Bachelor's thesis

Information and Communications Technology

2022

Dorin Doncenco

Exploring Medical Image Data Augmentation and Synthesis using conditional Generative Adversarial Networks

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and Communications Technology

2022

Dorin Doncenco

Exploring Medical Image Data Augmentation and Synthesis using conditional Generative Adversarial Networks

Obtaining healthcare data such as magnetic resonance imaging data for medical diagnosis is expensive and time-consuming. In this thesis, a method using generative adversarial networks is explored for synthesizing data of brain gliomas to improve the performance of image segmentation algorithms. The network was trained to create subjects with gliomas from a given label, and the network is able to synthesize visible tumors. The data was evaluated using DeepMedic, an image segmentation convolutional neural network. The performance of the model on the augmented dataset was benchmarked against the unaugmented dataset, and its performance was not improved. An analysis on the data is presented, and a future direction is given for how the generative adversarial network can be improved.

Acknowledgments:

The author wishes to acknowledge CSC – IT Center for Science, Finland, for computational resources provided. This work was supported by the Business Finland [Grant 33961/31/2020]. Thanks to Mojtaba Jafaritadi for providing guidance on the research.

Keywords:

deep learning, generative adversarial networks, gliomas, image segmentation, medical imaging

Content

List of abbreviations

BraTS	Brain Tumor Segmentation, a challenge for benchmarking automated segmentation algorithms
CT	Computerized Tomography, a medical imaging technique
DL	Deep Learning, a class of machine learning algorithms
DSC	Dice Similarity Coefficient, a metric to determine similarity between two samples
GAN	Generative Adversarial Networks, a deep learning framework
ML	Machine Learning, a subfield of artificial intelligence for optimizing algorithms automatically
MRI	Magnetic Resonance Imaging, a medical imaging technique
MSE	Mean Squared Error, a metric for estimating error
NIFTI	Neuroimaging Informatics Technology Initiative, an open file format
PET	Positron Emission Tomography, a functional imaging technique
ReLU	Rectified Linear Unit, an activation function
SN	Spectral Normalization, a normalization method for generative adversarial networks
VRAM	Video Random Access Memory

1 Introduction

Being able to segment and diagnose brain tumors early in their development is critical for planning treatment and increasing the survival rate of the patients. Manual segmentation carried out by an expert radiologist is the most accurate method of identifying the tumor tissue in magnetic resonance imaging (MRI). Unfortunately, this process is time-consuming and it is not completely perfect, as manually segmenting tumors repeatedly can lead to slight variations in the identified tissue due to inter-observer and intra-observer errors [1]. Researchers have been eager to automate medical image analysis and have successfully done so for many tasks using deep learning frameworks [2]. However, such frameworks require a great quantity of labeled data which need to be manually prepared and segmented [3]. The goal of this thesis is to solve the problem of insufficient data for automated medical image analysis by synthesizing new data samples to augment existing data sets. To do so, the following questions must be answered:

1) How can label-preserving synthetic MRI data be created to improve the performance of segmentation models?
2) How can realistic brain MRI images (corresponding to a chosen modality) be synthesized by a GAN model given abnormal brain tissue (e.g. tumor/lesion) information and the corresponding modalities as inputs?
3) How can the synthetic MRI data be used for the network training and validation in such a manner that improves network generalizability and accuracy and preserves the underlying properties of real-world data?

This work investigates the feasibility of synthesizing MRI dataset using generative adversarial networks (GANs) [4], by creating realistic looking images of brain gliomas. The following hypothesizes are researched:

1) Synthetic brain MRI data and lesion information can help improve the performance of neural network models for lesion segmentation tasks.
2) Realistic and context-aware synthetic data provide enough radiomic information for training deep neural networks.

The scope of the work presented here is to determine if:

- Realistic synthetic data can be generated to provide enough AI-based applications in medical imaging.
- A deep network can be trained on the synthetic and reference data for brain lesion segmentation.

In the upcoming subsections, the relevance of brain tumors and the groundwork for technologies used in this thesis will be introduced.

1.1 Gliomas

Glioblastoma is the most common type of brain cancer. Glioma tumors make up to 80% of all malignant brain tumors [5]. Gliomas have a high morbidity and mortality rate - with optimal treatment, the median survival rate is between 12 to 15 months for patients with glioblastomas, and 2 to 5 years for patients with anaplastic gliomas [6].

Patients suspected of having glioblastomas are usually subjected to do a computed tomography (CT) of the brain. If a mass is identified then further MRI scans are ordered, typically these being the following contrast enhanced sequences: T2-weighted (t2), T2-fluid-attenuated inversion recovery (flair), T1-weighted (t1), T1-weighted contrast-enhanced (t1ce). These scans are useful for pre-operative surgical planning and post-operative radiotherapy planning [7].

Gliomas can be classified by type of cell, by grade (severity), and by location – all which are important for diagnosis. It is crucial for radiologists to be able to correctly classify these tumors and observe their development in time, so that the patient can be prescribed a treatment, and be given an accurate survival prognosis [8]. While this can be carried out manually by an expert radiologist, it is a difficult task prone to errors. As such, it is vital to accurately and automatically detect the location and size of the gliomas in patients.

1.2 Computer Vision

Computer vision is a subfield of artificial intelligence, with the goal of automating tasks that the human visual system can do and perform them better or in other words, it aims to extract meaningful data from images [9]. With computer vision, images can be automatically classified into groups [10], autonomous vehicles can detect imminent dangers and be navigate their surroundings [11], and cameras can detect and even identify human faces [12].

A specific task of computer vision is image segmentation. Image segmentation deals with categorizing the pixels of an image into classes, making it easier to analyse. This can be achieved by using algorithms such as Otsu's threshold method [13], canny edge detection [14], K means clustering [15], and many others. However, these algorithms are difficult to optimize, as various parameters need to be tested until the user is satisfied with the results. Additionally, user analysis is subjective and is in itself not a good metric for optimization.

1.3 Deep Learning

The problem of manually optimizing algorithms brings the need to optimize algorithms automatically, which is what the discipline of Machine Learning (ML) attempts to solve [16]. ML algorithms improve themselves by learning from "experience" and data. This process of learning allows them to make predictions and decisions without being programmed to do so directly. ML is used for tasks that are normally labour intensive, expensive, or challenging. The difficulty of these tasks is due to the multidimensionality of data involved, making it very easy for humans to make mistakes when determining the relationship between multiple features and outcomes. This type of problems where the input is mapped to a single correct output can be solved by ML using supervised learning, a method of training ML algorithms using labeled data for classification [17] or regression [18] problems.

Deep Learning (DL), an advanced approach to ML, is used to automatically process raw data into features and then output a human-readable result for the user [19]. The main component is the artificial neural network (commonly known as “model”): a mechanism which is able to learn from data which has not had any feature hand-engineering, by learning to extract these features on its own. This is also particularly useful when dealing with visual data, as there is not a simple way to hand engineer features from pixel arrays.

Models consist of multiple layers. These layers are formed of neurons, and the individual neurons from each layer are connected to neurons in the previous and next layers, forming a network. The data (signal) enters the model through the input layer, which feeds forward the data to the middle/hidden layers. These layers process the data, passing it from one layer to the next, up to the output layer which returns the prediction in a human-readable format, as shown in Figure 1.

