017. [Online]. Available: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

[85] A. P. Dhawan, "A review on biomedical image processing and future trends," *Comput. Methods Programs Biomed.*, vol. 31, no. 3, pp. 141–183, Mar. 1990, doi: 10.1016/0169-2607(90)90001-P.

[86] H. Wechsler and J. Sklansky, "Finding the rib cage in chest radiographs," *Pattern Recognit.*, vol. 9, no. 1, pp. 21–30, Jan. 1977, doi: 10.1016/0031-3203(77)90027-9.

[87] Ballard and Sklansky, "A Ladder-Structured Decision Tree for Recognizing Tumors in Chest Radiographs," *IEEE Trans. Comput.*, vol. C–25, no. 5, pp. 503–513, May 1976, doi: 10.1109/TC.1976.1674638.

[88] Y. P. CHIEN, "Preprocessing and Feature Extraction of Picture Patterns.," Ph.D., Purdue University, United States -- Indiana. Accessed: Apr. 09, 2023. [Online]. Available: https://www.proquest.com/docview/302740388/citation/33A85B01E15B49FCPQ/1

[89] M. Cocklin, A. Gourlay, P. Jackson, G. Kaye, I. Kerr, and P. Lams, "Digital processing of chest radiographs," *Image Vis. Comput.*, vol. 1, no. 2, pp. 67–78, May 1983, doi: 10.1016/0262-8856(83)90044-6.

[90] K. Preston, M. J. B. Duff, S. Levialdi, P. E. Norgren, and J. Toriwaki, "Basics of cellular logic with some applications in medical image processing," *Proc. IEEE*, vol. 67, no. 5, pp. 826–856, May 1979, doi: 10.1109/PROC.1979.11331.

[91] R. H. Sherrier and G. A. Johnson, "Regionally adaptive histogram equalization of the chest," *IEEE Trans. Med. Imaging*, vol. 6, no. 1, pp. 1–7, 1987, doi: 10.1109/TMI.1987.4307791.

[92] H. P. McAdams, G. A. Johnson, S. A. Suddarth, and C. E. Ravin, "Histogram-directed processing of digital chest images," *Invest. Radiol.*, vol. 21, no. 3, pp. 253–259, Mar. 1986, doi: 10.1097/00004424-198603000-00011.

[93] G. Davis and S. T. Wallenslager, "Improvement of Chest Region CT Images through Automated Gray-Level Remapping," *IEEE Trans. Med. Imaging*, vol. 5, no. 1, pp. 30–34, 1986, doi: 10.1109/TMI.1986.4307736.

[94] S. M. Pizer, J. B. Zimmerman, and E. V. Staab, "Adaptive grey level assignment in CT scan display," *J. Comput. Assist. Tomogr.*, vol. 8, no. 2, pp. 300–305, Apr. 1984.

[95] R. W. Conners and C. A. Harlow, "Toward a structural textural analyzer based on statistical methods," *Comput. Graph. Image Process.*, vol. 12, no. 3, pp. 224–256, Mar. 1980, doi: 10.1016/0146-664X(80)90013-1.

[96] R. S. Ledley, H. K. Huang, and L. S. Rotolo, “A texture analysis method in classification of coal workers’ pneumoconiosis,” *Comput. Biol. Med.*, vol. 5, no. 1, pp. 53–67, Jun. 1975, doi: 10.1016/0010-4825(75)90018-9.

[97] H. WECHSLER, “Automatic Detection of Rib Contours in Chest Radiographs.,” Ph.D., University of California, Irvine, United States -- California. Accessed: Apr. 09, 2023. [Online]. Available: https://www.proquest.com/docview/302757560/citation/B837D40E4C3B452FPQ/1

[98] E. Hall and R. Turner, “Automated Measurements from Chest X-Rays for Lung Disease Classification,” Aug. 1975.

[99] K. Fukushima, “Neocognitron: A hierarchical neural network capable of visual pattern recognition,” *Neural Netw.*, vol. 1, no. 2, pp. 119–130, Jan. 1988, doi: 10.1016/0893-6080(88)90014-7.

[100] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, Art. no. 4, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[101] M. Ahmad, J. Khan, A. Yousaf, S. Ghuffar, and K. Khurshid, “Deep Learning: A Breakthrough in Medical Imaging,” *Curr. Med. Imaging Rev.*, vol. 16, pp. 946–956, Oct. 2020, doi: 10.2174/1573405615666191219100824.

[102] S. Balaji, “Binary Image classifier CNN using TensorFlow,” *Techiepedia*, Aug. 29, 2020. https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697 (accessed Jul. 18, 2023).

[103] Z. J. Wang *et al.*, “CNN 101: Interactive Visual Learning for Convolutional Neural Networks,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, in CHI EA ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–7. doi: 10.1145/3334480.3382899.

[104] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” arXiv, May 18, 2015. doi: 10.48550/arXiv.1505.04597.

[105] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, “U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications,” *IEEE Access*, vol. 9, pp. 82031–82057, 2021, doi: 10.1109/ACCESS.2021.3086020.

[106] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, “The Importance of Skip Connections in Biomedical Image Segmentation,” in *Deep Learning and Data Labeling for Medical Applications*, G. Carneiro, D. Mateus, L. Peter, A. Bradley, J. M. R. S. Tavares, V. Belagiannis, J. P. Papa, J. C. Nascimento, M. Loog, Z. Lu, J. S. Cardoso, and J. Cornebise, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 179–187. doi: 10.1007/978-3-319-46976-8_19.

[107] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 10, 2015. doi: 10.48550/arXiv.1512.03385.

[108] S. Li, J. Jiao, Y. Han, and T. Weissman, “Demystifying ResNet.” arXiv, May 20, 2017. doi: 10.48550/arXiv.1611.01186.

[109] O. P. Sharma, M. F. Oswanski, S. Jolly, S. K. Lauer, R. Dressel, and H. A. Stombaugh, “Perils of Rib Fractures,” *Am. Surg.*, vol. 74, no. 4, pp. 310–314, Apr. 2008, doi: 10.1177/000313480807400406.