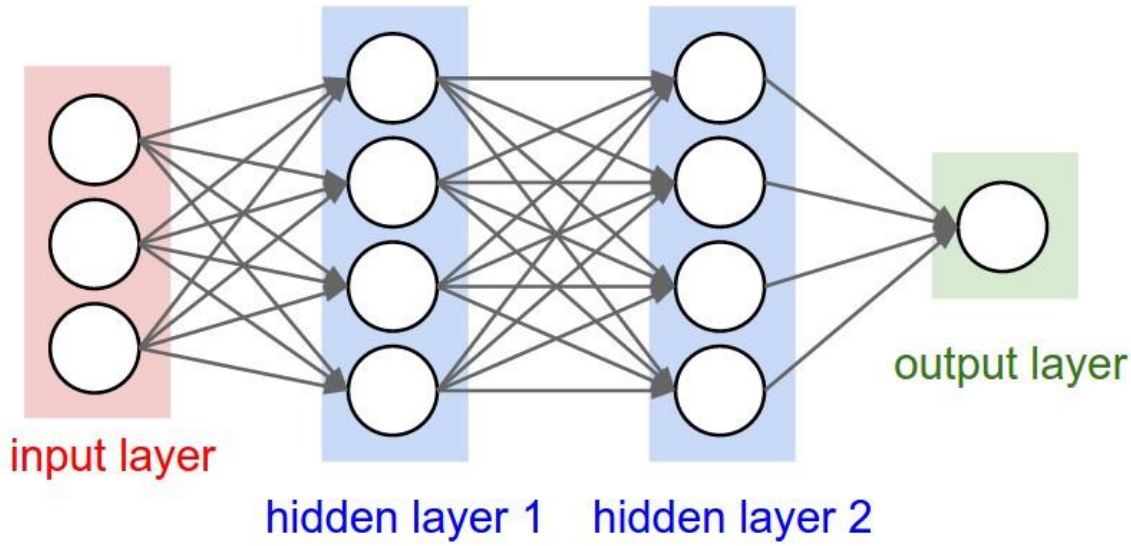

Figure 1. A basic neural network consisting of 4 layers. Source - Stanford CS course [20].

The process of data flowing forward through the network is called feed forward propagation. The forward propagation is implemented in three steps.

First, each input value x_i is multiplied by the weight w_i, obtaining the dot product x.w (Equation 1).

$$x.w = x_1 \times w_1 + x_2 \times w_2 + \ldots + x_n \times w_n$$

Equation 1. The first step processes the data from the input/previous layers as a dot product.

Second, a bias b is added to offset the activation function in the direction of the required output values (Equation 2).

$$z = x.w + b$$

Equation 2. Adding bias to the dot product.

Finally, it is necessary to use a non-linear activation function, since otherwise the neurons would be a simple linear function. Some common activation functions are sigmoid, softmax, rectified linear unit (ReLU), LeakyReLU, tanh. The sigmoid activation function is computed as shown below (Equation 3).

$$y = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Equation 3. The sigmoid activation function.

However, in order for the model to learn, it also needs feedback - which is provided by a loss function (cost function). The loss function adds a cost to the prediction, which is then fed back through the model using the backpropagation algorithm [21] which computes the gradient of the loss function and then adapts the weights of the network layers to properly “fit” the desired result – known as “training the model”.

1.4 Data Augmentation

ML approaches, including DL, require a huge amount of data. If the training dataset is small, a complex model can learn the correct outputs for each data point and over fit, known as overfitting [22]. One of the methods of avoiding this

problem is data augmentation [23]. In visual datasets, augmentation methods represent flipping, rotation, cropping, etc. Having more quality data leads to better model performances, so an automatic augmentation method which provides useful data is essential for use-cases when getting labeled data is expensive.

1.5 Generative Adversarial Networks

Data augmentation can also be achieved with the help of GANs [4], a framework which is able to train models capable of generating realistic images. Two agents compete against each other in a zero-sum game: the generator and the discriminator. The generator attempts to replicate images found in the dataset, without being able to see them directly. The discriminator receives real samples from the dataset, and synthesized (fake) samples made by the generator. It makes its prediction on which samples are real and fake, after which it receives the true labels (real or fake) and learns from them. The generator further gets information from the discriminator's predictions on how well the fake data performed through the backpropagation algorithm, learning how to create more realistic data afterwards, as shown in Figure 2.

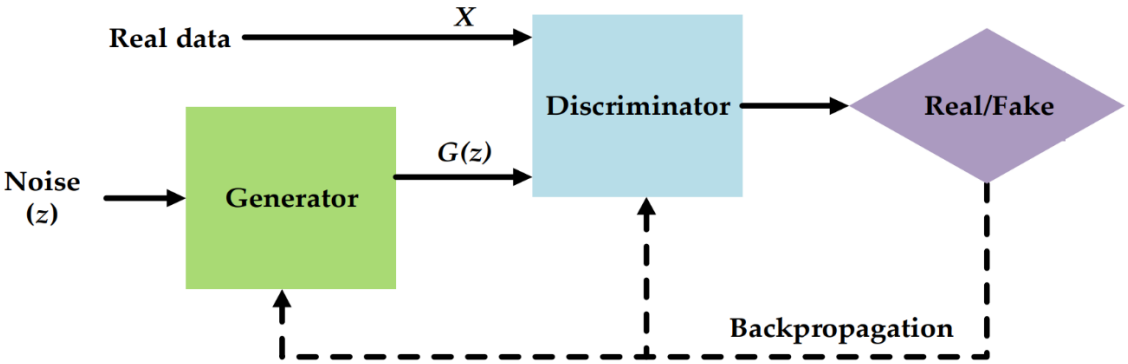

Figure 2. The base idea for the generative adversarial network framework. The generator G takes a latent noise vector z as input, generates fake data G(z) which gets fed to the discriminator D along real data X. The discriminator makes a prediction on whether the input images are real or fake, and then the backpropagation algorithm updates the model weights. Source paper [24].

1.6 Conditional GANs

While the initial GANs were generating images from random noise, more advanced GANs were able to separate the noise in a way that the user would be able to “control” the generated output using a condition [25]. Pix2Pix is a GAN that controls the output with an input image, thus translating the input image to a desired output/style [26]. Upon this, a new algorithm for unpair cycle consistent image-to-image translation can be completed with CycleGAN, which allows translations such as from a satellite view to a map view, creating art from sketches or visualizing or creating a night-time scenery from a daytime picture [27]. The network follows the U-Net model [28], and it makes use of the encoder-decoder architecture to extract low-level features, while also adding skip connections to maintain the positional information from the high-level features during decoding.

1.7 GANs for Medical Image Analysis

Despite GANs being a relatively new concept, it has quickly found its way in medical imaging, with many researchers finding promising applications for them [29] [30]. For example, GANs can be trained for image reconstruction: since doing CT and positron emission tomography (PET) scans is potentially hazardous to the health of the patient due to the radiation required for acquiring these images, it is necessary to reduce the amount of radiation during these examinations at the trade-off of lower quality / increased noise in images. GANs have been used to denoise these low-dose radiation CT and PET scans, and it has been shown they are feasible for this application [31] [32].

Considering that it is necessary to reduce the number of scans carried out in order to reduce radiation dosage, GANs have been shown to aid this as well [33]: by having PET scans done, it is useful to implement image translation to a CT scan which means less harm done to the subjects – and vice versa. In the case of MRIs, carrying out scans of multiple sequences (flair, t1, t2) is time-consuming and difficult. As such, translating from one sequence to the others while

maintaining the core information of the scan, is a useful application of GANs [34] [35].

However, that requires carrying out scans of a subject and then converting them to different modalities or MRI sequences. An interesting application for GANs is generating new subjects that do not exist, in order to augment datasets used for developing segmentation algorithms. It has been shown that poor-performing models can highly benefit from GAN-based data augmentation, however, as the quality of models increases it is more difficult to observe an improvement in their performance [36]. TumorGAN is a model that was developed to generate 2D slices of MRI images of certain modalities through the use of a conditional label, to provide annotations that can be used for supervised learning [37]. MM-GAN has been implemented to generate 3D MRI images of brain tumors and liver lesions by providing a deformed label to produce abundant of data [38].

2 Data Preparation and Network Training

2.1 Brain Tumor Segmentation Dataset

The Brain Tumor Segmentation Challenge 2020 (BraTS2020) training dataset consists of 369 subjects, each subject having five 3D images: four scanned MRI sequences (flair, t1, t1ce, t2) and the ground truth (tumor segmentation), with each 3D image having the resolution 155x240x240. Each ground truth segmentation followed a thorough procedure, being manually annotated by a team of neuroradiologists [39][40]. The dataset is used to identify the best machine learning algorithm for brain tumor segmentation [41]. The data uses the Neuroimaging Informatics Technology Initiative (NIFTI) file format.

Building such a dataset uses many resources across multiple institutions, and it is difficult to develop a big enough dataset to train an efficient segmentation model for every serious disease affecting people, even though various projects have been created for specific diseases already [42]. Therefore, it is critical to be able to extrapolate patterns from a smaller dataset for augmentation purposes, to improve the performance of DL models.

2.2 Data Preprocessing

MRI images are obtained using various protocols and machines, which often leads to inconsistent intensity scales. Consequently, an important pre-processing step is normalizing the data, which improves the performance of image processing methods [43]. It has been noted that bounded output activation functions allow generators to learn quicker [44], so the output function of the generator used is the Tanh function. Since the generator can only output values between -1 and 1 with Tanh, the images were rescaled to the range [-1,1] to allow the generator to recreate images of the same values as the real ones.

The size of each image is 155x240x240 pixels. Using full images for training requires more video random access memory (VRAM) than available, thus 30

slices were cut from both the top and the bottom of the 3D image for a total of 60 slices, bringing the image size down to 95x240x240. The edges of the images are mostly background pixels without brain tissue, so cropping them out do not cause serious information loss. Unfortunately, the VRAM problem was not completely solved. Cropping more slices out would cut out a significant amount of brain tissue, so instead of cropping, images were downsampled from 95x240x240 down to 64x240x240.

The purpose of the generated images and conditions is to aid the training of a segmentation (classification) model. For segmentation tasks, it is undesirable to create new digital values during image preprocessing, and as such nearest neighbour interpolation was preferred over other downsampling (interpolation) methods [45].

During training, the axial view has been used for visual inspection as it provided a better idea on the quality of the image, which is why it was preferred to not apply changes to the axial view. The process of cropping and downsampling is visualized in Figure 3.

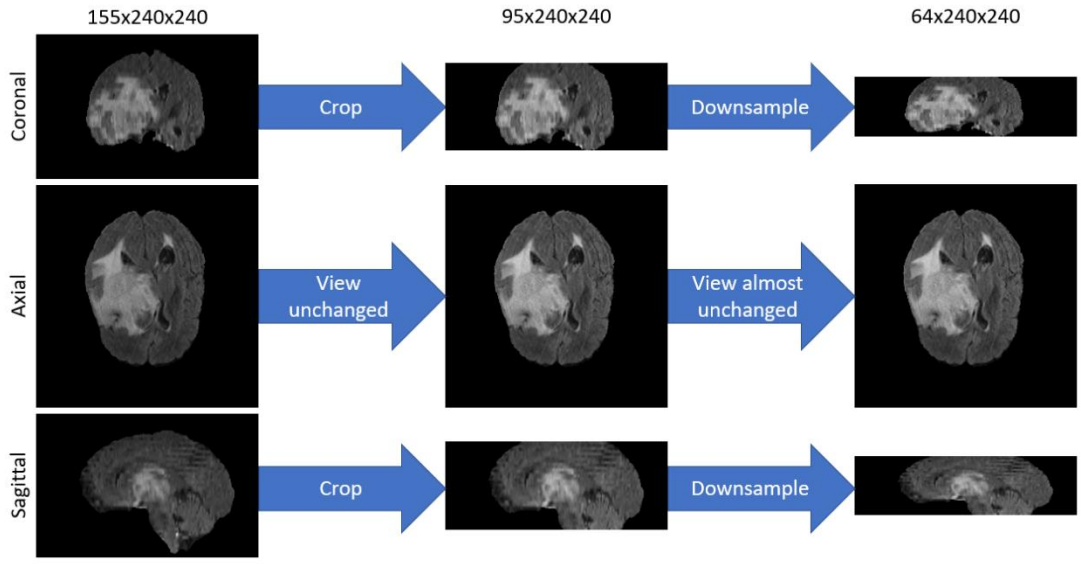

Figure 3. Cropping and resizing a sample image (flair sequence) from resolution 155x240x240 down to 64x240x240, shown from all (coronal, axial, sagittal) views.

2.3 Condition

In order to train a segmentation model on the generated data, it is necessary to have the ground truths for the data. In the BraTS2020 dataset, this has been done with manual segmentation, meaning that tumors have been annotated by expert radiologists. When generating new data, it is possible to control the ground truth by using a condition, as done in Pix2Pix [26]. For the current training pipeline, the condition is represented by a semantic label image, similar to the one used in TumorGAN [37].

The goal of the semantic label is to mix the data from two patients (A and B), or in other words: apply the tumor of patient A onto the brain of patient B. This is done by summing together the ground truth label of subject A with the brain mask of subject B. In some cases, the newly generated label has the tumor bigger than the new brain – to solve this, after transferring the tumor to patient B the brain mask B was applied to the new label to remove the tumor outside of it, thus fitting the new brain shape. The process is described in Figure 4.

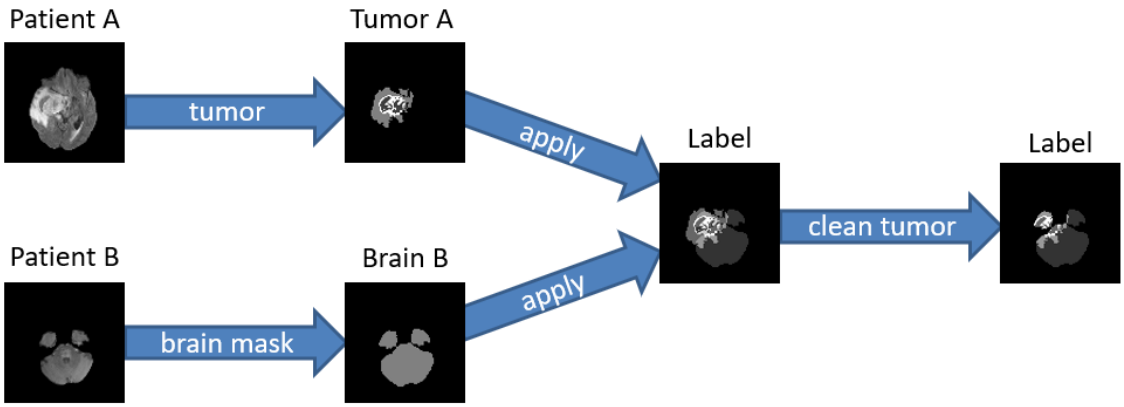

Figure 4. Creating the semantic label image of the tumor of patient A and brain shape of patient B, and then removing the external part of the tumor in the new brain.