[110] L. Fabricant, B. Ham, R. Mullins, and J. Mayberry, “Prolonged pain and disability are common after rib fractures,” *Am. J. Surg.*, vol. 205, no. 5, pp. 511–515; discusssion 515-516, May 2013, doi: 10.1016/j.amjsurg.2012.12.007.

[111] M. Kaiume *et al.*, “Rib fracture detection in computed tomography images using deep convolutional neural networks,” *Medicine (Baltimore)*, vol. 100, no. 20, p. e26024, May 2021, doi: 10.1097/MD.0000000000026024.

[112] B. Zhang *et al.*, “Improving rib fracture detection accuracy and reading efficiency with deep learning-based detection software: a clinical evaluation,” *Br. J. Radiol.*, vol. 94, no. 1118, p. 20200870, Feb. 2021, doi: 10.1259/bjr.20200870.

[113] X. H. Meng *et al.*, “A fully automated rib fracture detection system on chest CT images and its impact on radiologist performance,” *Skeletal Radiol.*, vol. 50, no. 9, pp. 1821–1828, Sep. 2021, doi: 10.1007/s00256-021-03709-8.

[114] Y. Hu, X. He, R. Zhang, L. Guo, L. Gao, and J. Wang, “Slice grouping and aggregation network for auxiliary diagnosis of rib fractures,” *Biomed. Signal Process. Control*, vol. 67, p. 102547, May 2021, doi: 10.1016/j.bspc.2021.102547.

[115] L. Yao *et al.*, “Rib fracture detection system based on deep learning,” *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, Dec. 2021, doi: 10.1038/s41598-021-03002-7.

[116] T. Weikert *et al.*, “Assessment of a Deep Learning Algorithm for the Detection of Rib Fractures on Whole-Body Trauma Computed Tomography,” *Korean J. Radiol.*, vol. 21, no. 7, pp. 891–899, Jul. 2020, doi: 10.3348/kjr.2019.0653.

[117] S. Wang, D. Wu, L. Ye, Z. Chen, Y. Zhan, and Y. Li, “Assessment of automatic rib fracture detection on chest CT using a deep learning algorithm,” *Eur. Radiol.*, vol. 33, no. 3, pp. 1824–1834, Mar. 2023, doi: 10.1007/s00330-022-09156-w.

[118] R. Castro-Zunti, K. J. Chae, Y. Choi, G. Y. Jin, and S.-B. Ko, “Assessing the speed-accuracy trade-offs of popular convolutional neural networks for single-crop rib fracture classification,” *Comput. Med. Imaging Graph. Off. J. Comput. Med. Imaging Soc.*, vol. 91, p. 101937, Jul. 2021, doi: 10.1016/j.compmedimag.2021.101937.

[119] L. Jin *et al.*, "Deep-learning-assisted detection and segmentation of rib fractures from CT scans: Development and validation of FracNet," *eBioMedicine*, vol. 62, p. 103106, Dec. 2020, doi: 10.1016/j.ebiom.2020.103106.

[120] M. Wu *et al.*, "Development and Evaluation of a Deep Learning Algorithm for Rib Segmentation and Fracture Detection from Multicenter Chest CT Images," *Radiol. Artif. Intell.*, vol. 3, no. 5, p. e200248, Sep. 2021, doi: 10.1148/ryai.2021200248.

[121] J. Zhang, Z. Li, S. Yan, H. Cao, J. Liu, and D. Wei, "An Algorithm for Automatic Rib Fracture Recognition Combined with nnU-Net and DenseNet," *Evid. Based Complement. Alternat. Med.*, vol. 2022, p. e5841451, Feb. 2022, doi: 10.1155/2022/5841451.

[122] J. Wu *et al.*, "Convolutional neural network for detecting rib fractures on chest radiographs: a feasibility study," *BMC Med. Imaging*, vol. 23, no. 1, p. 18, Jan. 2023, doi: 10.1186/s12880-023-00975-x.

[123] A.-C. Tsai, Y.-Y. Ou, C.-H. Lin, C.-W. Chen, and J.-F. Wang, "Rib Fracture Diagnosis System on Chest X-Rays with Deep Learning," in *2021 9th International Conference on Orange Technology (ICOT)*, Dec. 2021, pp. 1–4. doi: 10.1109/ICOT54518.2021.9680611.

[124] Q.-Q. Zhou *et al.*, "Automatic detection and classification of rib fractures based on patients' CT images and clinical information via convolutional neural network," *Eur. Radiol.*, vol. 31, no. 6, pp. 3815–3825, Jun. 2021, doi: 10.1007/s00330-020-07418-z.

[125] P.-H. C. Chen, Y. Liu, and L. Peng, "How to develop machine learning models for healthcare," *Nat. Mater.*, vol. 18, no. 5, pp. 410–414, May 2019, doi: 10.1038/s41563-019-0345-0.

[126] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism.*, vol. 69S, pp. S36–S40, Apr. 2017, doi: 10.1016/j.metabol.2017.01.011.

[127] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning." arXiv, Dec. 13, 2017. doi: 10.48550/arXiv.1712.04621.

[128] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, May 2018, pp. 117–122. doi: 10.1109/IIPHDW.2018.8388338.

[129] I. J. Goodfellow *et al.*, "Generative Adversarial Networks." arXiv, Jun. 10, 2014. Accessed: Apr. 12, 2023. [Online]. Available: http://arxiv.org/abs/1406.2661

[130] K. Armanious *et al.*, "MedGAN: Medical Image Translation using GANs," *Comput. Med. Imaging Graph.*, vol. 79, p. 101684, Jan. 2020, doi: 10.1016/j.compmedimag.2019.101684.

[131] C. Andrade, L. F. Teixeira, M. J. M. Vasconcelos, and L. Rosado, "Data Augmentation Using Adversarial Image-to-Image Translation for the Segmentation of Mobile-Acquired Dermatological Images," *J. Imaging*, vol. 7, no. 1, p. 2, Dec. 2020, doi: 10.3390/jimaging7010002.

[132] C. Han *et al.*, “Combining Noise-to-Image and Image-to-Image GANs: Brain MR Image Augmentation for Tumor Detection,” *IEEE Access*, vol. 7, pp. 156966–156977, 2019, doi: 10.1109/ACCESS.2019.2947606.

[133] P. Welander, S. Karlsson, and A. Eklund, “Generative Adversarial Networks for Image-to-Image Translation on Multi-Contrast MR Images - A Comparison of CycleGAN and UNIT.” arXiv, Jun. 20, 2018. doi: 10.48550/arXiv.1806.07777.

[134] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.” arXiv, Jan. 07, 2016. doi: 10.48550/arXiv.1511.06434.

[135] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, “Unrolled Generative Adversarial Networks.” arXiv, May 12, 2017. doi: 10.48550/arXiv.1611.02163.

[136] M. Arjovsky and L. Bottou, “Towards Principled Methods for Training Generative Adversarial Networks.” arXiv, Jan. 17, 2017. Accessed: Apr. 13, 2023. [Online]. Available: http://arxiv.org/abs/1701.04862

[137] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization Methods for Large-Scale Machine Learning.” arXiv, Feb. 08, 2018. doi: 10.48550/arXiv.1606.04838.

[138] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral Normalization for Generative Adversarial Networks.” arXiv, Feb. 16, 2018. Accessed: Apr. 13, 2023. [Online]. Available: http://arxiv.org/abs/1802.05957

[139] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least Squares Generative Adversarial Networks”.

[140] S. Nowozin, B. Cseke, and R. Tomioka, “f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.” arXiv, Jun. 02, 2016. doi: 10.48550/arXiv.1606.00709.

[141] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, “Stabilizing Training of Generative Adversarial Networks through Regularization,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Apr. 13, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/7bccfde7714a1ebadf06c5f4cea752c1-Abstract.html

[142] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, “Amortised MAP Inference for Image Super-resolution.” arXiv, Feb. 21, 2017. doi: 10.48550/arXiv.1610.04490.

[143] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, “On Convergence and Stability of GANs.” arXiv, Dec. 10, 2017. Accessed: Apr. 13, 2023. [Online]. Available: http://arxiv.org/abs/1705.07215

[144] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN.” arXiv, Dec. 06, 2017. doi: 10.48550/arXiv.1701.07875.

[145] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved Training of Wasserstein GANs.” arXiv, Dec. 25, 2017. doi: 10.48550/arXiv.1704.00028.

[146] L. Mescheder, A. Geiger, and S. Nowozin, “Which Training Methods for GANs do actually Converge?” arXiv, Jul. 31, 2018. Accessed: Apr. 13, 2023. [Online]. Available: http://arxiv.org/abs/1801.04406

[147] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets.” arXiv, Nov. 06, 2014. doi: 10.48550/arXiv.1411.1784.

[148] S. Ma, J. Fu, C. W. Chen, and T. Mei, “DA-GAN: Instance-level Image Translation by Deep Attention Generative Adversarial Networks (with Supplementary Materials).” arXiv, Feb. 18, 2018. Accessed: Apr. 14, 2023. [Online]. Available: http://arxiv.org/abs/1802.06454

[149] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation.” arXiv, Feb. 26, 2018. Accessed: Apr. 15, 2023. [Online]. Available: http://arxiv.org/abs/1710.10196

[150] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks.” arXiv, Mar. 29, 2019. doi: 10.48550/arXiv.1812.04948.

[151] A. Odena, C. Olah, and J. Shlens, “Conditional Image Synthesis With Auxiliary Classifier GANs.” arXiv, Jul. 20, 2017. doi: 10.48550/arXiv.1610.09585.

[152] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks.” arXiv, Nov. 26, 2018. doi: 10.48550/arXiv.1611.07004.

[153] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation.” arXiv, Sep. 21, 2018. Accessed: Apr. 15, 2023. [Online]. Available: http://arxiv.org/abs/1711.09020

[154] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.” arXiv, Aug. 24, 2020. doi: 10.48550/arXiv.1703.10593.

[155] Y. Pang, J. Lin, T. Qin, and Z. Chen, “Image-to-Image Translation: Methods and Applications.” arXiv, Jul. 03, 2021. Accessed: Apr. 14, 2023. [Online]. Available: http://arxiv.org/abs/2101.08629

[156] S. Tripathy, J. Kannala, and E. Rahtu, “Learning Image-to-Image Translation Using Paired and Unpaired Training Samples,” in *Computer Vision – ACCV 2018*, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 51–66. doi: 10.1007/978-3-030-20890-5_4.

[157] E. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks.” arXiv, Jun. 18, 2015. doi: 10.48550/arXiv.1506.05751.

[158] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Sep. 1999, pp. 1033–1038 vol.2. doi: 10.1109/ICCV.1999.790383.

[159] “CycleGAN.” https://hardikbansal.github.io/CycleGANBlog/ (accessed Jul. 18, 2023).

[160] Y. Chen *et al.*, “Generative Adversarial Networks in Medical Image augmentation: A review,” *Comput. Biol. Med.*, vol. 144, p. 105382, May 2022, doi: 10.1016/j.compbiomed.2022.105382.

[161] X. Yi, E. Walia, and P. Babyn, “Generative Adversarial Network in Medical Imaging: A Review,” *Med. Image Anal.*, vol. 58, p. 101552, Dec. 2019, doi: 10.1016/j.media.2019.101552.

[162] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[163] A. Borji, “Pros and cons of GAN evaluation measures,” *Comput. Vis. Image Underst.*, vol. 179, pp. 41–65, Feb. 2019, doi: 10.1016/j.cviu.2018.10.009.