This way, in a train dataset of 296 subjects it is possible to create 43660 (combination of 296 subjects taken 2 at a time) pairs of tumor/brain subjects, which is much more data than that which had been gathered and labeled

manually. The impact of having a very large amount of data is questionable due to diminishing returns of data [46]. However, it is important to note that for smaller datasets, this type of pairing would yield significant improvements.

2.4 Model architecture

The network architecture is similar to Pix2Pix [26], except that it uses a U-Net with 3D convolutions instead of 2D convolutions [47]. The U-Net architecture is able to process images and also generate images, allowing the creation of new subjects from visual conditions. Even though the intended use of the U-Net was to segment an image, in this GAN the architecture is used for the opposite task – from a segmented condition, generate a subject's image. This is exactly the task of this GAN, which is why the U-Net architecture works well for it.

While the usual U-Net uses 2D convolutional layers, the U-Net3D is adapted to be able to process 3D volumes by using 3D convolutional layers. This allows the network to get spatial awareness and have a better understanding of the brain texture. The network is 4 levels deep, with an encoder-decoder structure as shown in Figure 5. Each level on the encoder path decreases the image shapes by half, and doubles the convolutional filter count. The images are decreased in size using max-pooling 3D layers. On the decoding side, the images are upsampled using nearest neighbour interpolation, and the filter count is halved at each level. The model also employs skip connections from the encoder to the decoder of the same image sizes – at a given decoding layer, the inputs from the previous layer and the encoder are concatenated together as input.

The discriminator follows the PatchGAN [26] architecture which is the same as the 3D U-Net, except that there is no decoder – the last layer of the discriminator is connected to the output of the encoder Figure 6.

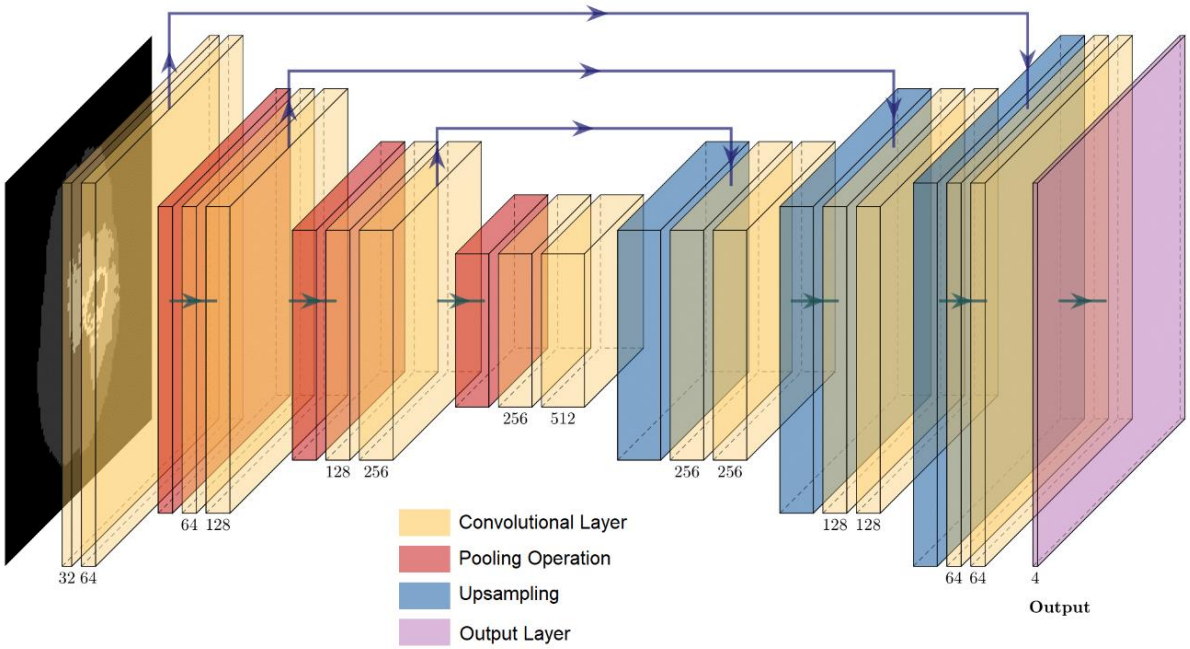

Figure 5. The architecture of the 3D U-Net generator. Code reference [48]. The input is a semantic label 3D image. Created with PlotNeuralNet [49].

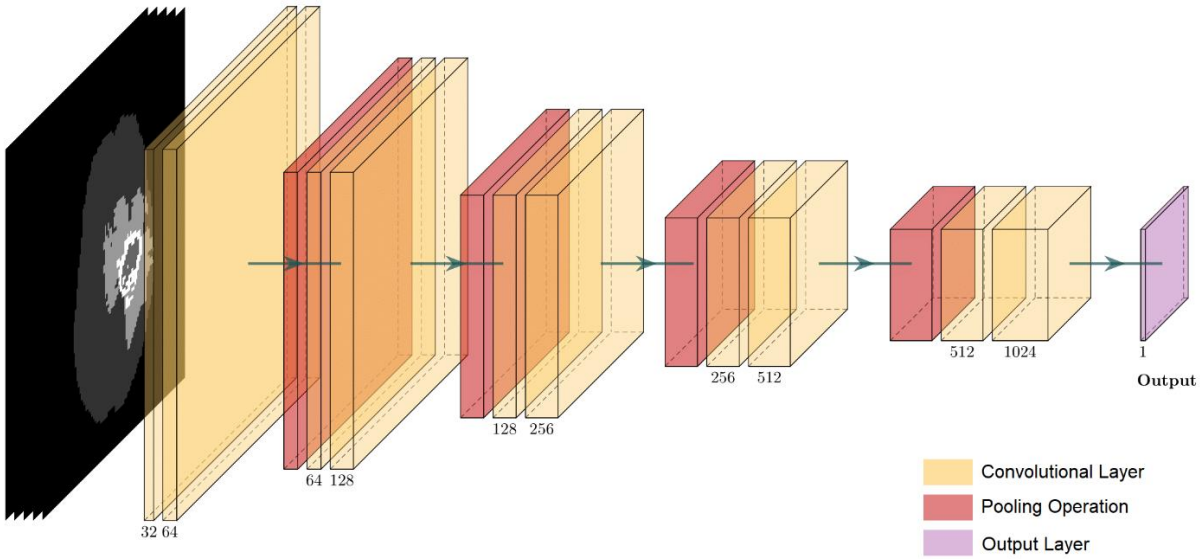

Figure 6. The architecture of the 3D PatchGAN discriminator. The input consists of the 4 MRI modalities and the semantic label. Created with PlotNeuralNet [49].