[164] T. Salimans *et al.*, “Improved Techniques for Training GANs,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2016. Accessed: Apr. 14, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html

[165] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Apr. 14, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html

[166] B. Segal, D. M. Rubin, G. Rubin, and A. Pantanowitz, “Evaluating the Clinical Realism of Synthetic Chest X-Rays Generated Using Progressively Growing GANs,” *Sn Comput. Sci.*, vol. 2, no. 4, p. 321, 2021, doi: 10.1007/s42979-021-00720-7.

[167] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, “Are GANs Created Equal? A Large-Scale Study,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2018. Accessed: Apr. 14, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/e46de7e1bcaaced9a54f1e9d0d2f800d-Abstract.html

[168] Z. Zhou, W. Zhang, and J. Wang, “Inception Score, Label Smoothing, Gradient Vanishing and -log(D(x)) Alternative,” Aug. 2017. doi: 10.48550/arXiv.1708.01729.

[169] A. Obukhov and M. Krasnyanskiy, “Quality Assessment Method for GAN Based on Modified Metrics Inception Score and Fréchet Inception Distance,” in *Software Engineering Perspectives in Intelligent Systems*, R. Silhavy, P. Silhavy, and Z. Prokopova, Eds., in Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, pp. 102–114. doi: 10.1007/978-3-030-63322-6_8.

[170] Y. Chen *et al.*, “AI-Based Reconstruction for Fast MRI—A Systematic Review and Meta-Analysis,” *Proc. IEEE*, vol. 110, no. 2, pp. 224–245, Feb. 2022, doi: 10.1109/JPROC.2022.3141367.

[171] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, “CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection,” *IEEE Access*, vol. 8, pp. 91916–91923, 2020, doi: 10.1109/ACCESS.2020.2994762.

[172] S. A. Kamran, K. F. Hossain, A. Tavakkoli, S. L. Zuckerbrod, K. M. Sanders, and S. A. Baker, “RV-GAN: Segmenting Retinal Vascular Structure in Fundus Photographs Using a Novel Multi-scale Generative Adversarial Network,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 34–44. doi: 10.1007/978-3-030-87237-3_4.

[173] S. U. H. Dar, M. Yurt, M. Shahdloo, M. E. Ildız, B. Tınaz, and T. Çukur, “Prior-Guided Image Reconstruction for Accelerated Multi-Contrast MRI via Generative Adversarial Networks,” *IEEE J. Sel. Top. Signal Process.*, vol. 14, no. 6, pp. 1072–1087, Oct. 2020, doi: 10.1109/JSTSP.2020.3001737.

[174] E. Cha, H. Chung, E. Y. Kim, and J. C. Ye, “Unpaired Training of Deep Learning tMRA for Flexible Spatio-Temporal Resolution,” *IEEE Trans. Med. Imaging*, vol. 40, no. 1, pp. 166–179, Jan. 2021, doi: 10.1109/TMI.2020.3023620.

[175] V. Edupuganti, M. Mardani, S. Vasanawala, and J. Pauly, “Uncertainty Quantification in Deep MRI Reconstruction.” arXiv, Apr. 25, 2020. doi: 10.48550/arXiv.1901.11228.

[176] M. Jiang *et al.*, “Accelerating CS-MRI Reconstruction With Fine-Tuning Wasserstein Generative Adversarial Network,” *IEEE Access*, vol. 7, pp. 152347–152357, 2019, doi: 10.1109/ACCESS.2019.2948220.

[177] G. Oh, B. Sim, H. Chung, L. Sunwoo, and J. C. Ye, “Unpaired Deep Learning for Accelerated MRI using Optimal Transport Driven CycleGAN.” arXiv, Aug. 29, 2020. doi: 10.48550/arXiv.2008.12967.

[178] R. Shaul, I. David, O. Shitrit, and T. Riklin Raviv, "Subsampled brain MRI reconstruction by generative adversarial neural networks," *Med. Image Anal.*, vol. 65, p. 101747, Oct. 2020, doi: 10.1016/j.media.2020.101747.

[179] S. Bera and P. K. Biswas, "Noise Conscious Training of Non Local Neural Network Powered by Self Attentive Spectral Normalized Markovian Patch GAN for Low Dose CT Denoising," *IEEE Trans. Med. Imaging*, vol. 40, no. 12, pp. 3663–3673, Dec. 2021, doi: 10.1109/TMI.2021.3094525.

[180] A. B. Qasim *et al.*, "Red-GAN: Attacking class imbalance via conditioned generation. Yet another perspective on medical image synthesis for skin lesion dermoscopy and brain tumor MRI." arXiv, Mar. 27, 2021. Accessed: Jul. 16, 2023. [Online]. Available: http://arxiv.org/abs/2004.10734

[181] Z. Huang, J. Zhang, Y. Zhang, and H. Shan, "DU-GAN: Generative Adversarial Networks With Dual-Domain U-Net-Based Discriminators for Low-Dose CT Denoising," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2021.3128703.

[182] M. Li, W. Hsu, X. Xie, J. Cong, and W. Gao, "SACNN: Self-Attention Convolutional Neural Network for Low-Dose CT Denoising With Self-Supervised Perceptual Loss Network," *IEEE Trans. Med. Imaging*, vol. 39, no. 7, pp. 2289–2301, Jul. 2020, doi: 10.1109/TMI.2020.2968472.

[183] Y. Ma, B. Wei, P. Feng, P. He, X. Guo, and G. Wang, "Low-Dose CT Image Denoising Using a Generative Adversarial Network With a Hybrid Loss Function for Noise Learning," *IEEE Access*, vol. 8, pp. 67519–67529, 2020, doi: 10.1109/ACCESS.2020.2986388.

[184] Z. Li, S. Zhou, J. Huang, L. Yu, and M. Jin, "Investigation of Low-Dose CT Image Denoising Using Unpaired Deep Learning Methods," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 5, no. 2, pp. 224–234, Mar. 2021, doi: 10.1109/TRPMS.2020.3007583.

[185] Z. Yin, K. Xia, Z. He, J. Zhang, S. Wang, and B. Zu, "Unpaired Image Denoising via Wasserstein GAN in Low-Dose CT Image with Multi-Perceptual Loss and Fidelity Loss," *Symmetry*, vol. 13, no. 1, Art. no. 1, Jan. 2021, doi: 10.3390/sym13010126.

[186] Q. Yang *et al.*, "Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss," *IEEE Trans. Med. Imaging*, vol. 37, no. 6, pp. 1348–1357, Jun. 2018, doi: 10.1109/TMI.2018.2827462.

[187] M. Gholizadeh-Ansari, J. Alirezaie, and P. Babyn, "Deep Learning for Low-Dose CT Denoising Using Perceptual Loss and Edge Detection Layer," *J. Digit. Imaging*, vol. 33, no. 2, pp. 504–515, Apr. 2020, doi: 10.1007/s10278-019-00274-4.

[188] W. Lingle *et al.*, "The Cancer Genome Atlas Breast Invasive Carcinoma Collection (TCGA-BRCA)." The Cancer Imaging Archive, 2016. doi: 10.7937/K9/TCIA.2016.AB2NAZRP.

[189] H. Xue *et al.*, "A 3D attention residual encoder–decoder least-square GAN for low-count PET denoising," *Nucl. Instrum. Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.*, vol. 983, p. 164638, Dec. 2020, doi: 10.1016/j.nima.2020.164638.

[190] Y. Luo *et al.*, "3D Transformer-GAN for High-Quality PET Reconstruction," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 276–285. doi: 10.1007/978-3-030-87231-1_27.

[191] J. Ouyang, K. T. Chen, E. Gong, J. Pauly, and G. Zaharchuk, "Ultra-low-dose PET reconstruction using generative adversarial network with feature matching and task-specific perceptual loss," *Med. Phys.*, vol. 46, no. 8, pp. 3555–3564, 2019, doi: 10.1002/mp.13626.

[192] K. T. Chen *et al.*, "Ultra-Low-Dose 18F-Florbetaben Amyloid PET Imaging Using Deep Learning with Multi-Contrast MRI Inputs," *Radiology*, vol. 290, no. 3, pp. 649–656, Mar. 2019, doi: 10.1148/radiol.2018180940.

[193] Z. Xie *et al.*, "Generative adversarial network based regularized image reconstruction for PET," *Phys. Med. Biol.*, vol. 65, no. 12, p. 125016, Jun. 2020, doi: 10.1088/1361-6560/ab8f72.

[194] H. Xue *et al.*, "LCPR-Net: low-count PET image reconstruction using the domain transform and cycle-consistent generative adversarial networks," *Quant. Imaging Med. Surg.*, vol. 11, no. 2, pp. 749–762, Feb. 2021, doi: 10.21037/qims-20-66.

[195] J. Ouyang, G. Wang, E. Gong, K. Chen, J. Pauly, and G. Zaharchuk, "Task-GAN: Improving Generative Adversarial Network for Image Reconstruction," in *Machine Learning for Medical Image Reconstruction*, F. Knoll, A. Maier, D. Rueckert, and J. C. Ye, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 193–204. doi: 10.1007/978-3-030-33843-5_18.

[196] C. Decourt and L. Duong, "Semi-supervised generative adversarial networks for the segmentation of the left ventricle in pediatric MRI," *Comput. Biol. Med.*, vol. 123, p. 103884, Aug. 2020, doi: 10.1016/j.compbiomed.2020.103884.

[197] M. Yang *et al.*, "Automated knee cartilage segmentation for heterogeneous clinical MRI using generative adversarial networks with transfer learning," *Quant. Imaging Med. Surg.*, vol. 12, no. 5, pp. 2620633–2622633, May 2022, doi: 10.21037/qims-21-459.

[198] M. M. R. Siddiquee *et al.*, "Learning Fixed Points in Generative Adversarial Networks: From Image-to-Image Translation to Disease Detection and Localization." arXiv, Aug. 29, 2019. Accessed: Apr. 15, 2023. [Online]. Available: http://arxiv.org/abs/1908.06965

[199] C. Han *et al.*, "Infinite Brain MR Images: PGGAN-Based Data Augmentation for Tumor Detection," in *Neural Approaches to Dynamics of Signal Exchanges*, A. Esposito, M. Faundez-Zanuy, F. C. Morabito, and E. Pasero, Eds., in Smart Innovation, Systems and Technologies. Singapore: Springer, 2020, pp. 291–303. doi: 10.1007/978-981-13-8950-4_27.

[200] G. Kwon, C. Han, and D. Kim, "Generation of 3D Brain MRI Using Auto-Encoding Generative Adversarial Networks," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 118–126. doi: 10.1007/978-3-030-32248-9_14.

[201] S. Kaur, H. Aggarwal, and R. Rani, "MR Image Synthesis Using Generative Adversarial Networks for Parkinson's Disease Classification," P. Bansal, M. Tushir, V. E. Balas, and R. Srivastava, Eds., in Advances in Intelligent Systems and Computing, vol. 1164. Singapore: Springer Singapore, 2021, pp. 317–327. doi: 10.1007/978-981-15-4992-2_30.

[202] X. Gu, H. Knutsson, M. Nilsson, and A. Eklund, "Generating Diffusion MRI Scalar Maps from T1 Weighted Images Using Generative Adversarial Networks," in *Image Analysis*, M. Felsberg, P.-E. Forssén, I.-M. Sintorn, and J. Unger, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 489–498. doi: 10.1007/978-3-030-20205-7_40.

[203] W. Tu, W. Hu, X. Liu, and J. He, "DRPAN: A novel Adversarial Network Approach for Retinal Vessel Segmentation," in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Jun. 2019, pp. 228–232. doi: 10.1109/ICIEA.2019.8833908.