2.5 Training experiment

The objective function (loss function) is the mean squared error (MSE) loss, which has been shown to be efficient for GANs in the Least Squares Generative Adversarial Network [50]. Equation 4 and Equation 5 show the objective functions of the two agents, where D represents the discriminator, G represents the generator, and x represents a series of images (input). The discriminator has to predict real images as 1 (label real), and synthesized images as 0 (label fake/synthetic). The generator, on the other hand, has to synthesize images that the discriminator would predict as 1. This is implemented as a minimax game [4], where both agents compete against each other. The generator loss gets lower (better generator) by learning how to synthesize samples which make the discriminator perform worse. As described previously, the generator does not aim to affect the discriminator's predictions on the real dataset – the generator never gets direct feedback from the real images.

$$L_D = \frac{1}{2}[D(x_{real}) - 1]^2 + \frac{1}{2}D(x_{fake})^2$$

Equation 4. The loss function (MSE) of the discriminator. It can be described as trying to minimise the loss the discriminator's predictions compared to the true labels.

$$L_G = [D(x_{fake}) - 1]^2$$

Equation 5. The loss function of the generator. The goal is to get the discriminator to predict the generated images as real. In other words, the generator wants to maximise the discriminator's loss when predicting the label of the generated images.

The backpropagation gradient is computing using the Adam optimizer [51] for both agents, and the step-size parameter (learning rate) for both agents is set to 1e-4.

Stabilization is a difficult problem in GANs, where the discriminator learns much faster than the generator, preventing the latter from being trained further. To balance each agent's training, spectral normalization (SN) [52] is used for each convolutional block.

Due to the high memory demands of the network and computing power constraints, the input batch size to the GAN is 1 (two patients are used for each subject synthesis, but only one subject is processed at a time). As such, instance normalization is used, as it has been shown to perform better than batch normalization for small batch sizes [53].

This setup requires a minimum of 30 GB VRAM for training. The training was done on the DGX server of the Turku University of Applied Sciences, and the graphical processing unit (GPU) used has been NVIDIA Tesla V100. The training is done using a single GPU (no parallel computing). The training takes 240 hours. The data preparation, semantic label, model architecture and training of the GAN are implemented using PyTorch 1.9.0 [54].

Both the generator and the discriminator are trained for 94720 iterations each, for a total of 1280 epochs. The train dataset contains 296 subjects, and as such an epoch consists of 148 iterations, where two unique subjects (A, B) are chosen at each step for a total of 296 training subjects each epoch.

The generator receives a semantic label as input, after which it creates a synthetic subject with 4 MRI sequences. The discriminator predicts whether the 4 sequences, together with the semantic label represent a real or a synthetic subject.

The discriminator and the generator are trained in an alternating order with each iteration, where the first iteration the discriminator is trained, then the generator is trained in the next iteration, and the process repeats until the last iteration. During the generator update, subject AB is synthesized using a semantic label image created from tumor of subject A and brain mask of subject B, after which the discriminator makes its prediction on the true label of the subject, and then the loss is backpropagated to the generator (Figure 7). During the discriminator

update, the generator synthesizes a new subject AB, after which the discriminator makes its predictions on whether subject B with the corresponding label and synthetic subject AB are real or synthetic (Figure 8).

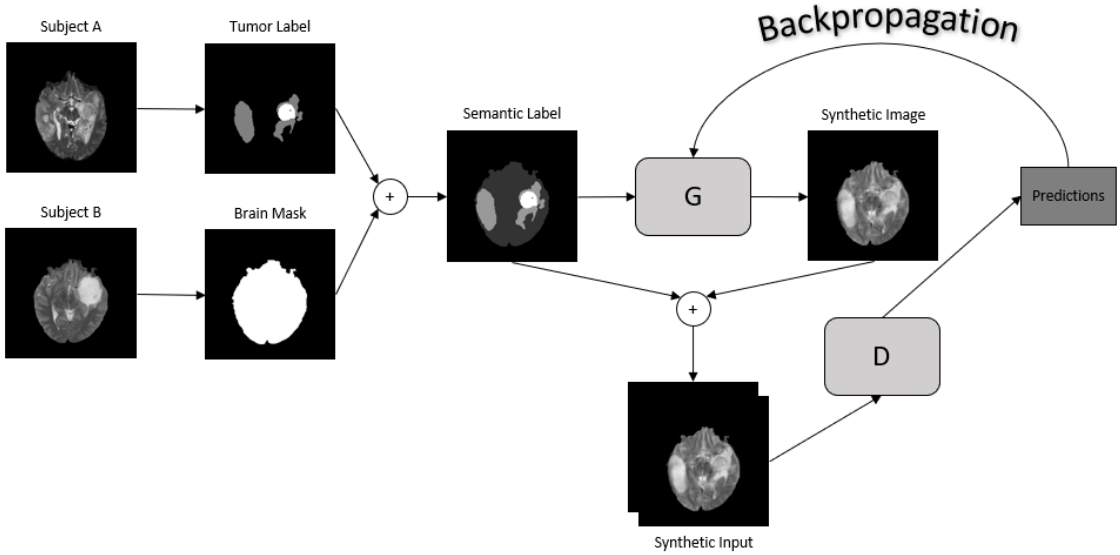

Figure 7. Generator (G) training pipeline. From the dataset, two different random patients A and B are chosen. The semantic label is created and then fed to the generator. The generator creates fake MRI images, which together with the semantic label make up the fake input to the discriminator. The discriminator (D) predicts whether the images are real or not, and the goal of the generator is to get the discriminator to predict them as “real”. Based on these predictions the weights of the generator are updated.

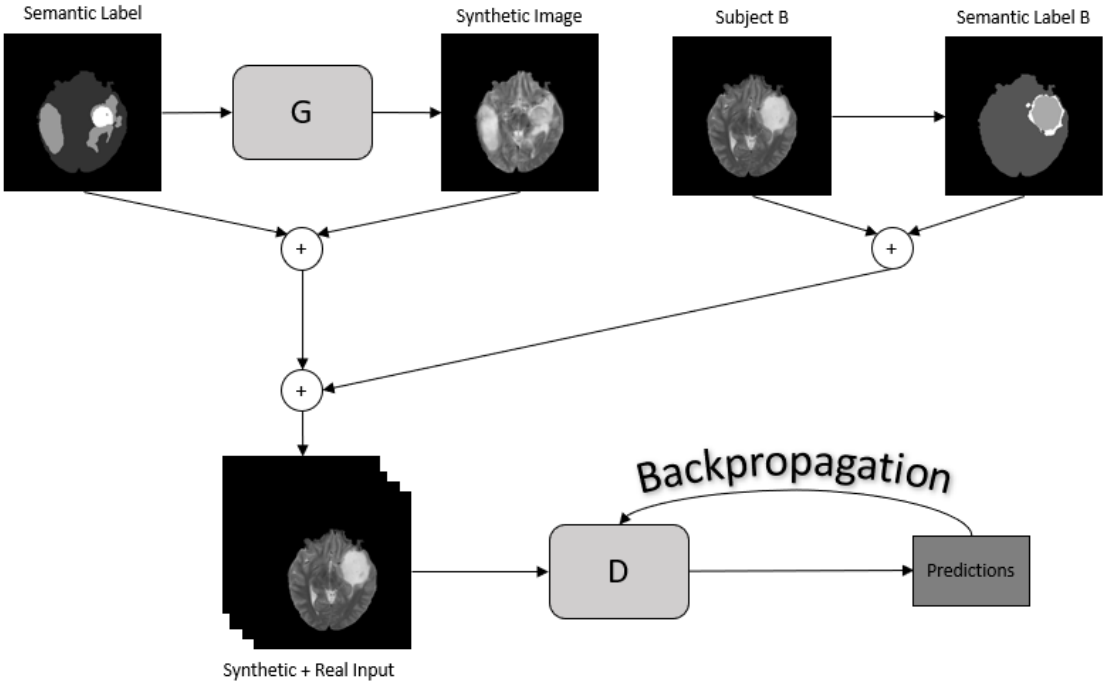

Figure 8. Discriminator (D) training pipeline. The generator (G) pipeline is identical to Figure 7, which creates the fake input to the discriminator. The real input is created by building a semantic label using the tumor and the MRI images from the same patient B. The fake and real inputs are fed to the discriminator, which makes its predictions with the goal of predicting “real” for the real input, and “synthetic” for the synthetic input. These predictions are used to update the discriminator .