[204] J. Son, S. J. Park, and K.-H. Jung, "Towards Accurate Segmentation of Retinal Vessels and the Optic Disc in Fundoscopic Images with Generative Adversarial Networks," *J. Digit. Imaging*, vol. 32, no. 3, pp. 499–512, Jun. 2019, doi: 10.1007/s10278-018-0126-3.

[205] H. Xie *et al.*, "AMD-GAN: Attention encoder and multi-branch structure based generative adversarial networks for fundus disease detection from scanning laser ophthalmoscopy images," *Neural Netw.*, vol. 132, pp. 477–490, Dec. 2020, doi: 10.1016/j.neunet.2020.09.005.

[206] Z. Yu, Q. Xiang, J. Meng, C. Kou, Q. Ren, and Y. Lu, "Retinal image synthesis from multiple-landmarks input with generative adversarial networks," *Biomed. Eng. OnLine*, vol. 18, no. 1, p. 62, May 2019, doi: 10.1186/s12938-019-0682-x.

[207] Y. Zhou, X. He, S. Cui, F. Zhu, L. Liu, and L. Shao, "High-Resolution Diabetic Retinopathy Image Synthesis Manipulated by Grading and Lesions," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 505–513. doi: 10.1007/978-3-030-32239-7_56.

[208] T. Zhang *et al.*, "SkrGAN: Sketching-Rendering Unconditional Generative Adversarial Networks for Medical Image Synthesis," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 777–785. doi: 10.1007/978-3-030-32251-9_85.

[209] B. Lei *et al.*, “Skin lesion segmentation via generative adversarial networks with dual discriminators,” *Med. Image Anal.*, vol. 64, p. 101716, Aug. 2020, doi: 10.1016/j.media.2020.101716.

[210] P. Sedigh, R. Sadeghian, and M. T. Masouleh, “Generating Synthetic Medical Images by Using GAN to Improve CNN Performance in Skin Cancer Classification,” in *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, Nov. 2019, pp. 497–502. doi: 10.1109/ICRoM48714.2019.9071823.

[211] A. Gupta, S. Venkatesh, S. Chopra, and C. Ledig, “Generative Image Translation for Data Augmentation of Bone Lesion Pathology,” in *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*, PMLR, May 2019, pp. 225–235. Accessed: Apr. 16, 2023. [Online]. Available: https://proceedings.mlr.press/v102/gupta19b.html

[212] N. Saffari *et al.*, “Fully Automated Breast Density Segmentation and Classification Using Deep Learning,” *Diagnostics*, vol. 10, no. 11, p. 988, Nov. 2020, doi: 10.3390/diagnostics10110988.

[213] T. Shen, C. Gou, F.-Y. Wang, Z. He, and W. Chen, “Learning from adversarial medical images for X-ray breast mass segmentation,” *Comput. Methods Programs Biomed.*, vol. 180, p. 105012, Oct. 2019, doi: 10.1016/j.cmpb.2019.105012.

[214] L. Han, Y. Lyu, C. Peng, and S. K. Zhou, “GAN-based disentanglement learning for chest X-ray rib suppression,” *Med. Image Anal.*, vol. 77, p. 102369, Apr. 2022, doi: 10.1016/j.media.2022.102369.

[215] S. Sundaram and N. Hulkund, “GAN-based Data Augmentation for Chest X-ray Classification.” arXiv, Jul. 06, 2021. doi: 10.48550/arXiv.2107.02970.

[216] H. Wang, H. Gu, P. Qin, and J. Wang, “U-shaped GAN for Semi-Supervised Learning and Unsupervised Domain Adaptation in High Resolution Chest Radiograph Segmentation,” *Front. Med.*, vol. 8, 2022, Accessed: Apr. 16, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmed.2021.782664

[217] G. Rani, A. Misra, V. S. Dhaka, E. Zumpano, and E. Vocaturo, “Spatial feature and resolution maximization GAN for bone suppression in chest radiographs,” *Comput. Methods Programs Biomed.*, vol. 224, p. 107024, Sep. 2022, doi: 10.1016/j.cmpb.2022.107024.

[218] I. Guha *et al.*, “Deep learning based high-resolution reconstruction of trabecular bone microstructures from low-resolution CT scans using GAN-CIRCLE,” in *Medical Imaging 2020: Biomedical Applications in Molecular, Structural, and Functional Imaging*, SPIE, Feb. 2020, pp. 204–214. doi: 10.1117/12.2549318.

[219] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, “Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks,” *Sci. Rep.*, vol. 9, no. 1, Art. no. 1, Nov. 2019, doi: 10.1038/s41598-019-52737-x.

[220] Z. Shi, Q. Hu, Y. Yue, Z. Wang, O. M. S. AL-Othmani, and H. Li, "Automatic Nodule Segmentation Method for CT Images Using Aggregation-U-Net Generative Adversarial Networks," *Sens. Imaging*, vol. 21, no. 1, p. 39, Jul. 2020, doi: 10.1007/s11220-020-00304-4.

[221] Z. Xu *et al.*, "Tunable CT Lung Nodule Synthesis Conditioned on Background Image and Semantic Features," in *Simulation and Synthesis in Medical Imaging*, N. Burgos, A. Gooya, and D. Svoboda, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 62–70. doi: 10.1007/978-3-030-32778-1_7.

[222] Y. Luo *et al.*, "Adaptive rectification based adversarial network with spectrum constraint for high-quality PET image synthesis," *Med. Image Anal.*, vol. 77, p. 102335, Apr. 2022, doi: 10.1016/j.media.2021.102335.

[223] J. Islam and Y. Zhang, "GAN-based synthetic brain PET image generation," *Brain Inform.*, vol. 7, no. 1, p. 3, Mar. 2020, doi: 10.1186/s40708-020-00104-2.

[224] J. Liang *et al.*, "Sketch guided and progressive growing GAN for realistic and editable ultrasound image synthesis," *Med. Image Anal.*, vol. 79, p. 102461, Jul. 2022, doi: 10.1016/j.media.2022.102461.

[225] D. Zhai, B. Hu, X. Gong, H. Zou, and J. Luo, "ASS-GAN: Asymmetric semi-supervised GAN for breast ultrasound image segmentation," *Neurocomputing*, vol. 493, pp. 204–216, Jul. 2022, doi: 10.1016/j.neucom.2022.04.021.

[226] T. Pang, J. H. D. Wong, W. L. Ng, and C. S. Chan, "Semi-supervised GAN-based Radiomics Model for Data Augmentation in Breast Ultrasound Mass Classification," *Comput. Methods Programs Biomed.*, vol. 203, p. 106018, May 2021, doi: 10.1016/j.cmpb.2021.106018.

[227] Y. Hang, "Thyroid Nodule Classification in Ultrasound Images by Fusion of Conventional Features and Res-GAN Deep Features," *J. Healthc. Eng.*, vol. 2021, p. e9917538, Jul. 2021, doi: 10.1155/2021/9917538.

[228] A. Z. Alsinan, C. Rule, M. Vives, V. M. Patel, and I. Hacihaliloglu, "GAN-Based Realistic Bone Ultrasound Image and Label Synthesis for Improved Segmentation," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 795–804. doi: 10.1007/978-3-030-59725-2_77.

[229] A. Shokraei Fard, D. C. Reutens, and V. Vegh, "From CNNs to GANs for cross-modality medical image estimation," *Comput. Biol. Med.*, vol. 146, p. 105556, Jul. 2022, doi: 10.1016/j.compbiomed.2022.105556.

[230] S. Hu, Y. Shen, S. Wang, and B. Lei, "Brain MR to PET Synthesis via Bidirectional Generative Adversarial Network," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A.

Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 698–707. doi: 10.1007/978-3-030-59713-9_67.

[231] A. Sikka, Skand, J. S. Virk, and D. R. Bathula, "MRI to PET Cross-Modality Translation using Globally and Locally Aware GAN (GLA-GAN) for Multi-Modal Diagnosis of Alzheimer's Disease." arXiv, Aug. 04, 2021. Accessed: Apr. 16, 2023. [Online]. Available: http://arxiv.org/abs/2108.02160

[232] F. Bazangani, F. J. P. Richard, B. Ghattas, and E. Guedj, "FDG-PET to T1 Weighted MRI Translation with 3D Elicit Generative Adversarial Network (E-GAN)," *Sensors*, vol. 22, no. 12, Art. no. 12, Jan. 2022, doi: 10.3390/s22124640.

[233] X. Gao, F. Shi, D. Shen, and M. Liu, "Task-Induced Pyramid and Attention GAN for Multimodal Brain Image Imputation and Classification in Alzheimer's Disease," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 1, pp. 36–43, Jan. 2022, doi: 10.1109/JBHI.2021.3097721.

[234] H.-C. Shin *et al.*, "GANBERT: Generative Adversarial Networks with Bidirectional Encoder Representations from Transformers for MRI to PET synthesis." arXiv, Aug. 10, 2020. doi: 10.48550/arXiv.2008.04393.

[235] M. Hammami, D. Friboulet, and R. Kechichian, "Cycle GAN-Based Data Augmentation For Multi-Organ Detection In CT Images Via Yolo," in *2020 IEEE International Conference on Image Processing (ICIP)*, Oct. 2020, pp. 390–393. doi: 10.1109/ICIP40778.2020.9191127.

[236] R. Oulbacha and S. Kadoury, "MRI to CT Synthesis of the Lumbar Spine from a Pseudo-3D Cycle GAN," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2020, pp. 1784–1787. doi: 10.1109/ISBI45749.2020.9098421.

[237] J. M. Wolterink, A. M. Dinkla, M. H. F. Savenije, P. R. Seevinck, C. A. T. van den Berg, and I. Išgum, "Deep MR to CT Synthesis Using Unpaired Data," in *Simulation and Synthesis in Medical Imaging*, S. A. Tsaftaris, A. Gooya, A. F. Frangi, and J. L. Prince, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 14–23. doi: 10.1007/978-3-319-68127-6_2.

[238] Y. Liu *et al.*, "CT synthesis from MRI using multi-cycle GAN for head-and-neck radiation therapy," *Comput. Med. Imaging Graph.*, vol. 91, p. 101953, Jul. 2021, doi: 10.1016/j.compmedimag.2021.101953.

[239] A. Ranjan, D. Lalwani, and R. Misra, "GAN for synthesizing CT from T2-weighted MRI data towards MR-guided radiation treatment," *Magn. Reson. Mater. Phys. Biol. Med.*, vol. 35, no. 3, pp. 449–457, Jun. 2022, doi: 10.1007/s10334-021-00974-5.

[240] A. Singh *et al.*, "Automated nonlinear registration of coronary PET to CT angiography using pseudo-CT generated from PET with generative adversarial networks," *J. Nucl. Cardiol.*, Jun. 2022, doi: 10.1007/s12350-022-03010-8.

[241] A. Ben-Cohen *et al.*, “Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection,” *Eng. Appl. Artif. Intell.*, vol. 78, pp. 186–194, Feb. 2019, doi: 10.1016/j.engappai.2018.11.013.

[242] K. Armanious, C. Jiang, S. Abdulatif, T. Küstner, S. Gatidis, and B. Yang, “Unsupervised Medical Image Translation Using Cycle-MedGAN,” in *2019 27th European Signal Processing Conference (EUSIPCO)*, Sep. 2019, pp. 1–5. doi: 10.23919/EUSIPCO.2019.8902799.

[243] S. E. Darling, S. L. Done, S. D. Friedman, and K. W. Feldman, “Frequency of intrathoracic injuries in children younger than 3 years with rib fractures,” *Pediatr. Radiol.*, vol. 44, no. 10, pp. 1230–1236, Oct. 2014, doi: 10.1007/s00247-014-2988-y.