3 Results

3.1 Visual inspection

The model has been trained to generate images which follow the input condition successfully, having brains with the given masks, and tumors in the appropriate locations with the tumor tissues being clearly different by the labeled stages. The synthesized images show the content to be slightly blurry. The brain regions are not properly replicated, with the finer details lacking. Whereas the images cannot be characterised as realistic-looking, the contrast between tumor tissue and brain tissue is evident (Figure 9).

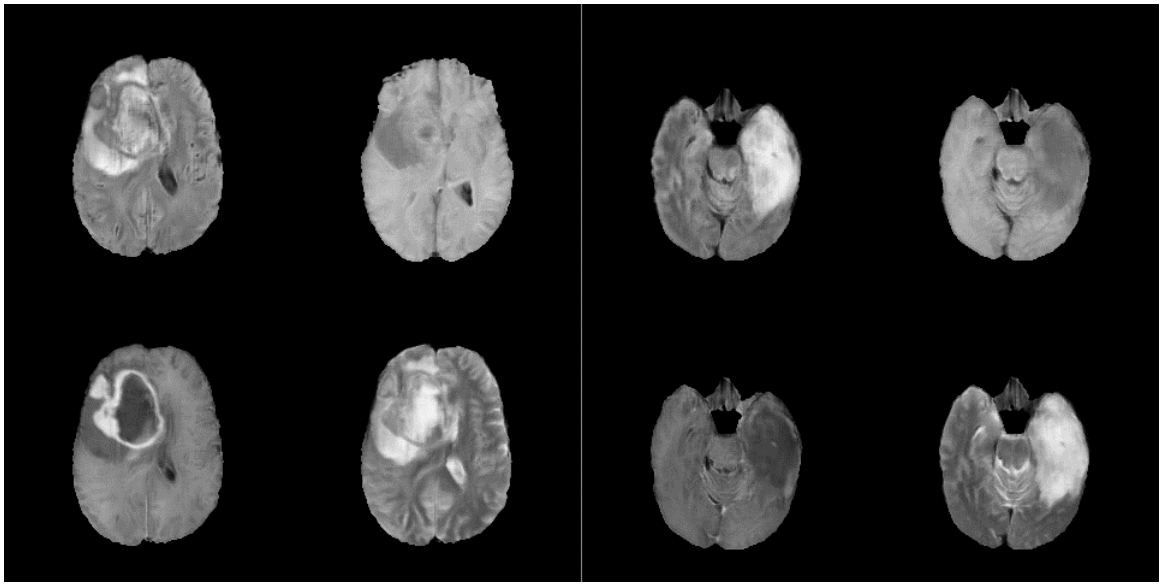

Figure 9. Two synthetic subject samples of all four MRI sequences, synthesized by the generator, from the axial view. Sequences: top left – flair, top right – t1, bottom left – t1ce, bottom right – t2. More samples available in Appendix 1 – Synthetic Samples (Figure 11).

One of the main problems with GANs is mode collapse, which happens when the generator learns to synthesize the same type of data repeatedly, with small (or no) variance between each sample [55]. It has been observed that the synthesized tumors generally are less diverse than their real counterparts, which suggests that a subtle mode collapse is happening during the training of the GAN.

3.2 Evaluation using a Segmentation Model

DeepMedic [56] is a convolutional neural network used to process biomedical 3D NIFTI data and detect structures, such as tumors, within it. The model makes use of a parallel processing architecture with two pathways: a normal resolution and a low-resolution pathway. The normal resolution pathway processes local tissue, and the low-resolution pathway processes a downsampled wider volume of data, allowing the model to distinguish the texture difference in otherwise locally-similar data by providing contextual information (Figure 10). The performance of DeepMedic is assessed by training the model on a train dataset, and then segmenting a test dataset with the trained model. The predicted segmentations are compared with the ground truth labels using the Dice similarity coefficient (DSC) (Equation 6) [57].

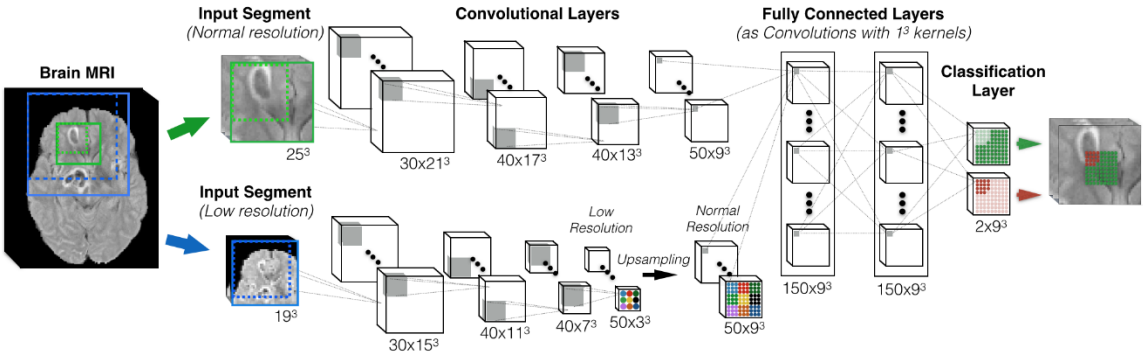

Figure 10. An example of a double-pathway architecture for multi-scale processing [56].

$$DSC = \frac{2 * TP}{2 * TP + FP + FN}$$

Equation 6. The equation for the DSC. TP - true positives, FP - false positives, FN - false negatives. This is the same equation as the F1 score.

To evaluate the synthetic data with DeepMedic, augmented datasets and fully synthetic datasets must be created for the experiments. Additionally, a GAN is

trained without using SN and it is used for creating another synthetic dataset, to determine whether the training is improved by the use of SN.

Synthetic subjects are obtained by choosing random pair of patient A and B from the training dataset. For both the GAN trained with and without SN, the following datasets are created:

1. An augmented dataset which includes the real train data of 296 real subjects and the synthetic data of 296 synthesized subjects
2. (1x) A dataset of 296 synthetic subjects
3. (2x) A dataset of 592 synthetic subjects
4. (3x) A dataset of 888 synthetic subjects

The performance of these datasets is compared against a real dataset which consists of only the real train data of 296 real subjects. The evaluation of each of these datasets is done on the test dataset which consists of 73 real subjects.

The data pre-processing that is used to train the GAN is also applied to the real data on which DeepMedic is trained and evaluated.

The DeepMedic model trained on each dataset performs the segmentation on the test dataset. The segmentation DSC results are shown in Table 1.

Class / Dataset	Whole Tumor	Tumor Core	Enhanced Tumor
Real	0.611	0.716	0.756
Real + 1x Synth SN	0.595	0.700	0.744
Real + 1x Synth	0.601	0.698	0.758
1x Synth SN	0.028	0.060	0.027
1x Synth	0.035	0.100	0.106
2x Synth SN	0.177	0.118	0.120
2x Synth	0.100	0.140	0.207
3x Synth SN	0.174	0.095	0.081
3x Synth	0.100	0.150	0.179

Table 1. Average Dice scores of the DeepMedic trained on various datasets. Synth SN datasets are synthesized by the GAN which is trained using spectral normalization.

3.3 Discussion

The performance of DeepMedic decreases when the dataset is augmented using synthetic data. Training the segmentation model using only synthetic data will lead to it performing very poorly. It has been noted that increasing the amount of synthetic data slightly increases the performance of the model when no real data are included in the dataset, which indicates that despite the poor quality of the data, there are some features which can be learnt by having more data. Previous experiments in the work have shown that increasing the amount of synthetic data in the augmented (real+synthetic) dataset leads to worse performance, so more synthetic data are better only when no real data are used. There is little difference between training the GAN with SN and without, which suggests that for this particular architecture there is no need of stabilizing the training.

Despite that the quality of the images have a good general structure, the texture of the brain is poor which leaves room for improvement. Further development can be done into using a local discriminator [58] and/or a perceptual loss [59]. The local discriminator would focus on the tumor area, helping the generator learn

how to better emulate the real tumors. The perceptual loss would require a model such as VGG19 [60] pre-trained on the ImageNet [61] dataset to analyse the synthesized images, however, the mentioned network was developed for analysing 2D images, and a method for implementing the 2D loss or using a 3D model is necessary.

Both of these improvements would increase the size of the gradient, which is already at the limit of the highest performing available GPUs. The computing power limit requires a method for reducing the memory requirements of the pipeline, without losing data or performance.

4 Conclusion

This thesis has explored a conditional GAN for (image-to-image) translation to generate synthetic brain MRI images of multiple sequences, including context radiomic features of the real data such as the brain tissue and cancer lesions using generative adversarial networks.

The relevant work in the area has been researched and presented. The network was trained on the Brain Tumor Segmentation Challenge 2020 dataset, and the training process has been described in detail. The generator can create previously inexistent subjects, while providing the annotations for the new subjects. The synthetic tumors are clearly differentiated from the healthy tissue. The finer structure of the brain is slightly blurry. The data generated by the model cannot be used for clinical decision-making procedures, as the synthesized data did not show signs of improvement when training segmentation models. The network requires fine-tuning, which remains an important task, and improvements for future works have been suggested in the discussion section.

References

[1] F. Retter, C. Plant, B. Burgeth, G. Botella, T. Schlossbauer, and A. Meyer-Bäse, “Computer-aided diagnosis for diagnostically challenging breast lesions in DCE-MRI based on image registration and integration of morphologic and dynamic characteristics,” *EURASIP J. Adv. Signal Process.*, vol. 2013, no. 1, p. 157, Oct. 2013, doi: 10.1186/1687-6180-2013-157.

[2] G. Litjens *et al.*, “A survey on deep learning in medical image analysis,” *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[3] D. Ravì *et al.*, “Deep Learning for Health Informatics,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 4–21, Jan. 2017, doi: 10.1109/JBHI.2016.2636665.

[4] I. Goodfellow *et al.*, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[5] M. L. Goodenberger and R. B. Jenkins, “Genetics of adult glioma,” *Cancer genetics*, vol. 205, no. 12, pp. 613–621, 2012.

[6] F. E. Bleeker, R. J. Molenaar, and S. Leenstra, “Recent advances in the molecular understanding of glioblastoma,” *Journal of neuro-oncology*, vol. 108, no. 1, pp. 11–27, 2012.

[7] G. Shukla *et al.*, “Advanced magnetic resonance imaging in glioblastoma: a review,” *Chin Clin Oncol*, vol. 6, no. 4, p. 40, 2017.

[8] J. W. Henson, P. Gaviani, and R. G. Gonzalez, “MRI in treatment of adult gliomas,” *The lancet oncology*, vol. 6, no. 3, pp. 167–175, 2005.

[9] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[10] S. Ito and S. Kubota, “Object classification using heterogeneous co-occurrence features,” in *European Conference on Computer Vision*, 2010, pp. 701–714.

[11] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE access*, vol. 8, pp. 58443–58469, 2020.

[12] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[13] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[14] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[15] S. Lloyd, “Least squares quantization in PCM,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[16] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[17] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” *Emerging artificial*

intelligence applications in computer engineering, vol. 160, no. 1, pp. 3–24, 2007.
[18] M. Fernández-Delgado, M. S. Sirsat, E. Cernadas, S. Alawadi, S. Barro, and M. Febrero-Bande, “An extensive experimental survey of regression methods,” *Neural Networks*, vol. 111, pp. 11–34, 2019.
[19] L. Deng and D. Yu, “Deep learning: methods and applications,” *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
[20] “CS231n Convolutional Neural Networks for Visual Recognition.” https://cs231n.github.io/neural-networks-1/ (accessed Mar. 28, 2022).
[21] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, “Backpropagation: The basic theory,” *Backpropagation: Theory, architectures and applications*, pp. 1–34, 1995.
[22] D. M. Hawkins, “The problem of overfitting,” *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.
[23] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[24] J. Feng *et al.*, “Generative Adversarial Networks Based on Collaborative Learning and Attention Mechanism for Hyperspectral Image Classification,” *Remote Sensing*, vol. 12, p. 1149, Apr. 2020, doi: 10.3390/rs12071149.
[25] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
[26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
[27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
[28] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.
[29] “MedGAN: Medical image translation using GANs - ScienceDirect.” https://www.sciencedirect.com/science/article/abs/pii/S0895611119300990 (accessed Feb. 01, 2022).
[30] X. Yi, E. Walia, and P. Babyn, “Generative Adversarial Network in Medical Imaging: A Review,” *Medical Image Analysis*, vol. 58, p. 101552, Aug. 2019, doi: 10.1016/j.media.2019.101552.
[31] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, “Generative Adversarial Networks for Noise Reduction in Low-Dose CT,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 12, pp. 2536–2545, Dec. 2017, doi: 10.1109/TMI.2017.2708987.
[32] Z. Hu *et al.*, “DPIR-Net: Direct PET Image Reconstruction Based on the Wasserstein Generative Adversarial Network,” *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 1, pp. 35–43, Jan. 2021, doi: 10.1109/TRPMS.2020.2995717.