[244] C. W. Paine, O. Fakeye, C. W. Christian, and J. N. Wood, “Prevalence of Abuse Among Young Children With Rib Fractures: A Systematic Review,” *Pediatr. Emerg. Care*, vol. 35, no. 2, pp. 96–103, Feb. 2019, doi: 10.1097/PEC.0000000000000911.

[245] D. F. Merten, M. A. Radkowski, and J. C. Leonidas, “The abused child: a radiological reappraisal,” *Radiology*, vol. 146, no. 2, pp. 377–381, Feb. 1983, doi: 10.1148/radiology.146.2.6849085.

[246] J. Burkow, G. Holste, J. Otjen, F. Perez, J. Junewick, and A. Alessio, “Avalanche decision schemes to improve pediatric rib fracture detection,” in *Medical Imaging 2022: Computer-Aided Diagnosis*, SPIE, Apr. 2022, pp. 611–618. doi: 10.1117/12.2611013.

[247] A. Ghosh *et al.*, “A Patch-Based Deep Learning Approach for Detecting Rib Fractures on Frontal Radiographs in Young Children,” *J. Digit. Imaging*, Mar. 2023, doi: 10.1007/s10278-023-00793-1.

[248] D. Hayashi *et al.*, “Automated detection of acute appendicular skeletal fractures in pediatric patients using deep learning,” *Skeletal Radiol.*, vol. 51, no. 11, pp. 2129–2139, Nov. 2022, doi: 10.1007/s00256-022-04070-0.

[249] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[250] P. Perez, M. Gangnet, and A. Blake, “Poisson image editing,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003, doi: https://doi.org/10.1145/882262.882269.

[251] J. Kim, “Gradient Domain Fusion,” 2021. https://www.andrew.cmu.edu/course/16-726/projects/juyongk/proj2/ (accessed Jul. 05, 2023).

[252] G. Jocher, “YOLOv5 by Ultralytics.” May 2020. doi: 10.5281/zenodo.3908559.

[253] G. Jocher, “YOLOv3 in PyTorch.” May 2020. doi: 10.5281/zenodo.3908559.

[254] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN." arXiv, Nov. 26, 2019. doi: 10.48550/arXiv.1911.11929.

[255] T. Kanayama *et al.*, "Gastric Cancer Detection from Endoscopic Images Using Synthesis by GAN," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 530–538. doi: 10.1007/978-3-030-32254-0_59.

[256] C. Han *et al.*, "Learning More with Less: Conditional PGGAN-based Data Augmentation for Brain Metastases Detection Using Highly-Rough Annotation on MR Images," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, in CIKM '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 119–127. doi: 10.1145/3357384.3357890.

[257] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson, "The Curse of Recursion: Training on Generated Data Makes Models Forget." arXiv, May 31, 2023. doi: 10.48550/arXiv.2305.17493.

[258] "DALL·E 2." https://openai.com/dall-e-2 (accessed Jul. 20, 2023).

[259] "Midjourney," *Midjourney*. https://www.midjourney.com/home/?callbackUrl=%2Fapp%2F (accessed Jul. 20, 2023).

[260] "Stable Diffusion Public Release," *Stability AI*. https://stability.ai/blog/stable-diffusion-public-release (accessed Jul. 20, 2023).

[261] L. Yang *et al.*, "Diffusion Models: A Comprehensive Survey of Methods and Applications." arXiv, Mar. 23, 2023. doi: 10.48550/arXiv.2209.00796.

[262] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-GAN: Training GANs with Diffusion." arXiv, Oct. 08, 2022. doi: 10.48550/arXiv.2206.02262.

[263] P. Dhariwal and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 8780–8794. Accessed: Jul. 20, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html

[264] M. Stypułkowski, K. Vougioukas, S. He, M. Zięba, S. Petridis, and M. Pantic, "Diffused Heads: Diffusion Models Beat GANs on Talking-Face Generation." arXiv, Jan. 06, 2023. doi: 10.48550/arXiv.2301.03396.

[265] G. Müller-Franzes *et al.*, "Diffusion Probabilistic Models beat GANs on Medical Images." arXiv, Dec. 14, 2022. doi: 10.48550/arXiv.2212.07501.

[266] M. U. Akbar, M. Larsson, and A. Eklund, “Brain tumor segmentation using synthetic MR images -- A comparison of GANs and diffusion models.” arXiv, Jun. 05, 2023. doi: 10.48550/arXiv.2306.02986.

APPENDIX: DATA, CODE, AND SUPPLEMENTAL INFORMATION

Data Repositories

dCTP Myocardial Perfusion Studies

Raw PET and CT scans, myocardial time attenuation curves, coronary CT angiography data, and patient demographs/risk factors can be found here: https://doi.org/10.7910/DVN/VUP5TC.

Implantable TaOx Polymeric Biomedical Devices

Raw μCT images of scaffolds are available upon request.

Pediatric Chest Radiographs

Due to the legal requirements, original radiographs is not available.

Code Repositories

dCTP Myocardial Perfusion Studies

Matlab code to generate perfusion estimation, as well as the trained models can be found here: https://github.com/tuethan/Machine-Learned-CT-Perfusion-Estimation.

Matlab code to generate $S_{FFR\text{-}CTPA}$ scores and to assess diagnostic accuracy can be found here: https://github.com/tuethan/FFR-CTPA-Diagnostic-Accuracy.

Implantable TaOx Polymeric Biomedical Devices

Matlab code for segmentation and for metric calculations can be found here: https://github.com/tuethan/TaOx-Scaffold-Segmentation.

Pediatric Chest Radiographs

Matlab code to generate near-pairs, to generate full synthetic radiographs, to train normal/NPP/FID-NPP cycleGANs, as well as the trained models are available here: https://github.com/tuethan/Pediatric-Chest-Radiograph-Data-Augmentation. Python code to train

the YOLOv5 object detector can be found in the same repository.