[33] A. Ben-Cohen *et al.*, “Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection,” *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 186–194, Feb. 2019, doi: 10.1016/j.engappai.2018.11.013.
[34] Q. Yang, N. Li, Z. Zhao, X. Fan, E. I.-C. Chang, and Y. Xu, “MRI Cross-Modality Image-to-Image Translation,” *Sci Rep*, vol. 10, no. 1, p. 3753, Feb. 2020, doi: 10.1038/s41598-020-60520-6.
[35] Y. Luo *et al.*, “Edge-preserving MRI image synthesis via adversarial network with iterative multi-scale fusion,” *Neurocomputing*, vol. 452, pp. 63–77, 2021.
[36] H.-C. Shin *et al.*, “Medical image synthesis for data augmentation and anonymization using generative adversarial networks,” in *International workshop on simulation and synthesis in medical imaging*, 2018, pp. 1–11.
[37] Q. Li, Z. Yu, Y. Wang, and H. Zheng, “TumorGAN: A multi-modal data augmentation framework for brain tumor segmentation,” *Sensors*, vol. 20, no. 15, p. 4203, 2020.
[38] Y. Sun, P. Yuan, and Y. Sun, “MM-GAN: 3D MRI Data Augmentation for Medical Image Segmentation via Generative Adversarial Networks,” in *2020 IEEE International Conference on Knowledge Graph (ICKG)*, 2020, pp. 227–234.
[39] B. H. Menze *et al.*, “The multimodal brain tumor image segmentation benchmark (BRATS),” *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.
[40] S. Bakas *et al.*, “Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features,” *Sci Data*, vol. 4, no. 1, p. 170117, Sep. 2017, doi: 10.1038/sdata.2017.117.
[41] S. Bakas *et al.*, “Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge,” *arXiv:1811.02629 [cs, stat]*, Apr. 2019, Accessed: Jan. 26, 2022. [Online]. Available: http://arxiv.org/abs/1811.02629
[42] A. L. Simpson *et al.*, “A large annotated medical image dataset for the development and evaluation of segmentation algorithms,” *arXiv preprint arXiv:1902.09063*, 2019.
[43] M. Shah *et al.*, “Evaluating intensity normalization on MRIs of human brain with multiple sclerosis,” *Medical image analysis*, vol. 15, no. 2, pp. 267–282, 2011.
[44] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *arXiv:1511.06434 [cs]*, Jan. 2016, Accessed: Feb. 02, 2022. [Online]. Available: http://arxiv.org/abs/1511.06434
[45] D. S. S. Baboo and M. R. Devi, “An Analysis of Different Resampling Methods in Coimbatore, District,” *Global Journal of Computer Science and Technology*, Dec. 2010, Accessed: Jan. 25, 2022. [Online]. Available: https://computerresearch.org/index.php/computer/article/view/670
[46] Z. Fabian, R. Heckel, and M. Soltanolkotabi, “Data augmentation for deep learning based accelerated MRI reconstruction with limited data,” in *International Conference on Machine Learning*, 2021, pp. 3057–3067.

[47] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,” *arXiv preprint arXiv:1606.06650*, 2016.
[48] A. Wolny, *pytorch-3dunet*. 2022. Accessed: Mar. 28, 2022. [Online]. Available: https://github.com/wolny/pytorch-3dunet
[49] H. Iqbal, *PlotNeuralNet*. 2022. Accessed: Mar. 28, 2022. [Online]. Available: https://github.com/HarisIqbal88/PlotNeuralNet
[50] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.
[51] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
[52] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.
[53] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.
[54] A. Paszke *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.
[55] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” *Advances in neural information processing systems*, vol. 29, 2016.
[56] K. Kamnitsas *et al.*, “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation,” *Medical image analysis*, vol. 36, pp. 61–78, 2017.
[57] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
[58] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.
[59] C. Ledig *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.
[60] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
[61] 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*, 2009, pp. 248–255.

Synthetic samples

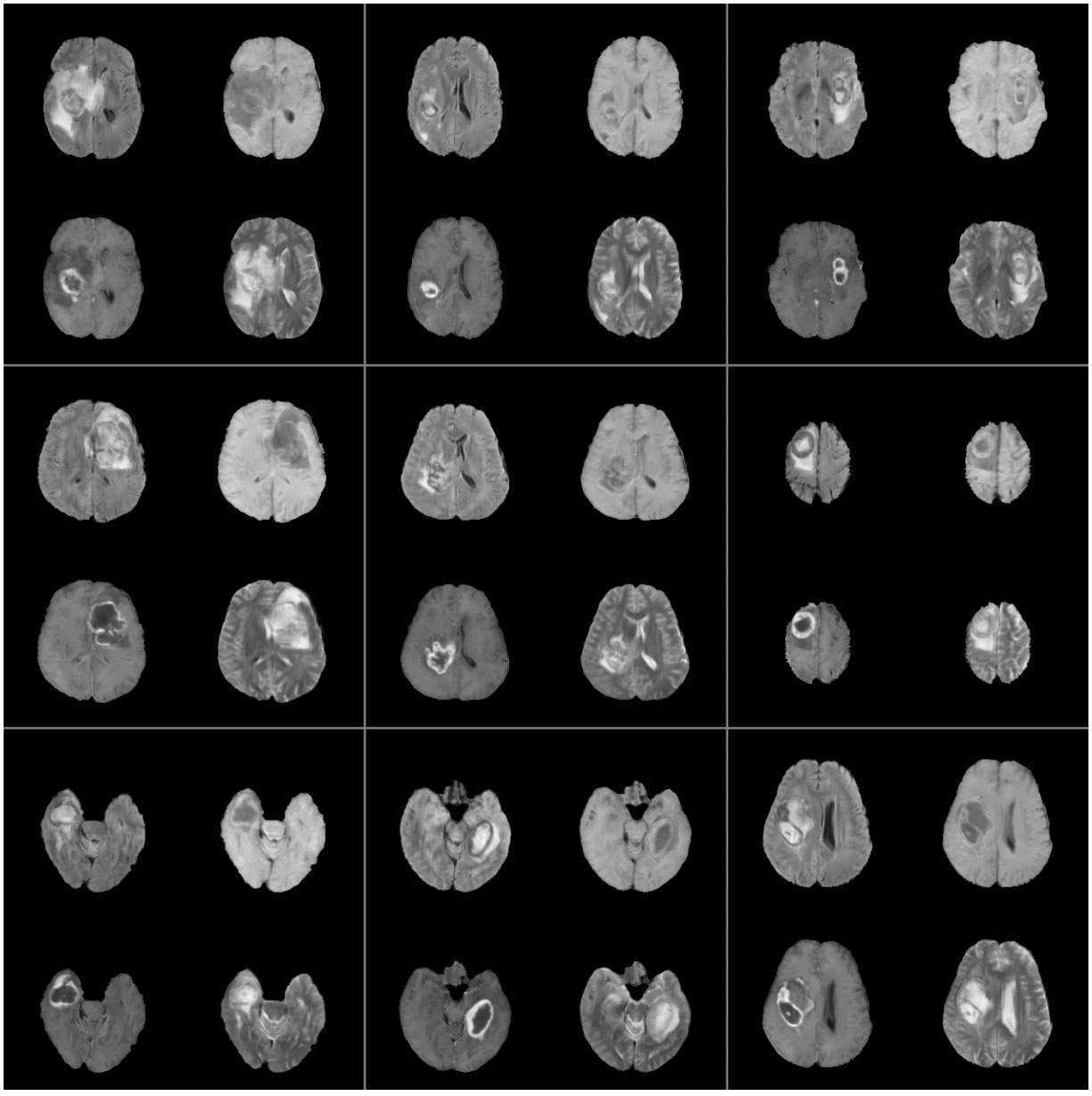

Figure 11. Axial view of 9 synthesized samples. Each sample shows 4 MRI sequences.